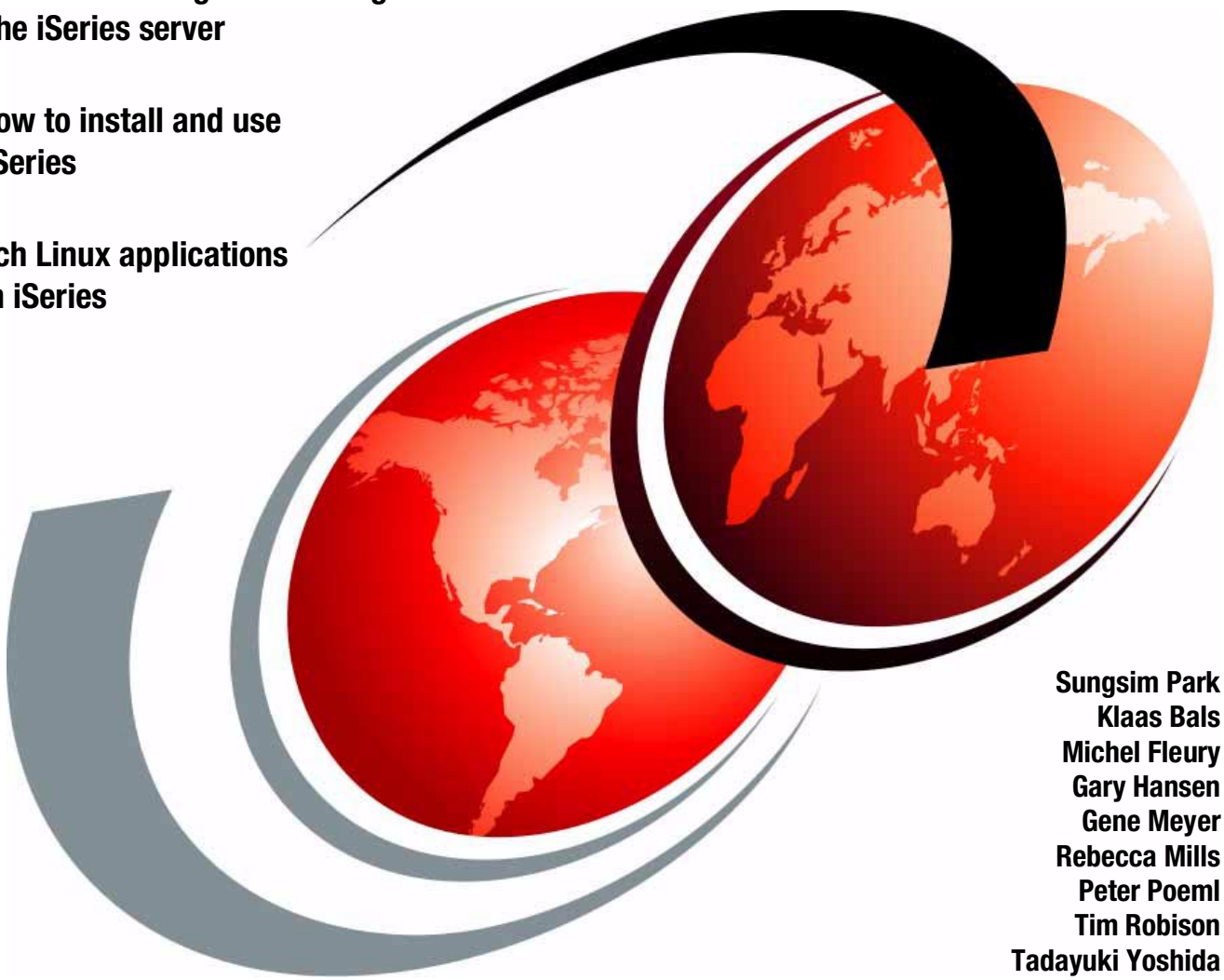
IBM

Linux on the IBM @server iSeries Server

An Implementation Guide

- Benefit from the advantages of running Linux on the iSeries server
- Find out how to install and use Linux on iSeries
- Learn which Linux applications can run on iSeries



Sungsim Park
Klaas Bals
Michel Fleury
Gary Hansen
Gene Meyer
Rebecca Mills
Peter Poeml
Tim Robison
Tadayuki Yoshida

Redbooks

ibm.com/redbooks

International Technical Support Organization

Linux on the IBM @server iSeries Server: An Implementation Guide

January 2002

SG24-6232-00

Take Note! Before using this information and the product it supports, be sure to read the general information in “Special notices” on page xi.

First Edition (January 2002)

This IBM Redbook was updated for softcopy publication on July 2, 2002.

This edition applies to Version 5, Release 1 of OS/400.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JLU Building 107-2
3605 Highway 52N
Rochester, Minnesota 55901-7829

Contents

Contents iii

Special notices xi

IBM trademarks xii

Preface xiii
The team that wrote this redbook. xiii
Special notice xv
Comments welcome. xv

Chapter 1. Getting acquainted with Linux on the iSeries server 1
1.1 What Linux is. 2
1.2 What open source is 3
1.2.1 Distributions 5
1.2.2 Linux on all platforms 5
1.3 Why use Linux on the iSeries server. 5
1.4 Linux on iSeries implementation 6
1.4.1 What is possible 6
1.4.2 Definitions 7
1.4.3 Environment 10
1.4.4 Shared processors 10
1.4.5 Virtual I/O 11
1.4.6 Native I/O: IOAs directly attached to the Linux partition 12
1.5 Linux application scenarios on iSeries 13
1.6 Performance 14
1.7 Summary 15

Chapter 2. Installation and operation of Linux on the iSeries server 17
2.1 System requirements 18
2.2 Getting started. 19
2.3 LPAR considerations. 20
2.3.1 Configuring a service tools user ID 21
2.3.2 Configuring a guest partition 23
2.4 Linux installation 36
2.4.1 Choosing a distribution 37
2.4.2 Installation details 37
2.4.3 Creating an initial network server description (CRTNWSD) 38
2.4.4 Boot parameters of the NWSD 38
2.4.5 Create Network Server Storage Space (CRTNWSSTG) 40
2.4.6 Connecting to the virtual Linux console 44
2.4.7 Starting the Linux partition 49
2.5 Virtual I/O 51
2.5.1 Virtual LAN configuration 52
2.5.2 Using the virtual CD-ROM. 60
2.5.3 Using virtual tape 62
2.6 Linux native adapters and disk (hosted or non-hosted partitions). 62
2.6.1 Allocating resources 62
2.6.2 Native SCSI support 80

2.6.3 Native LAN adapters . . . 81
2.6.4 Configuring for a non-hosted partition . . . 81
2.7 Troubleshooting . . . 81

Chapter 3. Linux administration . . . 87
3.1 General Linux concepts . . . 88
3.1.1 Configuration files . . . 88
3.1.2 Editing files using text editors . . . 88
3.1.3 Configuration tools . . . 89
3.1.4 The man command . . . 89
3.2 Devices . . . 90
3.2.1 Linux devices . . . 90
3.2.2 Creating devices . . . 90
3.2.3 iSeries-specific devices . . . 91
3.2.4 Device name changes . . . 94
3.3 Handling virtual and direct attached hard disks . . . 95
3.3.1 Looking at partition information of hard disks . . . 95
3.3.2 Creating and adding a virtual disk . . . 96
3.3.3 Partitioning the added disk . . . 97
3.3.4 Creating the file system (formatting the disk partition) . . . 99
3.3.5 Mounting a disk partition . . . 99
3.3.6 Unmounting a disk partition . . . 99
3.3.7 Automatically mounting a disk partition at boot time . . . 100
3.3.8 Moving a virtual disk between partitions . . . 101
3.4 Working with network devices (virtual and direct) . . . 101
3.5 Directory hierarchy . . . 102
3.5.1 Standard of directory structure . . . 102
3.5.2 Maintenance of the file system . . . 103
3.6 Users and groups . . . 103
3.6.1 Multi-user system . . . 103
3.6.2 Server-oriented system . . . 104
3.6.3 The root user . . . 104
3.6.4 Managing users . . . 104
3.6.5 Managing groups . . . 106
3.6.6 Layout of /etc/passwd . . . 107
3.6.7 Layout of /etc/group . . . 108
3.6.8 Synchronization of iSeries users and Linux users . . . 108
3.7 Monitoring the system . . . 108
3.7.1 The top command . . . 109
3.7.2 The ps, kill, and killall commands . . . 109
3.7.3 System logs . . . 111
3.8 Security issues . . . 112
3.8.1 Changing file ownership (chown and chgrp) . . . 112
3.8.2 Access control . . . 112
3.8.3 Permission bits . . . 112
3.8.4 Setting permissions . . . 113
3.8.5 Ext2 partition specific access control . . . 116
3.8.6 Pluggable Authentication Modules (PAM) . . . 116
3.9 National language support . . . 118
3.9.1 Locale . . . 118
3.9.2 Locale names . . . 118
3.9.3 Locale categories and settings . . . 119
3.9.4 Locale definition file . . . 121

3.10 Backup and recovery 121
3.10.1 Backup from the Linux side 121
3.10.2 Backup from the OS/400 side 131
3.10.3 Restore from the OS/400 side 133
3.11 Problem determination 135

Chapter 4. Advanced administration and development 137
4.1 Development process 138
4.1.1 Compiling using the GNU C compiler (gcc) 138
4.1.2 C++ 139
4.1.3 Scripts 139
4.1.4 Building with make 140
4.1.5 Debugging programs using gdb 142
4.2 Open source applications for Linux on iSeries 144
4.2.1 Differences between PowerPC and Intel processors 144
4.2.2 RedHat Package Manager (RPM) 145
4.2.3 Building source RPM files (src.rpm) 148
4.2.4 Compiling an open source program yourself 150
4.2.5 Differences among C library revisions 153
4.3 Building the kernel 154
4.3.1 Acquiring the kernel 154
4.3.2 Acquiring the iSeries kernel patch 155
4.3.3 Configuring the kernel 155
4.3.4 Must-have modules for Linux on iSeries 158
4.3.5 Direct attached LAN adapters 159
4.3.6 Direct attached I/O adapters (ibmsis) 159
4.3.7 Compiling the kernel 161
4.4 Installing the kernel 162
4.5 Problem determination 164

Chapter 5. X Windows and OpenOffice 165
5.1 X Windows concept 166
5.1.1 What X Windows is 166
5.1.2 X Window Servers for Microsoft Windows 167
5.1.3 X Window Servers for Linux 167
5.1.4 IBM Netvista Thin Client 167
5.2 Basic X operation 168
5.2.1 Basic steps 168
5.2.2 Security 169
5.2.3 Window managers 169
5.3 A graphical login screen (XDM) 171
5.3.1 Step 1: Checking the initial XDM configuration 172
5.3.2 Step 2: Keeping the local displays from starting 173
5.3.3 Step 3: Giving access to connect 173
5.3.4 Step 4: Starting the listener 174
5.3.5 Step 5: Configuring and starting the X Window Server 174
5.3.6 Step 6: Automatically starting XDM 176
5.3.7 Possible problems with XDM 177
5.4 Virtual Network Computing (VNC) 177
5.4.1 VNC concept 177
5.4.2 VNC installation and operation 177
5.5 OpenOffice 179
5.5.1 Installing OpenOffice 180

Chapter 6. Apache HTTP server on Linux . . . 183
6.1 OS/400 HTTP Server (powered by Apache) . . . 184
6.2 Apache modules . . . 184
6.3 Configuration . . . 184
6.3.1 Default Web page . . . 185
6.4 Operation . . . 185
6.4.1 Access control . . . 185
6.4.2 Virtual named hosts . . . 185
6.5 Dynamic content Web pages . . . 186
6.5.1 CGI programs . . . 187
6.5.2 PHP . . . 187
6.5.3 JavaServer Pages . . . 188
6.6 Problem determination . . . 188

Chapter 7. Tomcat Web Application Server (using Java) on Linux . . . 189
7.1 Introduction . . . 190
7.1.1 What Tomcat is . . . 190
7.1.2 License . . . 190
7.2 Installation and operation . . . 190
7.2.1 Installing a Java Virtual Machine . . . 190
7.2.2 Acquiring and installing Tomcat . . . 191
7.2.3 Starting and stopping Tomcat . . . 191
7.2.4 Log files and solving problems . . . 192
7.2.5 Changing the configuration . . . 193
7.2.6 Default home page . . . 193
7.3 Difference with IBM WebSphere Application Server . . . 194

Chapter 8. Firewall on iSeries Linux . . . 195
8.1 Understanding the concept of a firewall . . . 196
8.1.1 References . . . 196
8.2 Bringing Linux to the iSeries server . . . 197
8.3 Firewall on Linux . . . 197
8.3.1 iptables . . . 197
8.3.2 ipchains . . . 198
8.3.3 ipfwadm . . . 198
8.3.4 The kernel . . . 198
8.4 Netfilter filtering basics . . . 198
8.5 Performance . . . 199
8.6 Linux on iSeries firewall strategies . . . 199
8.6.1 Native LAN adapter requirement . . . 200
8.6.2 Basic configuration . . . 200
8.6.3 Perimeter network for logical partitions . . . 201
8.6.4 Perimeter network for other hosts . . . 201
8.6.5 OS/400 partition under control of firewall with a perimeter network . . . 202
8.6.6 OS/400 partition under control of firewall without a perimeter network . . . 203
8.7 Hosted or non-hosted partitions . . . 204
8.8 The iptables example . . . 204

Chapter 9. ssh and Telnet . . . 209
9.1 Remote login . . . 210
9.2 ssh (secure shell) . . . 210
9.2.1 Features . . . 211
9.2.2 How it works . . . 211
9.2.3 Creating a key . . . 213

9.2.4 Placing a key somewhere . . . 214
9.2.5 Using a key to login . . . 215
9.2.6 Using a key in conjunction with the ssh-agent . . . 215
9.2.7 X-forwarding . . . 217
9.2.8 ssh clients for Windows . . . 217
9.2.9 Using the server . . . 219
9.3 scp (secure copy) . . . 219
9.4 sftp (secure FTP) . . . 220
9.5 Telnet . . . 221
9.5.1 Security issues with Telnet . . . 221
9.5.2 Telnet client software . . . 221
9.6 Problem determination . . . 222

Chapter 10. FTP servers on Linux . . . 223
10.1 FTP: The protocol . . . 224
10.2 Security considerations . . . 225
10.3 Available FTP servers . . . 225
10.3.1 The BSD-derived FTPD . . . 225
10.3.2 WU-FTPD . . . 226
10.3.3 ProFTPD . . . 226
10.3.4 vsftpd . . . 226
10.3.5 Summary comparison . . . 227
10.4 Configuration files . . . 227
10.5 Default setup in different Linux distributions . . . 228
10.6 Testing your server . . . 228
10.7 The OS/400 FTP daemon: Transfer tips . . . 228
10.8 Problem determination . . . 229

Chapter 11. Samba . . . 231
11.1 Understanding Samba . . . 232
11.2 Preparing to use Samba . . . 232
11.2.1 Using Linux as a Samba client . . . 233
11.2.2 Connecting to the iSeries NetServer . . . 233
11.2.3 General things you should know about the Samba server . . . 234
11.3 Configuring the Samba server . . . 234
11.3.1 Configuring Samba for directory shares . . . 238
11.3.2 Connecting to the Samba server . . . 239
11.3.3 Samba Web Administration Tool (SWAT) . . . 240
11.4 Problem determination . . . 242

Chapter 12. Network File System . . . 243
12.1 What NFS is . . . 244
12.1.1 The history of NFS . . . 244
12.1.2 Other resources for NFS . . . 244
12.1.3 Two daemons: knfsd and nfsd . . . 244
12.2 Advantages of NFS . . . 244
12.3 Security issues with NFS . . . 245
12.4 User IDs . . . 245
12.5 Installing NFS servers in Linux . . . 246
12.6 Configuring NFS . . . 247
12.6.1 Linux as an NFS server . . . 247
12.6.2 iSeries as an NFS client . . . 248
12.6.3 iSeries NFS server . . . 249
12.6.4 Linux NFS client . . . 255

12.6.5 Diagnostics . . . 255
12.7 Problem determination . . . 256

Chapter 13. Linux print support . . . 257
13.1 What LPR/LPD is . . . 258
13.1.1 Using LPR/LPD for printing . . . 258
13.1.2 Configuring the /etc/printcap.local file manually . . . 258
13.2 LPRng . . . 259
13.3 Common UNIX Printing System . . . 260
13.4 What to use . . . 260

Chapter 14. E-mail systems on Linux . . . 261
14.1 E-mail basics . . . 262
14.2 How e-mail works . . . 263
14.3 The whole picture . . . 263
14.3.1 Mail store directories . . . 263
14.3.2 Mail transfer agents . . . 263
14.3.3 Retrieving messages . . . 264
14.3.4 POP3 and IMAP4 servers . . . 265
14.3.5 Fetchmail . . . 265
14.3.6 Mail user agents (e-mail clients) . . . 265
14.3.7 Mailing list software . . . 266
14.4 Problem determination . . . 266

Chapter 15. System time (NTP) and job scheduling (cron and at) . . . 267
15.1 Accuracy of time in iSeries Linux . . . 268
15.1.1 Determining the problem . . . 268
15.1.2 System clock and hardware clock . . . 268
15.2 Network Time Protocol (NTP) . . . 269
15.2.1 ntpdate . . . 269
15.2.2 xntpd . . . 269
15.2.3 ntptrace . . . 270
15.2.4 ntpq . . . 270
15.2.5 Integration with OS/400 . . . 271
15.3 cron . . . 271
15.3.1 Updating the hardware clock every hour . . . 272
15.4 The at daemon . . . 272

Chapter 16. Help information . . . 273
16.1 Online help . . . 274
16.1.1 Manual (man) pages . . . 274
16.1.2 info . . . 275
16.1.3 help . . . 275
16.2 Additional information . . . 275
16.2.1 Howto . . . 275
16.2.2 RFC . . . 276
16.3 The Internet . . . 276
16.4 Books . . . 277
16.4.1 IBM Redbooks . . . 277
16.5 Useful tools . . . 277
16.5.1 Finding a file with find and locate . . . 277
16.5.2 Determining the type of commands . . . 278
16.6 Other resources . . . 278

Appendix A. Application scenarios 279
Base scenario 280
All-in-one-box solution 280
Intranet application server 281
Windows NT domain controller 281
Web server with Apache and Hypertext Preprocessor (PHP) 281
Magic eMerchant 282
Linux cluster for availability 282

Appendix B. ODBC 285
What is ODBC 286
Linux on iSeries implementation 286

Appendix C. Hardware planning, ordering, and configuration examples 289
Hardware planning 290
Supported machines and processors 290
Virtual I/O 292
Native I/O 292
LPAR planning 293
LPAR planning process 293
LPAR Validation Tool 294
Configuration example 296
Planning a new system 296
Ordering 299
Hardware ordering 299
Software ordering 300
Getting help 300

Related publications 301
IBM Redbooks 301
Other resources 301
Referenced Web sites 301
How to get IBM Redbooks 302
IBM Redbooks collections 302

Index 303

Special notices

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

IBM trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

e (logo)® 
AFS®
AIX®
APPN®
AS/400®
BookMaster®
DB2®
DB2 Universal Database™
DFS™
Home Director™
IBM®
iSeries™
Netfinity®
Network Station™
OS/2®
OS/400®
Redbooks Logo 
Perform™
PowerPC®
pSeries™
Redbooks™
S/390®
SP™
WebSphere®
xSeries™
zSeries™
Lotus®
Approach®
Lotus Notes®
Notes®
Domino™

Other company trademarks

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere.,The Power To Manage., Anything. Anywhere.,TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Preface

Running Linux on the IBM @server iSeries server combines the strengths of Linux and OS/400 for an integrated solution. Linux delivers excellent open source solutions, while OS/400 is a premier integrated platform for business solutions. Linux enables a new stream of e-business applications for the iSeries platform that complements its strength as an integrated core business solution. Linux applications benefit from the iSeries platform's ability to provide resource flexibility, security, reliability, and connectivity to other applications on a single server.

This IBM Redbook begins with an overview of Linux, defines what open source means, and explains why using Linux on iSeries is beneficial. Then, it highlights how to install and use Linux on the iSeries server. It discusses the basic system administration tasks and Linux application development to help you manage your system and develop Linux applications on the iSeries server. It also introduces a wide range of services, such as Firewall, Apache, Samba, and e-mail, and explains the capabilities of each.

This redbook is intended to help beginner and intermediate Linux users, with an OS/400 background, to implement Linux on the iSeries server.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Rochester Center.

Sungsim Park is ITSO Specialist at the International Technical Support Organization, Rochester Center. She has over 15 years of experience in working with S/36, S/38, and iSeries servers. Before joining the ITSO in 2000, she taught IBM classes on all areas of iSeries as a senior education specialist in IBM Korea and she provided technical marketing support to AS/400 sales representatives, IBM Business Partners, and customers. Her areas of expertise include server consolidation and application development. She can be reached via e-mail at: sungsim@us.ibm.com

Klaas Bals is a Project Leader and Technical Designer for Inventive Designers, an IBM Business Partner for AS/400 and Lotus. Inventive Designers (http://www.inventivedesigners.com) is an Advanced AS/400 Development Partner that specializes in the development of such e-business tools as EverGreen/400 and DTM for AS/400e servers. He has done development on the iSeries server mainly in C, C++, and Java using such technologies as XML and XSL. He has over five years of experience in Linux, including the advanced development and deployment of Linux in a networking environment. Klaas is the webmaster for the Linux on iSeries Web site at http://www.iSeriesLinux.com. You can reach him by e-mail at: Klaas_Bals@inventivedesigners.com

Michel Fleury is a AS/400 IT Specialist in Denmark. He has over 15 years of experience in computing, with eight years in the AS/400 area. He has worked at IBM for six years with second-level defect support on Client Access, OS/400, and FSIOP, Integrated PC server (IPCS), Integrated Netfinity Server (INS), and Integrated xSeries Server for iSeries. His areas of expertise include network security, e-business solutions/applications, and client connectivity. He has written extensively on the Linux firewall, Apache, and FTP server. He can be contacted by e-mail at: michel@dk.ibm.com

Gary Hansen is an Advisory Software Engineer. He has over 15 years experience with software support for the AS/400 and iSeries. He has worked with many of the Client Access products for the OS/400 and has spent the last several years supporting the FSIOP, Integrated PC Server, Integrated Netfinity Server, and the Integrated xSeries Server for iSeries. He has written about and taught various classes on the integration products for OS/400. His areas of expertise are in client connectivity, networking, and OS/400 integration. You can reach him by e-mail at: hansengl@us.ibm.com

Gene Meyer is a Senior IT Specialist with iSeries Techline in Dallas, Texas. He has 35 years experience with data processing. He has 20 years with IBM working with VSE, System/34, System/36, System/38, AS/400, and iSeries. He holds a degree in Computer Science from Texas A & M Commerce and an MBA from the University of Texas Pan American. His areas of expertise include iSeries connectivity, programming, Domino, and hardware configuration. You can reach him via e-mail at: gmeyer1@us.ibm.com

Rebecca Mills is an Advisory IT Specialist in Australia. She has seven years of experience in the AS/400 field, including Techline and as a Field Technical Support Specialist. She holds a degree in Business from the University of Technology, Sydney. Her areas of expertise include Domino, logical partitioning (LPAR), and performance. She can be contacted by e-mail at: rmills@au.ibm.com

Peter Poeml is a Linux Specialist for SuSE, a leading Linux distributor. He joined SuSE in 2000. His areas of expertise include Linux development, package maintenance, technical support (especially on PPC platforms), system administration, and networking. He did the first port of the SuSE Linux distribution to the iSeries platform. You can reach him by e-mail at: poeml@suse.de

Tim Robison is an Instructor at Lake Superior College in Duluth, Minnesota. He has 18 years of experience in the IT field as a programmer and software analyst. His experience ranges from S/34 through iSeries server, Linux, and other forms of UNIX. He also runs a small Internet Service Provider company in North Eastern Minnesota where he uses Novell, Windows, and Linux platforms to provide Internet access to rural communities. You can contact him via e-mail at: trobison@meadows.net

Tadayuki Yoshida is a Software Engineer for IBM Japan. He has worked at IBM for a year, but has over four years of experience in Linux. He holds a master's degree in Computer Science from the Hokkaido University. His areas of expertise include national language support, internationalization of programs, and Linux system administration. He has written extensively on national language support, security issues, and development. He can be contacted at tadayuki@jp.ibm.com

Thanks to the following people from IBM Rochester for their contributions to this project:

Mike Aho
Brent Baude
Theodore Bauer
David Boutcher
Monte Bruesewitz
Jay Bryant
Keith Cooper
Daniel J Degroff
Selwyn Dickey
Erwin Earley
David Engebretsen
Tonya Holt
Steven Janssen
Brian King
Brett Leeser
Larry Loen
Carl Pecinovsky
Jeffrey Scheel
Kay Tate
Dave Wall

Special notice

This publication is intended to help beginner and intermediate Linux users, with an OS/400 background, to implement Linux on iSeries. The information in this publication is not intended as the specification of any programming interfaces. See the PUBLICATIONS section of the IBM Programming Announcement for more information about what publications are considered to be product documentation.

Comments welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at

  `ibm.com/redbooks`

- Send your comments in an Internet note to

  `redbook@us.ibm.com`

- Mail your comments to address on page ii

1

Getting acquainted with Linux on the iSeries server

This chapter provides a brief introduction to Linux and how it works with the iSeries server. It is intended for those who are new to Linux. Users who are familiar with Linux can locate the information they require by searching the Internet. This chapter explains:

- What Linux is
- What open source is
- Why use Linux on the iSeries server
- How Linux is integrated with iSeries
- How Linux can enhance the iSeries platform

1.1 What Linux is

Linux is an operating system that was initially created as a hobby by a young student, Linus Torvalds, at the University of Helsinki in Finland. The project was soon joined by other developers, and it grew to a big open source project. In only a few years, a worldwide community has evolved, derived from programmers who were attracted by the reliability and flexibility of this completely free operating system. Since the initial release, Linux has gone through several updates. The current full-featured version is 2.4 (released January 2001), and through continued development, it is becoming a very robust operating system.

Although Linux is not UNIX, it looks and feels like UNIX. Linux has a large and powerful selection of commands and APIs, many of which are identical or similar to UNIX. Most applications can be ported with little or no effort between the different UNIX variants, including porting between different Linux hardware platforms.

Linux is similar to the UNIX operating systems, but it is *not* UNIX. Linux was developed under the GNU (GNU's Not UNIX) General Public License (GPL) and is associated with "open source" and "free software".

The Linux kernel itself is of little use to the average user. The kernel supplies the interface to the hardware and peripherals and provides basic services, such as managing tasks and scheduling processor cycles to the applications, that operate on Linux as a foundation.

But an operating system is more than a kernel. It needs tools to interact with the kernel and to manage the system. These tools are implemented outside the kernel space, in so-called "userland". This approach keeps the kernel small, fast, and stable. Most of these tools already existed in GNU projects, when the Linux kernel was born, collectively known as *GNU/Runtime System*. When we say "Linux", we refer to the combination of the Linux kernel and the GNU tools, not just the kernel.

Linux is an efficient multi-tasking operating system that can run on a minimal amount of hardware by comparison to today's standards. It can run very well on a 486 processor, for example. It is a multi-user system, and as such, offers various levels of built-in security. It is also a stable system that is capable of sustained up-time. Finally, the term that one most often hears with regard to Linux is "free." Not only is it scalable and becoming more so, but the bulk of the code can be acquired free and comes with the source code which can be modified, at will, within the bounds of the GNU General Public License. See 1.2, "What open source is" on page 3.

You can find a summary of the operating system virtues and statistics on the Web at:
`http://www.db2mag.com`

For more details on Linux in general, see:

- Linux Online offers a large amount of information on available applications, distributions, and locations for downloading the code (for free), documentation, education (including online courses), and information regarding hardware as well as a variety of other information: `http://www.linux.org/apps/index.html`
- The Linux Kernel Archives site is where the kernel sources can be downloaded from, but it also contains links to various other related locations: `http://www.kernel.org/`
- The Linux Installation HOWTO site is sponsored by the "Linux Documentation Project" (LDP). LDP has free documentation and "HOWTO" documents that are short guides to do many of the tasks involved with Linux. This particular link is HOWTO install Linux: `http://www.linuxdoc.org/HOWTO/Installation-HOWTO/`

- Linux International offers a nice description of the technical merits of the operating system at: http://www.li.org

For more information about Linux on PowerPC Processors, see:

- Linux/PPC support site: http://www.penguinppc.org
- Development home for Linux for 64-bit PowerPC Processors: http://linuxppc64.org

For specific Linux on iSeries information, see:

- Linux PPC: http://linuxppc.org
- Linux for IBM @server iSeries: http://www.ibm.com/iseries/linux
- iSeriesLinux.com: http://www.iserieslinux.com

1.2 What open source is

It is difficult to discuss the phenomenal growth of Linux without discussing *open source*. There is a difference between open source and the GNU General Public License as noted on the GNU Web site (http://www.gnu.org). Linux was developed under GNU, which has articulated a philosophy that defines “free code” – the user’s right to use the code – rather than defining what they can’t do, which is the case with proprietary software. This license allows the user to alter, distribute, and even sell the code covered under its license as long as they allow those to whom they distribute the code to do the same.

The General Public License promotes free code on the GNU Web page. It also provides protection for the developer and prevents a user from altering the code and then asserting proprietorship over the code. This does not mean the code can’t be sold. According to the GNU Web site, “free software” allows a user to run, copy, distribute, study, change, and improve the software. It must also be available for commercial use.

The basic principles of GNU are centered around four basic freedoms for the user. The benefits of each freedom may apply to the user or ultimately to the community that shares the program. The freedoms include:

- Running the program for any purpose (Freedom 0)
- Studying the program and adapting it as necessary (Freedom 1)
- Redistributing the program (Freedom 2)
- Improving the program and releasing the improvements (Freedom 3)

For more information, go to: http://www.gnu.org/philosophy/free-sw.html

There is a distinction between “free software” and “open source” software. The GNU Web site (see the link above) describes the differences, but prefers the term “free software”. Protection of the free software is implemented by means of the “copyleft” agreement. Copyleft basically means that any user who redistributes the program must also pass the same freedoms for using it onto the users.

The software can, in fact, be acquired without cost. It can be altered and distributed by anyone who can and who so desires, as long as they extend the same freedom to the rest of the world. The same is true for much of the documentation that is associated with Linux.

At first glance, this seems to be a radical and unworkable way to develop and support code. However, it can be viewed as the manifestation of an evolutionary process that has been occurring for some time with internetworking between developers and programmers, but which had remained unheeded by the general public until recently. Since the early days of networking, where individuals would share code and make changes and then pass the improvements around, there has been an ongoing effort to establish standards that promote common interfaces that are nonproprietary.

The purpose of the standards are to enable *open system interfaces* and *plug n' play interfaces* to communicate with various heterogeneous systems. Also, standards help to alleviate the various difficulties encountered when attempting to communicate across diverse vendor hardware and software platforms. Open source can be viewed as a manifestation of this process and the process itself as a necessity for extending the development of internetworking.

The Linux "suite" of applications has been developed and supported via the Internet through the collaboration of diverse individuals and groups. You might suppose that this apparently random association would result in incomplete and unsupportable software. In actuality, it seems that when there is a need for new software or hardware support or a defect is discovered and reported, the new or repaired software can be swiftly done by the user who required the changes, or by the author directly.

These alterations can occur without the necessity of design review, cost analysis, and other considerations that a centralized development structure requires due to the availability of the source code. IBM has recognized the efficacy of this community. IBM sees the benefit of the rapid and innovative development of robust and stable code to provide the enabling layer for e-business applications. For additional information on IBM's interest and investment in Linux, see the following resources:

- *Linux for S/390*, SG24-4987
- *Linux on IBM Netfinity Servers: A Collection of Papers*, SG24-5994
- IBM software on Linux - Ready for business:
  http://www-4.ibm.com/software/is/mp/linux/

The outcome of this process is that Linux is rapidly filling in gaps that existed when it was primarily used by educational and development communities. Linux has been gaining favor as a server platform for various Internet functions including firewalls. Furthermore, it is starting to acquire more polished user interfaces. For example, KDE and GNOME provide a window environment that is much more useful to a desktop environment than the command line interface that was favored in the past. There is still a void of high profile applications, but this is also being rectified as more and more vendors recognize Linux as a maturing operating system.

Another result of the evolutionary development of Linux is that various pieces of the code are located on various Web sites. Because of this, it was difficult to install and upgrade the product. Another difficulty was keeping track of module dependencies and the need to acquire drivers for the hardware. The response to these issues is *distributions*. Distributions provide coherent bundles of software in one location that will furnish a variety of capabilities needed for a usable server or desktop. Generally they provide a lot of the same things, but there may be particular functions that you require that are not available with a given distributor.

1.2.1 Distributions

The user interface and applications have developed separately from the Linux kernel and have a variety of distributors. Most of them may be acquired without charge. If the user had to separately download the kernel and all of the applications that they need, it would be a daunting task. However, it *can be* done and *is* done by knowledgeable users who know exactly what they want. If you are just beginning or don't have time to download separate packages, several groups have taken it upon themselves to put together collections of applications with a version of the kernel which can be installed as a whole.

These are referred to as *distributions*. There are several mainstream Linux distributors. *SuSE*, *Turbolinux*, and *Red Hat* are the distributors that are currently working with the iSeries to distribute the drivers necessary to run Linux in an iSeries partition. There are many other notable distributions available. But, at this time, it is unknown which might provide the PowerPC extensions required for the iSeries hardware.

In most cases, the distributions have FTP sites from which the user can download the code. However, some of the distributions are very large and the download time can be lengthy. For a modest fee, the distributions can generally be acquired on CD and most come with documentation. Many of the distributors have added utilities to facilitate installation and maintenance of the product. Among these are GUI packages that manage the code installation, upgrades, and configuration utilities. It is worth some research before selecting a particular distribution. Required features may not be available. Relevant to the iSeries implementation is whether the distribution provides a PowerPC distribution with iSeries extensions.

1.2.2 Linux on all platforms

Linux was originally written to run on the Intel processor, but now runs on Compaq Alpha AXP, Sun SPARC and UltraSPARC, Motorola 680x0, PowerPC (including IBM @server pSeries, iSeries, and Macintosh), ESA/390 architectures, the newer 64-bit z/Architecture, ARM, Hitachi SuperH, MIPS, HP PA-RISC, Intel IA-64, DEC VAX, and others. With support for these processors, Linux runs on all IBM @server platforms (xSeries, iSeries, pSeries, and zSeries). The latest supported version of the kernel at the moment is 2.4. You can acquire this version from the Web at: `http://www.kernel.org`

There are several different versions of the kernel available. Most users will simply use the kernel that comes with their distribution. If they need to update the kernel, they are usually told to obtain a particular level and how to obtain the required iSeries extensions. Those who are building their own kernels should know that iSeries requires a 2.4 level kernel (not 2.2 or earlier) and the earliest distributions use at least 2.4.3 or later.

Important: The Linux on iSeries implementation is a PowerPC based distribution with a kernel compiled for iSeries.

1.3 Why use Linux on the iSeries server

With an integrated operating system, why would an iSeries shop look at an open system? With the growth of the Windows operating system in the 1990s, iSeries developers integrated the Windows operating system into OS/400 by providing drivers to allow for ease of interaction between the two operating systems. IBM announced that it will spend $1 billion on the development of Linux. This includes their commitment to making the Linux operating system run on all IBM @server platforms. And as a result, iSeries developers are providing this integration for Linux.

IBM CEO Lou Gerstner states, “IBM is making a commitment to Linux because IBM is convinced that Linux can do for business applications, what the Internet did for networking and communication.” That means extend opportunities for open access and expansion.

Irving Wladawsky Berger, IBM vice president of technology and strategy, pointed out, “We see Linux as a major force in IT and moving IT to the next generation of the Internet business.” Further, Berger said, “So we are supporting Linux across all of our hardware platforms, our middleware, and our services business.”

It is these commitments, and the iSeries-specific benefits, that make Linux on iSeries a viable product.

The iSeries provides the following benefits to the Linux customer:

- A hardware platform of proven reliability and stability
- The ability to deploy less than a whole CPU to a partition for simple server functions, yet retain the ability to partition servers through shared processors
- Virtual disk, enabling RAID-5 and disk striping to be deployed without extra effort
- Virtual LAN, which enables a Linux partition to communicate with OS/400 or another iSeries efficiently, but without expensive gigabit hardware
- The ability to “right size” disk storage to what is needed using virtual disk, rather than to the size of the device
- The ability to share OS/400 hardware that provides access to large amounts of DASD and centralized management of resources
- Server consolidation
- Potentially initiating interaction with a large portfolio of existing e-business and business applications
- A new customer base to Linux

The Linux environment offers the iSeries platform the following benefits:

- Involvement in the open source community
- New possibilities for Web applications and components
- Leverage IBM’s investment in Linux
- Expand its Internet flexibility and scope, as the ability to integrate with Linux expands
- Potentially gain access to new hardware through Linux drivers
- Attract new customers to the iSeries server

1.4 Linux on iSeries implementation

The remaining sections briefly survey the iSeries implementation, look at some of the ways the two environments can complement each other, and discuss how Linux might be used to enhance the iSeries environment.

1.4.1 What is possible

The IBM @server iSeries server provides the ability to run Linux on the iSeries server in a secondary partition. The primary partition must be running OS/400 V5R1 or later, which provides the support required to boot the kernel in a secondary partition. The Linux operating system needs to be installed and executed in a secondary iSeries partition. In the first release, the integration with OS/400 was minimal. Over time, the integration between OS/400 and Linux will be enhanced.

For those who are familiar with the Integrated xSeries Server for iSeries (formerly the Integrated Netfinity Server and Integrated PC Server), the most basic support of Linux resembles IBM's effort to provide affinity to the Microsoft Windows operating systems. The most notable difference is that Linux runs directly on the iSeries PowerPC hardware, rather than on an integrated adapter or external xSeries running on an Intel processor.

Attention: To run Linux on iSeries hardware, you must have OS/400 Version 5 or later installed in the primary partition. Linux is not enabled to run as the sole operating system on the iSeries server.

1.4.2 Definitions

Here are some common terms you should understand when working with Linux on the iSeries server:

- **Logical partition**: Logical partitions (LPAR) divide the iSeries hardware resources into multiple independent servers within a single iSeries server. Each partition has access to processors, memory and disks, and its own copy of OS. It is similar to a user who partitions a hard drive on a PC to run separate operating systems or to isolate data. Unlike the PC, however, multiple partitions can run different operating systems simultaneously. For more information on logical partitioning, see:
  http://www-1.ibm.com/servers/eserver/iseries/lpar
- **Primary partition**: There is only one primary partition on an iSeries machine. It is the primary partition that manages the other partitions and enables the guest partitions. For Linux partitions, OS/400 must be at V5R1 or higher on the primary partition. This partition has sole control of the physical service panel and the ultimate control over the machine. An IPL of this partition either IPLs the other partitions or ends them depending on how they are configured.
- **Secondary partition**: Secondary partitions are defined as partitions, but don't include the primary partition. These partitions run other copies of the OS/400 operating system. Each is an independent entity and has a separate instance of OS/400 and System Internal License Code (SLIC) running in it. The operating system and PTF level could be different than that in the primary or other secondary partitions within the limits allowed by LPAR. Secondary partitions have a virtual service panel that is accessed via DST or SST.
- **Guest partition**: A guest partition is a unique secondary partition that enables a *guest* operating system to run *natively* on the iSeries hardware independently of OS/400. It is not possible to configure a guest partition on the iSeries hardware without OS/400 version V5R1 or later installed in the primary partition. Currently only Linux is enabled in a guest partition. This partition does not have OS/400 or System Licensed Internal Code (SLIC) installed in it.
- **Hosted partition**: A hosted partition has the root file system on the virtual disk. A network server description (NWSD) is required to IPL this partition and provide access to virtual services. In addition, you need to create at least one network server storage space (NWSSTG) and link it to the NWSD for the virtual disk or disks. The only resources required for Linux in this situation are the minimum amount of main storage required for any secondary and a portion of processor resource. The hosting partition does not have to be the primary partition. It is possible to use native IOA and their devices directly as well as the virtual services in a hosted partition.
- **Non-hosted partition**: A non-hosted partition has the root file system on the native disk. Linux controls the DASD and can control the LAN Adapters, CD-ROM or DVD-RAM, and tape drive. The partition is IPLed from the Virtual Service Panel or by NWSD. Since Linux is in control of the adapters, it is the main source of diagnostic messages. This does not

include the low level partition management functions which remain with the primary partition as they do for all secondary and guest partitions.

- **Direct attached IOA**: This is also referred to as native IOA. These are adapters for communications, DASD, and tape devices that have been removed from owning IOPs, are assigned to a guest partition, and run directly under the control of Linux. They are no longer associated with an owning IOP when they are removed. When they are allocated to the guest partition, the adapters can be viewed under the “Work with system partitions” menu in the service tools. Figure 1-1 shows a native I/O listed with the proper device name under the Linux controlled partition.

```
                      Display Allocated I/O Resources
                                                          System:   AS03
System partition(s) to display . . . .  *ALL  *ALL, 0-1
Level of detail to display . . . . . .  *ALL  *ALL, *BUS, *IOP, *IOA, *DEV

Par                                                Serial      Part
ID   Description                    Type-Model     Number      Number
         Communications Chann       605A-001      00-1072020
     System Bus           23        283B-         38-0301080   04N4723
      Communications IOP             2890-001      10-81042    0000023L4306
       Communications IOA            2890-001      10-81042    0000023L4306
        Communications Port          2744-001      10-1010042  0000023L4288
        Communications Port          2744-001      10-82017    0000023L4288
        Virtual Port                 6B00-001      10-81042    000008193654
     System Bus           24        283B-         38-0301080   04N4723
  1  LINUX    Partition               9406-270      10-5HYMM
     System Bus           23        283B-         38-0301080   04N4723
      Multiple Function IOA          2763-
     System Bus           24        283B-         38-0301080   04N4723
                                                                      More...
* Indicates load source.
F3=Exit   F5=Refresh          F6=Print   F9=Toggle empty pos
F10=Display logical address   F11=Display partition status   F12=Cancel
```

Figure 1-1 Displaying adapters allocated to a guest partition

- **Hypervisor**: This is the code that the primary partition uses to supervise the partitions. The hypervisor code mediates between the partitions to ensure partition integrity.
- **NWSD**: This stands for network server description. This is an OS/400 configuration object on a hosting partition. It was inherited from the Integrated xSeries Server for iSeries. It is used in both the hosted and non-hosted environment to load the kernel for the initial installation of Linux. In the “hosted” environment, it is used to IPL the partition, boot Linux, and must be active for the server to function.
- **Virtual I/O**: This is I/O support that is provided for the guest partition. CD-ROMs, tape drives, and DASD are provided by a hosting partition through the use of an NWSD. Virtual LAN and virtual console are provided by lower layers of code.

Summary of guest partition definitions

The key point to take from the definitions is that there are essentially two types of guest partitions: hosted and non-hosted. The options available to the partition depend on how the partition is configured. As it was noted in the definitions, a hosted partition does not preclude utilizing IOA directly attached to Linux, but a non-hosted partition excludes the use of virtual DASD, tape, or CD-ROM, since it does not have a hosting partition to provide these functions. The virtual console and virtual LAN are available to both types of partition. Table 1-1 presents the options that are available.

Attention: It is possible to have a hosted partition and use both virtual and native (directly attached) I/O. It is not possible to have a non-hosted partition and use virtual DASD, virtual tape, or virtual optical. The virtual console and virtual LAN are available to both environments.

Table 1-1 Comparison between hosted and non-hosted Linux partitions

Resource	Hosted	Non-hosted
Boot disk	*NWSSTG, STMF, via IOA, A or B slot	Via IOA, A or B slot
Other DASD	*NWSSTG or via IOA	Via IOA
Tape	Virtual or via IOA*	Via IOA*
CD-ROM, DVD-RAM	Virtual or via IOA*	Via IOA*
Ethernet LAN	Virtual or via IOA	Via IOA
Console	Telnet to port 2301 on a host partition	Telnet to port 2301 on a primary partition
IOA*: CD-ROM, DVD-RAM, and tape IOAs support only internal drives for Linux.		

You also need to consider the items listed in Table 1-2 and Table 1-3 when choosing hosted or non-hosted partitions. Both methods have positive and negative reasons for their selection of implementation.

Table 1-2 Hosted Linux partition (root file system on Virtual DASD)

Positives	Negatives
Always boots by varying on a NWSD	Cannot be active if the hosting system is in restricted state*
Can be saved via iSeries SAV command to obtain a snap shot of whole Linux partition	
Can be copied easily to create a second Linux partition	
Always has access to virtual tape/CD from the hosting partition	
Uses the iSeries disk protection scheme (RAID or Mirroring)	
Disk sizes can be created from 1MB to 64 GB. Not limited to 8 GB or 17 GB increments like native disks	

Positives	Negatives
***Restricted state**: A state in OS/400 where most OS/400 services are shut down to enable certain forms of machine service and upgrade to take place. If the hosted partition is in this state, its Linux partitions cannot be active.	

Table 1-3 Non-hosted Linux partition (root file system on Native DASD)

Positives	Negatives
Can be booted by varying on a NWSD or from the LPAR screens	Cannot be saved via iSeries SAV command
Can gain access to virtual tape/CD from a hosting partition if booted by varying on NWSD	Cannot be copied easily to create a second Linux partition
Can be active if all iSeries partitions are in a restricted condition if the Linux partition is booted from the LPAR screens (although console will be inaccessible)	Cannot use hardware disk protection (at first release)
	Wastes valuable rack space since the whole DASD cage (entire IOA) has to be assigned to the partition; a more expensive solution
	Granular to whole disk units (8 GB or 17 GB). Use of Linux Logical disk Volume Manager (LVM) or equivalent is required to make larger drives
	If hardware disk protection is available, it will take multiple physical drives to achieve protection, taking possibly far more space than required

1.4.3 Environment

The secondary partition that Linux will be installed in should be set up differently than other secondary partitions. This can be inferred by the fact that it must be defined as a *guest partition*. It is not specifically a *Linux* partition, but rather a guest partition. The intent is to allow a guest operating system to install on the iSeries platform and to operate independently of OS/400. This, of course, implies that OS/400 is installed in another partition to provide the guest function.

The chief differences between the iSeries implementation and that of a standalone Netfinity server, Compaq server, or other computer are the installation and the fact that the bulk of these other implementations are on Intel processors. Installation is discussed in more detail in the Chapter 2, “Installation and operation of Linux on the iSeries server” on page 17.

1.4.4 Shared processors

If the system has selected processors, it will be possible for a single processor to be split among four partitions; the processor is shared between those partitions in amounts specified by the user with minimum increments of 1% (.01) of a processor and a minimum of 10% (.1) per partition. This has the advantage of getting more function for the processor investment in cases where Linux CPU requirements are low. For more information, see Appendix C, “Hardware planning, ordering, and configuration examples” on page 289.

1.4.5 Virtual I/O

Creating a Linux partition only requires processor and memory resources be allocated to the partition. The remaining resources such as storage, tape, CD-ROM can be supplied via a *virtual I/O* from a hosting OS/400 partition. Virtual I/O requires that a hosting partition is configured to supply the resources to the hosted partition. Virtual I/O is discussed in more detail in 2.5, “Virtual I/O” on page 51.

There are five virtual resources available that can be divided into two groups. Two of the services do not require the host OS/400 partition to be active. They can be active when the system is up to DST. However, the primary partition must be up and active. These services are:

- **Virtual LAN**: It is desirable to have communication between partitions that are concurrently active. The mechanism for communicating between secondary partitions running OS/400 has been *Virtual OptiConnect*. The guest partitions do not use this mechanism because Linux does not support Virtual OptiConnect. Therefore, another mechanism was constructed, which is referred to as *virtual LAN*. It is one of several virtual services available to the guest partition. The configuration of the virtual LAN is covered in 2.5.1, “Virtual LAN configuration” on page 52. It is a built-in, high speed mechanism for directly transferring data between partitions without LAN adapters or other hardware. Sixteen virtual LANs can be configured from the LPAR configuration screens. The virtual LAN itself is not an actual physical connection, but emulates a 1 Gb Ethernet LAN.
- **Virtual console**: The virtual console provides a way to interact with Linux when it is being installed, or when TCP/IP has been configured wrong so that there is no access from the LAN. It can be accessed from Telnet client on the network by connecting to port 2301 of Linux hosting partition.

> **Important:** The primary partition must be active.

The other three services require OS/400 on the hosting partition to be active and a network server description (NWSD) to be active. These are all similar to the services provided by the Integrated xSeries Server for iSeries:

- **Virtual DASD**: These are referred to as network server storage (NWSSTG). Each one appears to a server as a single hard drive, which is then known as “virtual disk”. They can be used to store the install image or the boot image, or configured to be additional user file systems. They can be as large as 64 GB and as many as 48 of them can be attached to a “guest” partition.
- **Virtual tape**: The OS/400 tape drive can be used by Linux for Linux-based save/restore of files and directories in a hosted partition. This is referred to as *file-level backup*. Refer to Chapter 3, “Linux administration” on page 87, for more information.
- **Virtual CD-ROM**: The OS/400 CD-ROM may be shared and used for a Linux installation to load other packages or for normal CD-ROM use.

Figure 1-2 illustrates how virtual devices work. They communicate via hypervisor code to the device mapping functions of the host (OS/400) partition that directs the messages to or from the correct device. For instance, a Telnet request to the Linux server is directed through the LAN adapter on the hosting OS/400 partition and routed by the OS/400 IP forwarding function through the internal LAN. Linux drivers pick up these messages on the virtual LAN and send them to the destination application. This is similar to the integration functions that allow the Windows NT server running on the Integrated xSeries Server for iSeries hardware to use the tape and optical drives on the OS/400. Note the console mapper that enables the virtual console to connect to the guest partition.

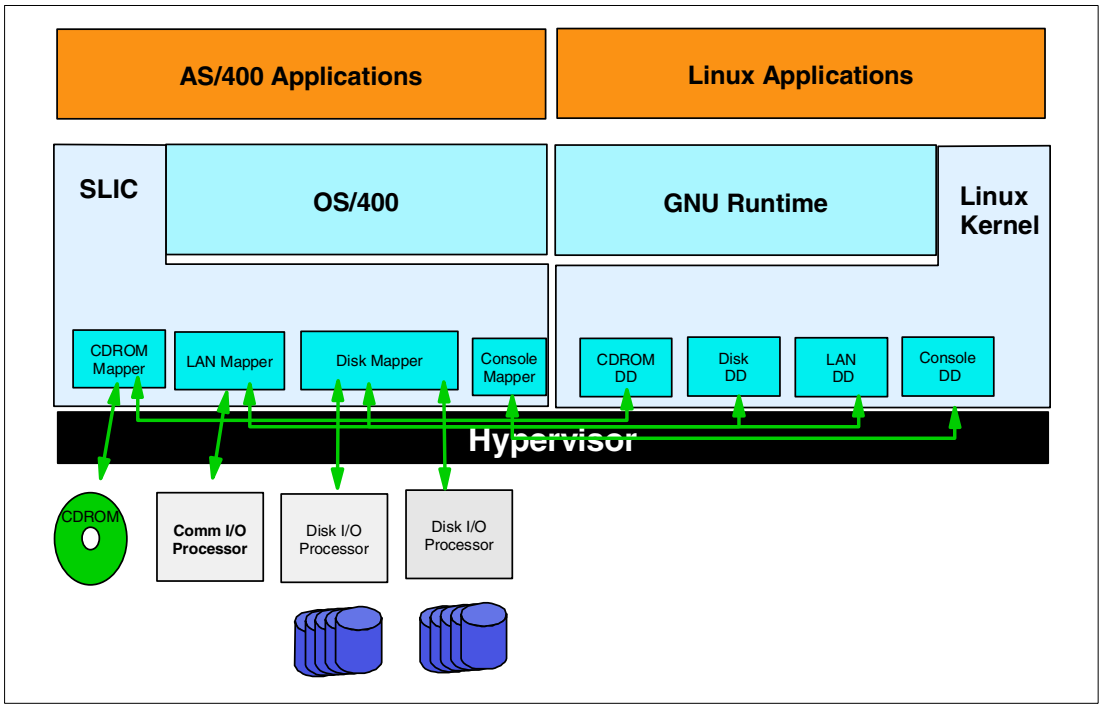


Figure 1-2 Overview of virtual I/O

1.4.6 Native I/O: IOAs directly attached to the Linux partition

Alternatively, the user may want to achieve minimal dependence on the OS/400 and maximum separation of resources so that all I/O to the Linux partition is under control of the Linux operating system and the server boots independently of the other partitions. They can configure a non-hosted guest partition by allocating DASD, CD-ROM, DVD-RAM, tape drive, and LAN adapters as direct attached IOAs controlled by the Linux operating system. The supported tape, CD-ROM, and DVD-RAM are the internal quarter-inch tape, internal CD-ROM, and DVD-RAM in a 5074 or 5079 expansion tower. The initial installation requires assistance from the OS/400 partition to load the kernel and start the install. After Linux is installed, the kernel can be stored in such a way as to exclude hosted assistance. It can then be IPLed from its own virtual service panel.

Attention: Although Linux controls its own I/O, the partition still relies on the primary partition and hypervisor for low-level services. If the primary partition is re-IPLed for example, the guest partitions and secondary partitions will likewise be brought down or IPLed depending on the way they were defined. Also, a failure in the primary partition that causes it to abnormally end will likewise terminate the guest partitions. Re-IPLing the primary partition will cause active Linux partitions to react on the subsequent boot as if they had a power failure type crash. If possible, terminate the Linux partitions normally (for example, halt or shutdown) before you re-IPL the primary or hosting partitions.

Supported native I/O features

The supported native I/O features are detailed in Table C-5 on page 292. The SCSI controller actually provides a PCI interface so that the Linux drivers can locate and communicate to the iSeries hardware.

NWSD for non-hosted partitions

To install the Linux partition, you must have a network server description. Once the installation completes, the partition can be configured to boot independently. Refer to 2.6.4, “Configuring for a non-hosted partition” on page 81, for more information.

The non-hosted partition is an example in which all I/O would be driven by Linux and the OS/400 is only responsible for protecting the integrity of the partition’s resources from other partitions. The non-hosted configuration is best suited to serve as a firewall or similar environment. However, a more likely scenario that will appeal to many is to use the directly attached communications adapters rather than routing the communications across the virtual LAN and out an adapter on the hosting partition while using the virtual DASD and virtual optical devices on the hosted partition. Other mixed environment combinations are possible, but we do not discuss them here. Figure 1-3 shows a direct attached IOA that communicates with the Linux device drivers.

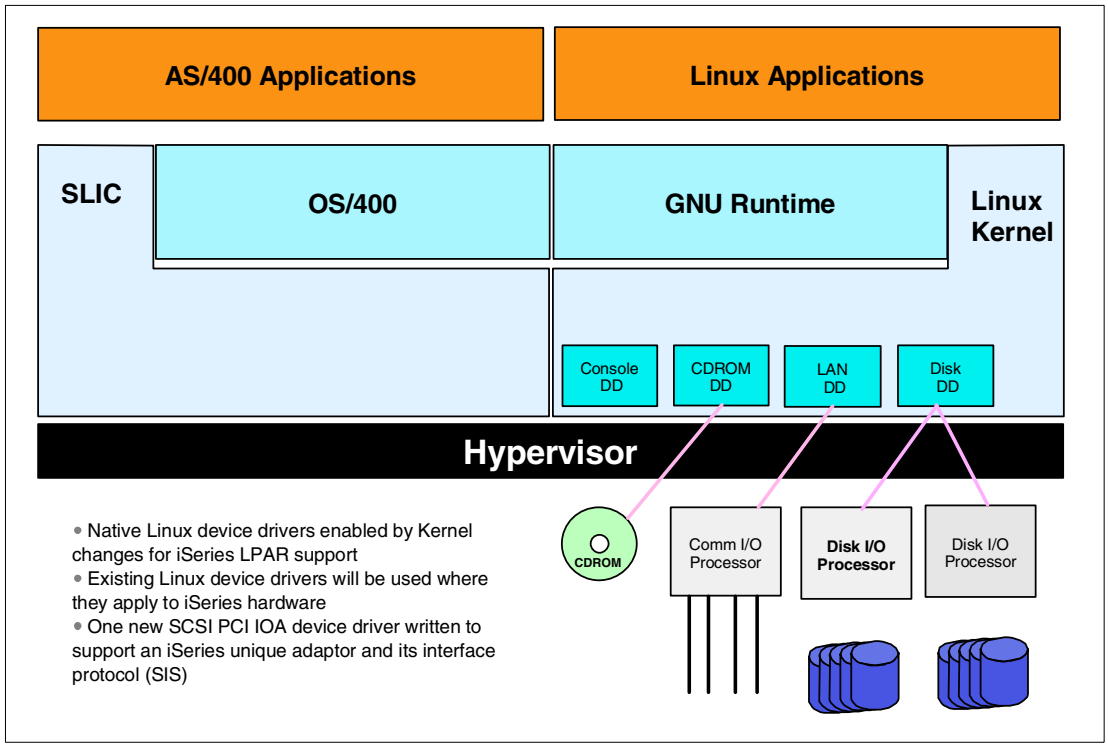


Figure 1-3 Direct attached IOA

1.5 Linux application scenarios on iSeries

What are the advantages and disadvantages of the Linux environment on iSeries?

The “i” in iSeries represents integration. It currently supports the following environments:

- OS/400
- Intel processors on Integrated xSeries Server for iSeries Servers
- Domino
- AIX applications in the PASE environment

And now it supports Linux.

Although the first release is aimed basically at installing a fully functional Linux server and running it in a secondary partition, there are still some scenarios that might be immediately appealing. Figure 1-4 illustrates a feasible scenario that would use two Linux firewall partitions to establish a perimeter network all within the same system.

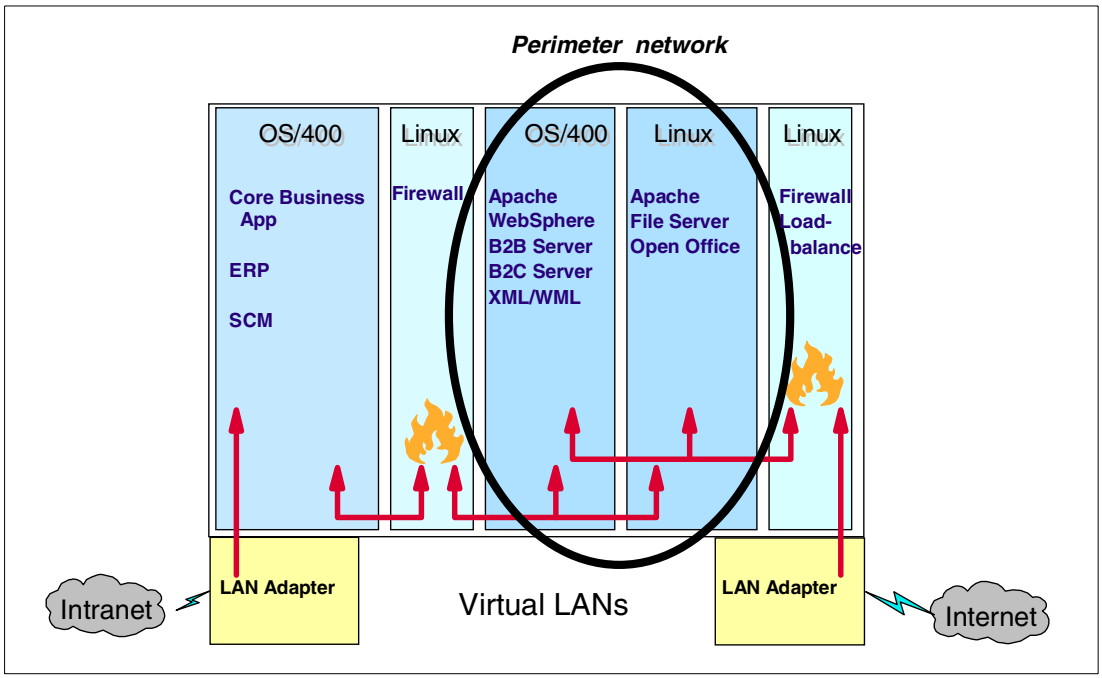


Figure 1-4 Scenario using Linux partitions to create a perimeter network (DMZ)

Another scenario that might interest some would be to use the Apache Web server that comes with most Linux distributions.

Server consolidation is one of the primary attractions for running Linux in an iSeries partition. High-end hardware supports 31 Linux partitions and can provide access to a very large amount of DASD – 64 GB x 48 drives. This capacity enables the possibility of many other implementations, such as:

- Partitions for testing Linux patches and applications before putting them into production
- Hot swap servers (could maintain mirrored partitions to swap with a corrupted partition)
- A form of Storage Area Network
- Clustered servers
- A separate Linux partition for application development

1.6 Performance

The information on Linux performance on iSeries can be found in *iSeries Performance Capabilities Reference V5R1*, SC41-0607. This book is available from the iSeries Information Center CD-ROM or from the Web site at:
`http://publib.boulder.ibm.com/pubs/html/as400/v5r1/ic2924/index.htm`

As performance hints become available, they will be posted at the following Web site:
`http://www.ibm.com/servers/eserver/iseries/linux/performance.html`

1.7 Summary

Linux and the open source community have a very flexible and agile Internet presence and development community that can benefit the iSeries platform by providing exposure to the open source community and a multitude of Internet applications. Linux is growing rapidly as a server platform. It is attracting attention because of its reputation for stability, scalability, and flexibility, which are available at very little immediate cost. In many cases, some of this cost may be offset by the need for in-house programming and support. This also implies the ability to provide very specific solutions rather than altering a shrink-wrap solution that does not totally fit particular needs.

iSeries hardware and OS/400 provide a platform of proven stability and world class performance. Little needs to be said in regard to the proven capabilities and broad popularity of OS/400, since this publication is directed at OS/400 users implementing Linux.

Each operating system sharing the hardware can potentially provide benefits to the other that will only increase as the integration is enhanced over time. This implementation will be unique because it is PowerPC based and requires the existence of OS/400 in the primary partition to provide the mechanism for initially booting Linux on the hardware.

Two distinct environments are possible:

- A hosted environment in which Linux depends on the hosting OS/400 partition for some or all of its I/O and is booted and shutdown from OS/400. The major advantage of this is the ability to save Linux data for emergency backup and access to large amounts of DASD. A mixed environment, which is really a blend of the two solutions such as native communications using dedicated LAN adapters while the rest of the I/O is virtual, seems likely to be the most popular solution and is the recommended solution for most environments. Such a “mixed” environment is actually a subset of the hosted environment because any of the virtual services to DASD, CD-ROM, or communications require a NWSD and, therefore, a hosting partition.
- A non-hosted environment in which all I/O is “native” and controlled by Linux. Linux boots using “native” DASD and does not have dependencies upon OS/400 after installation, except for the very low-level partitioning code upon which all secondary partitions depend. This has no effect on Linux. Beyond the fact that IPLing the primary partition, or a serious problem that brings down the primary partition, will likewise bring down the Linux partition, this is also true for other secondary partitions.

2

Installation and operation of Linux on the iSeries server

This chapter covers the following topics:

- System requirements
- LPAR
  - Configuring a hosted partition
  - Considerations for non-hosted partitions
- Linux setup
  - Distributions
  - Virtual I/O
    - Virtual Console
    - Virtual LAN
    - Virtual DASD
    - Virtual Optical
    - Virtual Tape
  - Native I/O
    - Native LAN Adapters
    - Native DASD
    - Native Optical
    - Native Tape
- General problem determination suggestions
  - Common problems

2.1 System requirements

The ability to divide OS/400 into logically separate partitions that have their own resources and can concurrently run different copies of OS/400 (LPAR) has been available on the iSeries server since V4R4. A primary partition running OS/400 is required to manage the other partitions.

New processors and other enhancements in V5R1 now allow processors to be pooled and a fractional amount of a processor to be allocated to a partition from this pool. This is very desirable for the primary partition to be allocated for partition management only. It is also a significant factor in implementing guest partitions since Linux may not require an entire processor for the workload it runs. OS/400 V5R1 enables the guest partition support that allows Linux to run directly on the iSeries hardware as a guest operating system. The following system requirements pertain to guest partition support:

Attention: When references are made to processor capabilities, see the tables in Appendix C, “Hardware planning, ordering, and configuration examples” on page 289.

- OS/400 version V5R1 or higher must be running in the primary partition. For more information on primary partitions or for detailed information on partitioning iSeries servers, see: http://www-1.ibm.com/servers/eserver/iseries/lpar/
- Make sure all latest Linux related PTFs have been installed. For a list, see http://www.ibm.com/iseries/linux/ptfs.html
- For Linux support the iSeries server must be the newer Model 270 and 8xx servers that have specific processor feature codes. See the tables in Appendix C, “Hardware planning, ordering, and configuration examples” on page 289, or the Information Center Web site at: http://publib.boulder.ibm.com/html/as400/infocenter.html
- At least 256 MB main memory is the minimum requirement for the primary partition that runs OS/400. For a Linux partition, at least 64 MB main memory is required and 128 MB or greater main memory is recommended. (Actually, at least 65 MB is required to support memory management related data.)
- To utilize the shared processor capability that allows processors to be pooled and assigned in fractional increments, the system must have a particular processor feature code. See the tables in Appendix C, “Hardware planning, ordering, and configuration examples” on page 289.

  Attention: If the partition is defined to share processors and the processor is not the correct type of processor, you won’t receive a warning, but the guest partition will fail to start.

- With machines that do not support the shared processor capabilities, the user must set the QPRCMLTTSK system value to “0”. To verify this, use the command:

  ```
  DSPSYSVAL SYSVAL(QPRCMLTTSK)
  ```

- The interactive processing feature cannot be allocated to a guest partition since Linux does support this feature.
- A maximum of 31 guest partitions can be defined.
- Virtual LAN can be used to establish multiple high speed inter-partition connections.
- Optionally, LAN adapters, DASD, tape, and optical IOA can be dedicated to the Linux partition in *native* or *direct attach mode*. However, not all IOAs are supported in this manner, so check Appendix C, “Hardware planning, ordering, and configuration examples”

on page 289, or check the Information Web site at:
http://publib.boulder.ibm.com/html/as400/infocenter.html

- If you plan to use directly attached DASD, referred to as “native” DASD in some publications, a proprietary IBM SCSI driver is required. The name of the driver is *ibmsis.o*. The latest version can be found at http://www.ibm.com/servers/eserver/iseries/linux in the developer resources section. It will be available on an IBM Web site as a “tarball”, which is a file that has been archived and compressed using the Linux tar utility (some distributors may package this as part of their distribution).
- A Linux distribution that supplies the iSeries PowerPC extensions and installation instructions is required. At this time, SuSE, Red Hat, and Turbolinux are working with iSeries to provide their software distributions.

2.2 Getting started

The process for installing and configuring Linux in a guest partition involves the following steps:

1. Planning: For the detail hardware planning information, please review “Hardware planning” on page 290.
   - What resources will your Linux partition require and what do you have available?
     - Do you have LAN adapters, DASD, CD-ROM, and tape drives that can be dedicated to Linux and not utilized by OS/400? Generally this means that you have multiples of each device that can be dedicated. If you do not have this configuration and need to share DASD, CD-ROM or tape drive, then you need to create a hosted partition.
     - Do you want to be able to use OS/400 to save an entire virtual disk’s worth of Linux data? If so, you need a hosted partition.
     - Are you going to use the Linux server as a firewall, Web server in a demilitarized zone, or other implementation that requires a degree of isolation from OS/400? If so, you will want directly attached LAN adapters and may want DASD directly attached. These conditions could be implemented using either a hosted or non-hosted environment. If you choose to implement a non-hosted environment, then you need to have a dedicated CD-ROM and tape drive. The primary consideration for a non-hosted environment is whether you have the resources to dedicate to the partition. Problem determination in a non-hosted environment may be slightly more difficult since the partition and its hardware are isolated from OS/400.
2. Creating a guest partition:

   This is discussed in detail on 2.3.2, “Configuring a guest partition” on page 23.
3. Installing Linux:
   - Installation depends on the distribution chosen and the process each distributor has implemented. It may require that you obtain documentation from the distributor.
   - To install Linux on iSeries, you need a network server description, and a Network Server Storage Space. These are discussed in detail in 2.4, “Linux installation” on page 36.
4. Implementing Linux applications: The following chapters have information regarding Linux applications and concerns.

2.3 LPAR considerations

To install Linux on iSeries, minimal mastery of configuring guest partitions is required. Before partitioning any system, you should understand the capabilities of the processors on your system and try to estimate what resources each partition will require. Since this will vary by what is to be installed and how it will be used, it would be extremely difficult to provide any specific guidelines. You should be aware of the minimum requirements for a guest partition as stated in 2.2, “Getting started” on page 19, and may want to research your Linux implementation.

In V5R1, Linux partitions do not support dynamic resource movement. Therefore, if you want to change processors or memory, the Linux partition needs to be re-booted for the changes to take effect. For I/O to be added or removed, the partition must be varied off while you are performing these actions.

If you plan to pool processors and share them among partitions, you need to assess how much processing power will be required by each partition. For more information on shared processors, see 1.4.4, “Shared processors” on page 10. There are no benchmarks available at this time to aid in assessing the needs for Linux. However, it would be wise to research the Linux documentation with regard to a particular environment you intend to implement. The specifications for iSeries processors should be available via the configurator. In the absence of concrete guidelines, you should do some testing to determine what configuration suits your environment, before the actual implementation of shared processors.

If you will use virtual I/O, you need to consider that the hosting partition provides the resources. While this should not seriously impact the Linux partition, the requests from the Linux partition are additional tasks that the hosting partition has to service. If the system is busy, it could impact the Linux partition. There could be contention for I/O resources on the hosting partition – in particular, for virtual DASD. Possible solutions, if this becomes a problem, include allocating virtual DASD resources from a separate dedicated ASP or using “native” DASD directly attached to the Linux partition, which Linux accesses directly through its drivers and which is not shared by OS/400.

Virtual LAN I/O is another consideration. Requests can be passed through it, to the network by way of a OS/400 communications adapter, and replies are received in the same way. However, this communication must be routed through the OS/400 (which adds an additional hop to the route), so there may be some difference in round trip times as compared to a directly attached LAN adapter. Data transfer across the virtual LAN may not be at the expected rate depending on what is being accessed and how it is being accessed.

The virtual Ethernet can transfer 8996 byte frames, so you may enhance performance by configuring the maximum frame size in the line description of the virtual LAN to this value. The impact will be identical to the effect such a change would have on any other LAN. It may be either positive or negative depending on whether the devices it is communicating with have the same capabilities. But keep in mind that for communication over different networks with different maximum transfer unit sizes, you need to use the smallest one in order to avoid fragmentation of packages (which costs performance).

In a non-hosted environment, you must determine how to back up the data, since the Linux data will not be in the OS/400 single level storage. FTP across the virtual LAN would be a possibility. Of course, if the partition has a tape drive directly attached, Linux applications can be used to save data. See 3.10, “Backup and recovery” on page 121, for more information on save/restore in this environment.

Some combination of both virtual and native resources is the expected mode of operations. This would be a hosted environment, and we expect that this mixed environment will be used extensively. The non-hosted partition will have reduced diagnostic capabilities and requires more I/O hardware. We recommend the hosted environment unless maximum separation of the two environments is required.

2.3.1 Configuring a service tools user ID

To work with system partitions, you need to do some configurations in System Service Tools (SST). *Service tools user IDs* have always been required to access functions in Dedicated Service Tools (DST). However, as of V5R1, service tools user IDs are also required to access System Service Tools (SST). We recommend that you create a service tools user ID for each partition you configure on the machine. That way, different people can have different authorities and cannot access partitions other than the ones for which they have granted authority. QSECOFR is the only service tools user ID on the shipped system that has the authority to add users to the service machine. We recommend that you create the other service tools user ID that has all privileges. Note that service tools user IDs are disabled after three failed attempts to sign on to DST or SST.

Logical partition configuring requires the service tools user ID to have *partition remote panel key authority* for the partition-number you need to configure. To free resources, you also need authority for the partitions from which you need to free the resources.

For more information on creating service tools user IDs, see:

- IBM-Supplied User Profiles and Dedicated Service Tool (DST) Users:
  `http://publib.boulder.ibm.com/pubs/html/as400/v5r1/ic2924/books/c415302510.htm#HDRIBMPRF`
- Logical partition authority:
  `http://publib.boulder.ibm.com/pubs/html/as400/v5r1/ic2924/index.htm?info/rzaj6/rzaj6authority.htm`

Service tools user IDs can only be created and changed through DST. To configure the service tools user ID, follow these steps:

1. To bring up DST on the system console, go to the iSeries front panel and choose option **21**.
2. Sign on to DST with a service tools user ID that has authority to add users. The password can be different then the normal sign on. Note that passwords for a service tools user ID are case sensitive. It will be disabled after three unsuccessful sign-on attempts. If QSECOFR service tools user ID is enabled, it can be used for this.
3. Choose option **5** (Work with DST environment).
4. Choose option **3** (Service tools user profile), which displays the Work with Service Tools User Profiles display shown in Figure 2-1.

```
                        Work with Service Tools User Profiles
                                                                  System: XXXXXXXXX
Type option, press Enter.
  1=Create              2=Change password          3=Delete
  4=Display             5=Enable                   6=Disable
  7=Change privileges   8=Change description
     User
Opt  Profile     Description                                          Status
     Name1       Service user                                         Enabled
```

Figure 2-1 Work with Service Tools User Profiles display

5. Choose **1** (Create). Then the display shown in Figure 2-2 appears.

```
                      Create Service Tools User Profile
                                                    System:   vvvvvvvvvvvv
Service Tools user profile name . . . . :  Consoleusr
  Type choices, press enter
                Password . . . . . . . . . . . . .
                             Allow profile access before
Storage management recovery . . . . . . .  1   1=Yes, 2=No
Set password to expire . . . . . . . . . .  2   1=Yes, 2=No
Description. . . . . . . . . . . . . . .   Console user
```

Figure 2-2 Create Service Tools User Profile display

6. You can create a service tools user ID just for the function that is required. The user ID and password are set here. And if the user will manage non-hosted partitions, select the option to allow access before storage management recovery. Press the Enter key, and the display in Figure 2-3 appears.

```
                      Change Service Tools User Privileges
                                                   System:    nnnnnnnnnnnn
Service tools user profile name . . . . . Consoleusers
Type option, press Enter.
  1=Revoke   2=Grant
Option   Functions                                          Status
         None                                               Revoked
         Disk Units - operations                            Revoked
         System partitions - operations                     Revoked
         System partitions - administration                 Revoked
         Partition remote panel key XXXXXXXX 000            Revoked
  2      Partition remote panel key XXXXXXXX 001            Revoked
```

Figure 2-3 Change Service Tools User Privileges display

7. Enable the user to access the remote console for partition 001. If they require access to other panels, there are more options for partition 002, partition 003, etc.

The only authority the console user needs is the one that is defined above. The default for other options is revoked.

2.3.2 Configuring a guest partition

Configuring partitions is done from the System Service Tools or the Dedicated Service Tools. This example assumes that SST is used when a hosted partition is created. Furthermore, it defers until later the discussion of allocating resources other than processor and memory. There are only a few differences in creating non-hosted partitions, which are discussed afterwards. Defining shared processors is an LPAR issue, but important since someone may not want to dedicate a single processor to a Linux partition. This is discussed in 1.4.4, “Shared processors” on page 10.

You can learn more about configuring guest partitions on the Web at:
http://publib.boulder.ibm.com/pubs/html/as400/v5r1/ic2924/index.htm?info/rzalm/rzalmconfig.htm

To configure a guest partition, resources have to be removed from an existing partition and allocated to the guest partition and then that partition can be configured. This process requires several steps as described in the following sections.

Freeing resources from existing partitions (with dedicated processors)

This section discusses freeing resources in a dedicated processor environment. See “Freeing resources (in a shared processor environment)” on page 26 for a shared processor environment.

Restriction: For machines that do not support the shared processor capabilities, the user must set the QPRCMLTTSK system value to “0”. To verify this, use the command:

```
DSPSYSVAL SYSVAL(QPRCMLTTSK)
```

See the tables in Appendix C, “Hardware planning, ordering, and configuration examples” on page 289, for lists of supported hardware.

1. To remove resources start System Service Tools (STRSST) and log in using the service tools user ID as created in 2.3.1, “Configuring a service tools user ID” on page 21. Select option **5** (Work with system partitions) as shown in Figure 2-4.

```
                    System Service Tools (SST)

Select one of the following:

     1. Start a service tool
     2. Work with active service tools
     3. Work with disk units
     4. Work with diskette data recovery
     5. Work with system partitions
     6. Work with system capacity




Selection
   5

F3=Exit         F10=Command entry         F12=Cancel
```

Figure 2-4 System Service Tools

2. Select option **3** (Work with partition configuration) to reach the display shown in Figure 2-5. On a new system, only the primary partition will exist and it will own all the resources. To allocate resources that are owned by the partition, the resources must be removed from the partition that owns them before they are added to the new partition.

```
                        Work with System Partitions
                                                              System:   SEATTLE
Attention:  Incorrect use of this utility can cause damage
to data in this system.  See service documentation.

  Number of partitions  . . . . . :  4
  Partition manager release . . . . . :  V5R1M0 L000

  Partition identifier  . . . . . :  0
  Partition name  . . . . . :  PRIMARY   *

Select one of the following:

     1. Display partition information
     2. Work with partition status
     3. Work with partition configuration
     4. Recover configuration data
     5. Create a new partition

Selection
   3

F3=Exit   F12=Cancel
```

Figure 2-5 Work with System Partitions

3. Select option **2** (Change partition processing resources) as shown in Figure 2-6.

```
                     Work with Partition Configuration
                                                              System:   SEATTLE
Type option, press Enter.
  1=Change partition name      2=Change partition processing resources
  3=Add I/O resources          4=Remove I/O resources
  5=Change bus ownership type  6=Select load source resource

        Partition
Option  Identifier  Name
    2     0         PRIMARY
          1         LINUX1
          2         LINUX2
          3         LINUX3









F3=Exit   F5=Refresh                       F9=Work with shared processor pool
F10=Work with Virtual LAN configuration    F11=Work with partition status
F12=Cancel                                 F23=More options
```

Figure 2-6 Work with Partition Configuration

Selecting option 2 (Change partition processing resources) accesses the screen seen in Figure 2-7.

```
                  Change Partition Processing Resources
                                                          System:   SEATTLE
Type changes, press Enter.

  Partition identifier and name  . . . . . :  1    PRIMARY

  Current / available number of processors . . . :  1   / 0
  New number of processors . . . . . .  1
  Use shared processor pool  . . . . . .  2  1=Yes, 2=No

  Current / available size of main storage (MB)  :  256      / 0
  New size of main storage (MB)  . . . . . :  256




F3=Exit   F9=Include limits   F10=Work with shared processor pool
F11=Display partition processing configuration   F12=Cancel
```

Figure 2-7 Changing the processors and memory

4. Change the number of processors and the amount of memory to reflect the totals that you wish to remove. On a system with available processors, there would be a prompt to allow the user to change the number of processors and memory. When this process has been accomplished, a screen appears that allows you to confirm the new resource allocations. Press Enter to confirm your selections. Then, you should instantly return to the Remove I/O Resources screen and be notified that the operation was successful. After removing the processing power and memory, you can return to the screen shown in Figure 2-8 and remove I/O resources by selecting option **4**. We do not remove I/O resources at this point.

```
                    Work with Partition Configuration
                                                          System:   SEATTLE
Type option, press Enter.
  1=Change partition name      2=Change partition processing resources
  3=Add I/O resources          4=Remove I/O resources
  5=Change bus ownership type  6=Select load source resource

        Partition
Option  Identifier  Name
          0         PRIMARY
          1         LINUX1
          2         LINUX2
          3         LINUX3









F3=Exit   F5=Refresh                      F9=Work with shared processor pool
F10=Work with Virtual LAN configuration   F11=Work with partition status
F12=Cancel                                F23=More options
```

Figure 2-8 Removing I/O resources

Freeing resources (in a shared processor environment)

To set up a shared processor environment, use the following steps:

1. To remove a resource, start System Service Tools (STRSST) and log in using the service tools user ID created in 2.3.1, “Configuring a service tools user ID” on page 21. Select option **5** (Work with system partitions) as shown in Figure 2-4 on page 23.
2. Select option **3** (Work with partition configuration) to access the display shown in Figure 2-9. On a new system, there is only the primary partition, which owns all the resources. To allocate resources that are owned by the partition, the resources are removed from the partition that owns them before they are added to the new partition.

```
                        Work with System Partitions
                                                              System:   AS03
Attention:  Incorrect use of this utility can cause damage
to data in this system.  See service documentation.

  Number of partitions  . . . . . :  2
  Partition manager release . . . . . :  V5R1M0 L000

  Partition identifier  . . . . . :  0
  Partition name  . . . . . :  PRIMARY   *

Select one of the following:

     1. Display partition information
     2. Work with partition status
     3. Work with partition configuration
     4. Recover configuration data
     5. Create a new partition

Selection
     3
F3=Exit   F12=Cancel
```

Figure 2-9 Work with System Partitions

3. Type **3** (Work with partition configuration), and press Enter. The display in Figure 2-10 appears.

```
                     Work with Partition Configuration
                                                              System:   AS03
Type option, press Enter.
  1=Change partition name      2=Change partition processing resources
  3=Add I/O resources          4=Remove I/O resources
  5=Change bus ownership type  6=Select load source resource

        Partition
Option  Identifier  Name
          0         PRIMARY
          1         LINUX









F3=Exit   F5=Refresh                       F9=Work with shared processor pool
F10=Work with Virtual LAN configuration    F11=Work with partition status
F12=Cancel                                 F23=More options
```

Figure 2-10 Work with Partition Configuration

4. Type **F9** (Work with shared processor pool), which brings you to the display in Figure 2-11.

```
                      Work with Shared Processor Pool
                                                              System:   S104DFNM
Enter non-zero value to create the shared processor pool.

Number of system processors . . . . . . . . . . . :  2
Number of available system processors . . . . . . :  0
Number of system processors allocated to pool . . .  0

     Par            --Total Processors--  -----------Units Used-----------
Opt  ID   Name      Cur / Pnd  Min / Max     Cur / Pnd        Min / Max

  (No pool exists)
```

Figure 2-11 Work with Shared Processor Pool

5. Specify the number of system processors to allocate to the pool and press Enter. The display shown in Figure 2-12 appears.

```
                      Work with Partition Configuration
                                                              System:   S104DFNM
Type option, press Enter.
  1=Change partition name      2=Change partition processing resources
  3=Add I/O resources          4=Remove I/O resources
  5=Change bus ownership type  6=Select load source resource

        Partition
Option  Identifier  Name
  2       0         PRIMARY
```

Figure 2-12 Work with Partition Configuration

6. Type option **2** (Change partition processing resources) as shown in Figure 2-12. The display shown in Figure 2-13 appears.

```
                  Change Partition Processing Resources
                                                        System:   S104DFNM
Type changes, press Enter.

  Partition identifier and name  . . . . . :  0    PRIMARY

  Current / available number of processors . . . :  2   / 0
  New number of processors . . . . . .  2
  Use shared processor pool  . . . . . .  1  1=Yes, 2=No

  Current / available size of main storage (MB)  :  1536     / 0
  New size of main storage (MB)  . . . . . .  1536

  Current / available interactive feature  . . . :  100 / 0   %
  New interactive feature  . . . . . .  100 %









F3=Exit   F9=Include limits   F10=Work with shared processor pool
F11=Display partition processing configuration   F12=Cancel
```

Figure 2-13 Change Partition Processing Resources

7. Set Use shared processor pool to **1** (Yes) as shown in Figure 2-13 and press Enter. The display shown in Figure 2-14.

```
                  Change Partition Processing Resources
                                                        System:   S104DFNM
Type changes, press Enter.

  Partition identifier and name  . . . . . :  0    PRIMARY

  Current / available number of processors . . . :  2   / 0
  New number of processors . . . . . .  2
  Use shared processor pool  . . . . . .  1  1=Yes, 2=No
    New shared processor pool units  . . . . . .  0.50

  Current / available size of main storage (MB)  :  1536     / 0
  New size of main storage (MB)  . . . . . .  1536

  Current / available interactive feature  . . . :  100 / 0   %
  New interactive feature  . . . . . .  100 %
```

Figure 2-14 Change Partition Processing Resources

8. Fill in the New shared processor pool units parameter. In the example above, the primary partition is given a *0.50* processor. Press Enter when you have defined the fractional amount. Then, the display in Figure 2-15 appears.

```
                        Confirm Changed Partition
                                                          System:   S104DFNM
Verify information, press Enter.

  Partition identifier and name  . . . . . :  0    PRIMARY
  Number of partition processors . . . . . :  2
  Minimum / maximum number of processors . . . . :  1   / 2
  Use shared processor pool  . . . . . :  Yes
    Shared processor pool units  . . . . . :    0.50
    Minimum / maximum processor pool units . . . :    0.10 /   2.00
  Size of partition main storage (MB)  . . . . . :  1536
  Minimum / maximum size of main storage (MB)  . :  288      / 1536
  Partition interactive feature  . . . . . :  100 %
  Minimum / maximum interactive feature  . . . . :  3   / 100 %
  Virtual OptiConnect / Virtual LAN  . . . . . :  No     / No
```

Figure 2-15 Confirm Changed Partition

9. Verify that the amount in the Shared processor pool units parameter is what you wanted. Then press Enter to change the partition. Note that this process is no different for a guest partition than it is for an OS/400 partition.

Attention: Please note that shared processors are only available on selected processors. Not all processor models that support guest partitions can provide shared processors. Some will require that a whole number of processors is allocated to partitions. See “Hardware planning” on page 290 for more information.

Creating the new guest partition

To create the new guest partition, follow these steps:

1. Once resources are made available for the guest partition, you need to create the new partition. Return to the main screen, and select option **5** (Create a new partition) as shown in Figure 2-16.

```
                        Work with System Partitions
                                                              System:   SEATTLE
 Attention:  Incorrect use of this utility can cause damage
 to data in this system . See service documentation.

   Number of partitions  . . . . . :  4
   Partition manager release . . . . . :  V5R1M0 L000

   Partition identifier  . . . . . :  0
   Partition name  . . . . . :  PRIMARY   *

 Select one of the following:

      1. Display partition information
      2. Work with partition status
      3. Work with partition configuration
      4. Recover configuration data
      5. Create a new partition

 Selection
5

 F3=Exit   F12=Cancel
```

Figure 2-16 Work with System Partitions

2. To create a guest partition, continue with the following screens. Select option **2** (Guest) as shown in Figure 2-17.

```
                        Select Operating Environment
                                                              System:   SEATTLE
 Select one of the following:

      1. OS/400
      2. Guest









 Selection
      2

 F3=Exit   F12=Cancel
```

Figure 2-17 Selecting to configure a guest partition

3. On the screen that appears (Figure 2-18), follow these steps:
   a. Specify:
      - A name for the partition. Any name that is appropriate or meaningful to your environment is recommended.
      - A numeric partition number (**Note**: The primary partition is always “0”).
   b. Specify the number of processors for this partition:
      - For whole processors, specify a number of processors out of the number available. Note when this system was new, 3 processors would have been available and a selection of 1, 2, or 3 would be acceptable.
      - To use shared processors. Select F10 and this will allow the selection of the processor pool. Next select **1=Yes** for “Use shared processor pool”. This would bring up a prompt to enter the fractional amount.
   c. Enter the size of the partition main storage (a minimum of 64 MB; 128 MB or greater is recommended).
   d. Press the Enter key.

```
                          Create New Partition
                                                          System:   SEATTLE
Complete blanks, press Enter.

  Partition identifier and name  . . . . . .  4

  Number of available system processors  . . . . :  1
  Number of partition processors . . . . . .  1
  Use shared processor pool  . . . . . .  2  1=Yes, 2=No

  Size of available system main storage (MB) . . :  0
  Size of partition main storage (MB)  . . . . . .

  Interactive feature available  . . . . . :  0   %
  Partition interactive feature  . . . . . .     %




F3=Exit   F9=Include limits   F10=Work with shared processor pool
F11=Display partition processing configuration   F12=Cancel
```

Figure 2-18 Create New Partition

4. A confirmation screen follows. If the values are correct, press the Enter key. Then, the next screen, Select Communication Options, shown in Figure 2-19, is displayed. Specify the virtual LAN port for the partition. LAN 0 is selected in this case, but any of them could be selected. Make sure there is communication between the host partition and guest partition by having the host partition and guest partition attached to the same VLAN.

```
                        Select Communication Options
                                                              System:   SEATTLE
  Partition identifier . . . . . . . . . . . . . . :  4
  Partition name . . . . . . . . . . . . . . . . . :  LINUX1

Type changes, press Enter.
  1=Yes  2=No

  -------------------Virtual LAN Identifiers-------------------
  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
  1   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2









F3=Exit   F11=Display communication options   F12=Cancel
```

Figure 2-19 Configuring the virtual LAN

5. A confirmation screen follows showing the information about the partition that was just created. If correct, press the Enter key as shown in Figure 2-20.

```
                            Create New Partition
                                                              System:   SEATTLE
Complete blanks, press Enter.

  Partition identifier and name . . . . . . . . . . 4

  Number of available system processors  . . . . :  1
  Number of partition processors . . . . . . . . .  1
  Use shared processor pool  . . . . . . . . . . .  2  1=Yes, 2=No

  Size of available system main storage (MB) . . :  0
  Size of partition main storage (MB)  . . . . . .  66









F3=Exit   F9=Include limits   F10=Work with shared processor pool
F11=Display partition processing configuration   F12=Cancel
```

Figure 2-20 Confirming selections

6. From the Work with Partition Configuration screen, it is necessary to specify a hosting partition. Select **F23** (More options) as shown in Figure 2-21.

```
                     Work with Partition Configuration
                                                              System:   SEATTLE
Type option, press Enter.
  1=Change partition name      2=Change partition processing resources
  3=Add I/O resources          4=Remove I/O resources
  5=Change bus ownership type  6=Select load source resource

        Partition
Option  Identifier  Name
          0         PRIMARY
          1         LINUX1
          2         LINUX2
          3         LINUX3









F3=Exit   F5=Refresh                        F9=Work with shared processor pool
F10=Work with Virtual LAN configuration     F11=Work with partition status
F12=Cancel                                  F23=More options
```

Figure 2-21 Work with Partition Configuration

Setting the hosting partition

To set the hosting partition, follow these steps:

1. From the Work with Partition Configuration screen, enter option **13** (Change host) next to the Linux partition to be “hosted”. See Figure 2-22.

```
                    Work with Partition Configuration
                                                          System:   SEATTLE
Type option, press Enter.
  7=Select console resource          8=Select alternate IPL resource
  9=Select default electronic customer support resource
 10=Change comm options  11=Delete  12=Change oper env  13=Change host

        Partition
Option  Identifier  Name
          0         PRIMARY
          1         LINUX1
          2         LINUX2
          3         LINUX3









F3=Exit   F5=Refresh                      F9=Work with shared processor pool
F10=Work with Virtual LAN configuration   F11=Work with partition status
F12=Cancel                                F23=More options
```

Figure 2-22 Work with Partition Configuration

2. A list of partitions available to host the Linux partition is presented. In this case, only the primary partition was available so we entered a **1** next to it. But, if there was another OS/400 secondary partition available, we could have selected it as the host. See the selection of available hosts in Figure 2-23.

```
                           Change Host Partition
                                                          System:   SEATTLE
  Partition identifier . . . . . . . . . . . . . :  1
  Partition name . . . . . . . . . . . . . . . . :  LINUX1

Type option, press Enter.
  1=Select

        -----------Host Partition------------
Option  Identifier  Name      Version/Release
  1       0         PRIMARY   V5R1M0 L000      *









* Indicates current host.
F3=Exit   F12=Cancel
```

Figure 2-23 Selecting the host partition

Setting up the virtual LAN

To set up the virtual LAN, follow this process:

1. From the Work with Partition Configuration screen, select **F10** (Work with Virtual LAN Configuration). The last step is to set up the virtual LAN in case you want to use it. Make sure the virtual LAN number you want to use is available to all Linux and OS/400 partitions that need this virtual LAN channel. An example of a virtual LAN configuration is shown in Figure 2-24. We selected LAN 0 because it was the first one available in sequence, but there is no particular reason that it had to be “0”.
2. If the hosting partition is not connected, complete these steps:
   a. Choose option **2** (Change).
   b. Add the host partition to the virtual LAN.

```
                    Work with Virtual LAN Configuration
                                                          System:   SEATTLE
Type options, press Enter.
  2=Change

    Par          -------------------Virtual LAN Identifiers-------------------
Opt ID  Name     0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
      0 PRIMARY  1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
      1 LINUX1   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
      2 LINUX2   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .
      3 LINUX3   1   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .









'1' Indicates LAN in use.  '.' Indicates LAN not in use.
F3=Exit                              F9=Show all partitions
F11=Display communication options    F12=Cancel
```

Figure 2-24 Check the primary partition is connected to the virtual LAN

Finally, you must IPL the primary partition in order for your new partitions to be activated. You should use the PWRDWNSYS command from active OS/400 secondary partitions to ensure a normal shutdown occurs. Then, if possible, use the PWRDWNSYS command for the primary partition.

2.4 Linux installation

This section takes you through some important considerations when installing Linux on iSeries.

Important: The actual procedure for installing Linux is provided by each of the individual distributions. We provide a few notes and a sketch of some considerations for the initial installation and the boot process thereafter. These are far from complete. For instructions about the installation, please see the documentation provided by the distributor or contact the distributor for clarification of the documentation.

2.4.1 Choosing a distribution

IBM is currently working with the following distributors to provide the distribution of the iSeries PowerPC extensions. Please refer to these distributors' Web sites for more information:

- SuSE: http://www.suse.com/index_us.html
- Turbolinux: http://www.turbolinux.com
- Red Hat: http://www.redhat.com

Or you can visit the IBM iSeries Linux home page at: http://www.ibm.com/iseries/linux

You need a distributor that has a PowerPC-enabled version of the Linux kernel including support for iSeries. Refer to Chapter 4, "Advanced administration and development" on page 137, for an in-depth discussion about the kernel and needed modules.

2.4.2 Installation details

As mentioned already, Linux cannot be installed as the only operating system on an iSeries server, but must be installed in a secondary partition of a system running OS/400 V5R1 (or later) in the primary partition. Another fundamental difference when installing Linux on the iSeries server results from the difference between the way in which OS/400 is loaded as opposed to other computers that use BIOS.

When you install Linux on PCs, it installs information into the Master Boot Record (MBR) so that the system boots to Linux's boot loader on a partition marked as "bootable". After the installation, Linux uses the Linux Loader (LILO), which is its boot loader, to boot the kernel. On other systems Linux can be installed to boot from another boot loader, such as System Commander or Partition Magic, or alternatively it can boot other operating systems.

Since iSeries hardware has only needed to boot OS/400 until now, there is no generic process for loading and installing operating systems other than OS/400. Therefore, there needs to be a process to load the kernel into the partition, initialize it, and link to the root file system in order to start the rest of Linux.

The mechanism to do this initially is the network server description (NWSD). A NWSD is necessary for the installation of both the hosted and non-hosted environments. The NWSD resides on the hosting partition.

Important: Both for hosted and non-hosted partitions, the installation process in done using a network server description. All non-hosted partitions initially are loaded in a hosted environment that gives the advantage that the OS/400 Optical Device can be used to install from the CDs.

After the initial installation, the NWSD remains the mechanism for booting a hosted Linux partition. However, subsequent boots of the non-hosted environment would no longer require the NWSD. Non-hosted partition would be booted from the SST Virtual Service Panel.

Please refer to each distributors' documentation for the installation procedure in detail.

Before you begin, verify that all LPAR configurations are done. If this partition will use virtual services, the partition needs to have a defined host.

2.4.3 Creating an initial network server description (CRTNWSD)

The NWSD object is shared with the Integrated xSeries Server for iSeries, so not all of the parameters apply. To create a NWSD, use the Create Network Server Description (CRTNWSD) command.

The important parameters used in Figure 2-25 are:

- **Network server description** is the name of the NWSD (in this example `LINUX1`).
- **Resource name** is `*NONE`, which means you do not reference physical resources.
- **Network server type** is `*GUEST`.
- **Online at IPL** is `*NO` (this could be set to `*YES`).
- **Partition** is the name of partition into which Linux will be installed (`LINUX1` in this example).
- **Code Page** is `437` (the default `*LNGVER` is not supported, so you must enter a specific code page).
- **Restricted device resources** `*NONE` restricts the tape and optical devices that the server can use. This would reference devices shared by the OS/400 and is strictly optional but may be important if you experience device contention.

The other pertinent parameters in the NWSD are shown in Figure 2-27 on page 41 and are discussed in the following section.

```
                    Create Network Server Desc (CRTNWSD)

Type choices, press Enter.

Network server description . . . > LINUX1          Name
Resource name  . . . . . . . . . > *NONE           Name, *NONE
Network server type  . . . . . . > *GUEST          *WINDOWSNT, *GUEST
Online at IPL  . . . . . . . . .   *no             *YES, *NO
Vary on wait . . . . . . . . . .   *NOWAIT         *NOWAIT, 1-15 minutes
Partition  . . . . . . . . . . .   LINUX1          Name
Code page  . . . . . . . . . . .   437             *LNGVER, 437, 850, 852, 857...
Server message queue . . . . . .   *JOBLOG         Name, *JOBLOG, *NONE
  Library  . . . . . . . . . . .                   Name, *LIBL, *CURLIB
TCP/IP port configuration:
  Port . . . . . . . . . . . . .   *NONE           *NONE, *INTERNAL, 1, 2, 3
  Internet address . . . . . . .
  Subnet mask  . . . . . . . . .
  Maximum transmission unit  . .                   Number
              + for more values

                                                                      More...
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 2-25 Create Network Server Description

2.4.4 Boot parameters of the NWSD

The following parameters in the network server description control from where the Linux kernel will be booted. See Figure 2-26 on page 40 for more information. You can also refer to 4.3.7, “Compiling the kernel” on page 161, for information about how to install the kernel into these IPL sources.

Important: Refer to the distributor's installation documentation for exact values of these parameters because they are all different.

- **IPL source** is set to `*NWSSTG` in this example. This parameter defines where the kernel is located that the server will be booted from and has the following possibilities (see Figure 2-26 on page 40):
  - **`*NWSSTG`** means that the boot source is a virtual disk associated with the NWSD. It must have a partition formatted as a primary partition of type "PReP Boot" (type 0x41) marked as bootable. An optimal size for the PReP partition would be 7 MB to 8 MB. If the size of the PReP partition is too small, then the kernel cannot fit in the PReP partition. For more information on partitioning, see 3.2, "Devices" on page 90.
  - **`*STMF`** means the boot source is a file in the Integrated File System (IFS) of the OS/400 partition. The kernel is, therefore, a read out of a stream file. This could include a CD-ROM that would be mounted into the Integrated File System on iSeries under the directory QOPT and is most likely what will be used to install from the CD-ROM.
  - **`A`** or **`B`** means the Linux system boots from slot A or B. The semantics are derived from the iSeries IPL options, which allow different copies of OS/400 to be loaded. To do this, the /proc file system is used with a command similar to the following example:

    ```
    dd if=/usr/src/linux/vmlinux of=/proc/iSeries/mf/A/vmlinux
    ```

    Refer to 4.3, "Building the kernel" on page 154, for detailed information on working with the kernel. The advantage of doing this would be a faster startup. However, there may also be disadvantages, among which it cannot be saved and restored.
  - **`*PANEL`** means that in the SST Partition configuration panel must specify which IPL source must be used.
  - The other values are reserved.
- **IPL stream file** is the path to the stream file that the server would boot from when *STMF is specified as the IPL source.
- **IPL parameters** offer a way of to give boot time parameters to a Linux system, such as can be specified at the *lilo* prompt on a PC Linux system. For more information on these parameters, refer to each distributor's documentation.

Note: If you boot the kernel without network server description, but from a kernel slot, the IPL parameters given here will not be used. Instead, the kernel is given the arguments that have been stored in the proc file system.

```
                    Create Network Server Desc (CRTNWSD)

Type choices, press Enter.

Ports:
  Port number  . . . . . .   *NONE         1, 2, 3, *INTERNAL, *NONE
  Line description . . . . . .              Name
              + for more values
Synchronize date and time  . . .   *TYPE         *TYPE, *YES, *NO
IPL source . . . . . . . . . . .   *NWSSTG       *NWSSTG, *PANEL, *STMF, A...
IPL stream file  . . . . . . . .   *NONE

IPL parameters . . . . . . . . .   *NONE
Authority  . . . . . . . . . . .   *LIBCRTAUT    Name, *LIBCRTAUT, *CHANGE...
Text 'description' . . . . . . .   *BLANK




                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 2-26 Boot or IPL parameters for the Linux partition

A network server description can be viewed and its status determined by using the command:

```
WRKCFGSTS *NWS
```

From the status screen, the NWSD can be varied on or off. Please note that this process is the required way to IPL the partition for a hosted partition.

2.4.5 Create Network Server Storage Space (CRTNWSSTG)

Network server storage spaces (NWSSTG) are basically stream files in the IFS that are made to look like local hard drives to the server. These could contain the kernel and boot images or be linked on OS/400 and mounted (in Linux) as a separate partition or partitions to contain user data. In the case that the server is booting from a network server storage space, it should be the first drive linked. See Figure 2-27. NEWLIN, RHBETAB, and TURBO are all the boot files for their respective servers. LINUX2ALT is additional user space for NWSD Redhat71, which has another NWSSTG linked previously that is not displayed on this screen. The view shown in Figure 2-27 is of existing network server storage spaces using the CL command WRKNWSSTG.

```
                  Work with Network Server Storage Spaces
                                                              System:   SEATTLE
Type options, press Enter.
  1=Create   3=Copy   4=Delete   5=Display   6=Print   10=Add link
  11=Remove link

                Percent
Opt  Name       Used     Size   Server      Drive  Format    Access      ASP

     LINUX2ALT       0     1800  REDHAT71        2   *OPEN     *UPDATE       1
     NEWLIN          0      800  REDLIN          1   *OPEN     *UPDATE       1
     RHBETA7B        0     1536  REDHAT7B        1   *OPEN     *UPDATE       1
     RHBETA71        0     1536                      *OPEN                   1
     RHSMALL         0     1280                      *NTFS                   1
     TURBO           0     1024  LINUX3TL        1   *OPEN     *UPDATE       1



                                                                          Bottom
Parameters or command
===>
F3=Exit   F4=Prompt   F5=Refresh   F6=Print list   F9=Retrieve
F11=Display text      F12=Cancel   F17=Position to
```

Figure 2-27 Work with Network Server Storage Spaces

Network server storage spaces are created using the CL command CRTNWSSTG as illustrated in Figure 2-28 or alternatively using option 1 from the WRKNWSSTG display in Figure 2-27. They can be created as large as 64 GB for a single network server storage space, and a maximum of 48 network server storage spaces can be linked to a server.

```
                    Create NWS Storage Space (CRTNWSSTG)

Type choices, press Enter.

Network server storage space . . > USERDTA       Name
Size . . . . . . . . . . . . . . > 2000          *CALC, 1-64000 megabytes
From storage space . . . . . . .   *NONE         Name, *NONE
Format . . . . . . . . . . . . . > *OPEN         *NTFS, *FAT, *FAT32, *OPEN
Auxiliary storage pool ID  . . .   1             1-99
Text 'description' . . . . . . .   *BLANK









                                                                          Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 2-28 CRTNWSSTG command

Generally *OPEN is the choice for Linux because it allows the operating system to format the drive. Linux supports several file system types, so this is convenient. Note also that the space can be allocated from auxiliary storage pools (ASPs) other than the system pool, which is where it is allocated from by default. Once the network server storage space is created, it needs to be linked. See Figure 2-30 on page 44 to link the network server description and network server storage space (ADDNWSSTL).

Note: One of the most frequent problems associated with creating network server storage spaces has been creating them at a size that is not sufficient for one's needs. Once it is created, there is no way to expand the size dynamically. Also creating a larger network server storage space and attempting to decrease the size at a later date does not work as expected. So if you find that the network server storage space created is filling up and you want it to be larger, create and link another network server storage space and then copy the data between them. See 3.3.2, "Creating and adding a virtual disk" on page 96, for more information.

What are IFS stream files

The discussion of the network server description contains references to *STMF as an IPL source, for example, where the kernel is booted into the Linux partition. It is quite possible that this will be the primary mechanism involved in the installation process.

A stream file is read and written as one long "stream" of data as opposed to a physical file on OS/400 that is record defined. This type of file in an OS/400 environment is typically associated with personal computer files, but that is a very restricted view. The Integrated File System came about in order to provide a space for diverse file systems, and many of the file systems are of the *STMF format.

The scenario that seems most likely to be part of the installation scenario would have the IPL source in the network server description configured to specify *STMF. The IPL stream file would define the path to the kernel and a RAM disk that contains the root file system so that the installation could take over and directly access the CD-ROM. WRKLNK is the command used to access the Integrated File System. Issued without parameters, the command brings up the root of the IFS. Use the WRKLNK command to verify the path to the installation stream file.

To actually associate the storage space with a server, it must be linked to the NWSD. This establishes the pointers in the NWSD so that the association can be maintained. Use either the WRKLNK command or run WRKNWSSTG and select option **10**. Figure 2-29 shows this display.

```
                  Work with Network Server Storage Spaces
                                                              System:   SEATTLE
Type options, press Enter.
  1=Create   3=Copy   4=Delete   5=Display   6=Print   10=Add link
  11=Remove link

                 Percent
Opt  Name        Used     Size   Server      Drive  Format    Access      ASP

     HIHIHO          0       10  REDHAT71        4   *OPEN     *UPDATE       1
     LINUX2ALT       0     1800  REDHAT71        2   *OPEN     *UPDATE       1
     NEWLIN          0      800  REDLIN          1   *OPEN     *UPDATE       1
     REDHBTWO        0     2000  REDHAT7B        2   *OPEN     *UPDATE       1
     RHBETA7B        0     1536  REDHAT7B        1   *OPEN     *UPDATE       1
     RHBETA71        0     1536                      *OPEN                   1
     RHSMALL         0     1280                      *NTFS                   1
     SECONDLIN       0      200  REDLIN          2   *OPEN     *UPDATE       1
     TEMP            0        1  REDHAT71        5   *OPEN     *UPDATE       1
                                                                        More...
Parameters or command
===>
F3=Exit   F4=Prompt   F5=Refresh   F6=Print list   F9=Retrieve
F11=Display text      F12=Cancel   F17=Position to
```

Figure 2-29 Working with network server storage spaces

To remove the link, however, the server must be varied off and you must issue the Remove Network Server Storage Space (RMVNWSSTG) command or you can use option 11 from the WRKNWSSTG display. Storage spaces reside in the Integrated File System under the /qfpnwsstg directory and can be viewed using the command:

```
WRKLNK '/qfpnwsstg/*'
```

They can also be saved from there and restored, using the SAV/RST commands or any save that will save all user data. Restore into a previously created network storage space since SAV does not capture all of the attributes. See 3.10, “Backup and recovery” on page 121, for backup and recovery information.

Note: The storage space link can be added dynamically (Figure 2-30). For example, the space will show as linked and it will not be able to be unlinked or deleted until the server is varied off. However the Linux server will not see the disk until the server is varied off and on. So although the link is established, the actual device is not recognized by Linux until after the server is varied off.

When the partition is varied off and on again, the storage space is shown as an extra disk to Linux. For more information on adding virtual disks and using them in Linux, refer to 3.3, “Handling virtual and direct attached hard disks” on page 95.

Keep in mind that the option to link with *READ access *does not* currently work.

```
                    Add Network Server Storage Link (ADDNWSSTGL)

Type choices, press Enter.

Network server storage space . . > USERDTA       Name
Network server description . . . > REDHAT71      Name
Drive letter . . . . . .          *FIRSTAVAIL    K-Z
Dynamic storage link . . . . . .  *yes           *NO, *YES
Network server type  . . . . . .  *NWSD          Character value
Drive sequence number  . . . . .  *CALC          3-18, *CALC

                          Additional Parameters

Access . . . . . .               *UPDATE       *UPDATE, *READ

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 2-30 Add Server Storage Link

2.4.6 Connecting to the virtual Linux console

The virtual console is used to access the Linux partition to do the initial installation, do diagnostic searches for messages, or to access Linux if you can't access Linux from the network. Virtual console is actually a program running in the low level code below the operating system listening for requests on port 2301 of the hosting partition or the primary partition. It can be utilized from either a Telnet client on OS/400 (not the preferred method) or a PC client Telnet connection.

To connect to the virtual console, Telnet to port 2301 or 21 on the hosting or primary partition and log on with a service profile. Upon connection of a client, the daemon listens on port 2301 and creates a VT100 session that can be accessed directly from a 5250 green screen or a Telnet client running on a PC or other servers.

At the time this redbook was written, you must use care when choosing an appropriate Telnet client because of issues with the mode in which the virtual console is running. As such, there are issues with keystrokes particularly in the 5250 Telnet that make using installation tools and editors such as **vi** or **emacs** difficult.

Telnet client on a PC

There are no problems connecting to the virtual console using a Linux PC or any other UNIX server. From a Linux PC, you would use the **telnet lisa 2301** command to connect to the virtual console, if *lisa* is your iSeries machine. To end the Telnet session, you would use the **telnet quit** command. Before this command can be used, you need to know that you have to press Ctrl-] to enter the Telnet command mode.

However, when using a Telnet client from a Windows PC, the default Telnet client is not recommended. Therefore we strongly recommend using PuTTY. PuTTY is a free program that can be used for Telnet as well as SSH sessions. Download it from the Web at:
http://www.chiark.greenend.org.uk/~sgtatham/putty/

When starting PuTTY, you see the window shown in Figure 2-31. The only two options you need to change are Hostname (where you fill in the IP address or hostname of the primary or hosting OS/400 partition) and Port, which must be set to 2301. See 9.2.8, “ssh clients for Windows” on page 217, for more information about PuTTY and how to customize it.

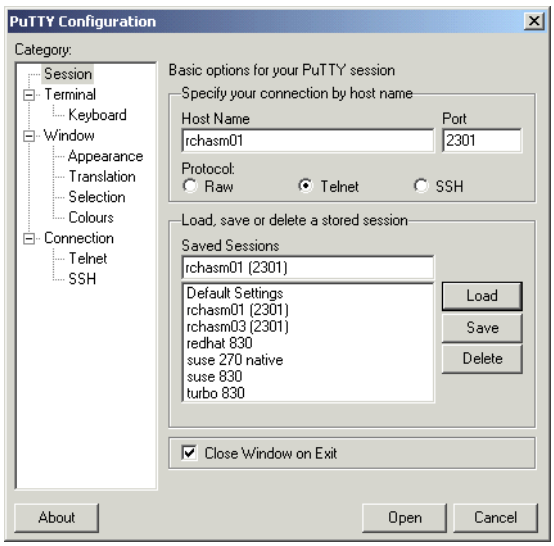


Figure 2-31 Starting a virtual console session with PuTTY

Telnet client in OS/400

Using Telnet from an OS/400 partition is not recommended, because of the previously mentioned errors. However, in case you really have no choice, the following displays show how to use the OS/400 Telnet feature. Figure 2-32 and Figure 2-33 show the parameters.

1. Run **TELNET** from the OS/400 command line, and prompt with F4. See Figure 2-32.

```
                        Start TCP/IP TELNET (TELNET)

Type choices, press Enter.

Remote system  . . . . . .   127.0.0.1

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display        F24=More keys
```

Figure 2-32 Starting the Telnet client for console

2. Fill in the loopback (**127.0.0.1**) as the remote system to connect to, and press F10 to see additional parameters as illustrated in Figure 2-33.

```
                        Start TCP/IP TELNET (TELNET)

Type choices, press Enter.

ASCII tab stops........   *DFT                 0-133, *DFT, *NONE
                  or more values
Coded character set identifier     *MULTINAT     1-65533, *MULTINAT...
     ASCII operating mode ID....   *VT100        *VT220B7, *VT220B8, *VT100...
           Port.............. . .   2301          1-65534, *DFT
Remote virtual display . . . . .   *DFT          Name, *DFT
Remote user  . . . . . .   *NONE         Name, *NONE, *CURRENT
Remote password  . . . . . .   *NONE


Remote password encryption . . .   *DES7         *DES7, *SHA1, *NONE
Remote initial program . . . . .   *RMTUSRPRF    Name, *RMTUSRPRF, *NONE
Remote initial menu  . . . . . .   *RMTUSRPRF    Name, *RMTUSRPRF, *SIGNOFF
Remote current library . . . . .   *RMTUSRPRF    Name, *RMTUSRPRF
Remote keyboard type . . . . . .   *RMTSYS       *RMTSYS, *LCL
Remote codepage  . . . . . .   *RMTSYS       *RMTSYS, *LCL
                                                                      More...
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 2-33 Starting the Telnet client for console extended view

3. Fill in the ASCII operation mode ID (which must be ***VT100**) and the port to connect to (in case of the virtual console, fill in **2301**.)
4. Press the Enter key, which brings up the console screen and a lists of the available partitions as seen in Figure 2-34.

```
OS/400 Guest Partition Console
 1: LINUX1
 2: LINUX2
 3: LINUX3
```

Figure 2-34 The login screen for the virtual console

5. Enter the partition number and press the Enter key, which brings up the display shown in Figure 2-35. There is a prompt for the service tools user ID and password. The user ID is echoed to the display at this time. If the server is not up, the console will wait until the partition becomes active.

```
OS/400 Guest Partition Console
 1: LINUX1
 2: LINUX@
1
LINUX1    : Enter OS/400 service tools user id:
NNNNNNNNN
LINUX1    : Enter OS/400 service tools password:
mmmmmmmmmmmmmmmm

LINUX1:      Console connecting
LNUX1:       Console connected

sh-2.04#
```

Figure 2-35 Filling in the logon information

The user ID and password that are prompted for in Figure 2-35 are for a *system service profile*. (For more information, see 2.3.1, “Configuring a service tools user ID” on page 21.) If you forget to specify port 2301, you will be connected to an OS/400 Telnet session.

Virtual console considerations

The virtual console is primarily for:

- The initial installation
- Solving problems that result in the server becoming inaccessible to the LAN
- The configuration tool for the native DASD and diagnostic capabilities

Alternatively the Client Access function Operations Navigator - LAN Connectivity, hereafter referred to as “LAN console”, can be used. It offers the advantage of encrypted passwords.

Identifying console connections

The case may be that you need to know who is accessing your virtual console or you want to terminate the console connections. If you experience performance issues or if something seems to be hung, you may want to do this also. This can be accomplished by issuing the command from the OS/400 command line:

```
netstat *cnn
```

When reviewing the entries, you should find at least one entry for *as-vcons* as illustrated in Figure 2-36.

If you select F14 (Display port numbers), it should show a local port of 2301 in listening status. This is the daemon and does not need to be cancelled. Scroll down. If there are connections to the console, there will be other entries for *as-vcons* with the remote port of 2301 and a local port of some random port number. Unlike the daemon entry, there will be an IP address displayed, and the connection will show as established. The later entries are the console connections and may be terminated at the discretion of the administrator. The daemon is shown in Figure 2-36.

```
                     Work with TCP/IP Connection Status
                                                          System:   SEATTLE
Type options, press Enter.
  3=Enable debug   4=End   5=Display details   6=Disable debug
  8=Display jobs

     Remote           Remote     Local
Opt  Address          Port       Port       Idle Time  State
     *                *          ftp-con >  008:21:47  Listen
     *                *          telnet     000:14:55  Listen
     *                *          smtp       056:34:39  Listen
     *                *          snmp       005:29:25  *UDP
     *                *          389        000:29:43  Listen
     *                *          drda       056:35:44  Listen
     *                *          ddm        056:35:44  Listen
     *                *          ddm-ssl    056:35:44  Listen
     *                *          as-svrmap  031:25:18  Listen
     *                *          lpd        056:35:38  Listen
     *                *          as-vcons   000:01:09  Listen
     *                *          as-sts     049:16:17  Listen
                                                                  More...
F5=Refresh   F11=Display byte counts   F13=Sort by column
F14=Display port numbers   F22=Display entire field   F24=More keys
```

Figure 2-36 Work with TCP/IP Connection Status

When you scroll down, you see the console connections as shown in Figure 2-37.

```
                    Work with TCP/IP Connection Status
                                                              System:   SEATTLE
Type options, press Enter.
  3=Enable debug   4=End   5=Display details   6=Disable debug
  8=Display jobs

     Remote           Remote     Local
Opt  Address          Port       Port       Idle Time  State
     *                *          as-tran >  079:39:18  Listen
     *                *          as-vrtp >  079:39:17  Listen
                      2039       telnet     000:04:52  Established
                      2408       telnet     000:00:00  Established
                      2465       telnet     000:03:18  Established
                      2527       as-sts     072:55:09  Established
                      2539       as-sts     072:55:07  Established
   127.0.0.1          5099       as-vcons   000:03:18  Established
   127.0.0.1          5099       as-vcons   000:03:18  Established


                                                                      Bottom
F5=Refresh   F11=Display byte counts   F13=Sort by column
F14=Display port numbers   F22=Display entire field   F24=More keys
```

Figure 2-37 Established sessions to the 'virtual' console

2.4.7 Starting the Linux partition

After you connect to the virtual console, you can start the Linux partition to do the initial installation. Both for hosted and non-hosted partitions, the installation process in done using a network server description. This implies that all non-hosted partitions initially are loaded in a hosted environment which gives the advantage that the OS/400 Optical Device can be used to load the CDs.

It's only after the installation that you decide if the partition is hosted or non-hosted, by the way you start the Linux partition. Hosted partitions are started by varying on the NWDS, and non-hosted partitions are started in the SST Virtual Service Panel.

Important: If you start a *hosted partition* from the SST partition status display of the Linux partition, the partition startup will fail. To correct this, simply return to the SST partition status screen and stop the partition. Then vary on the NWSD from the WRKNWSD screen.

Varying on the NWSD

This startup method is used for hosted partitions and during the installation of non-hosted partitions. To vary on the network server description, use the VRYCFG command in the hosting partition (Figure 2-38).

```
                          Vary Configuration (VRYCFG)

 Type choices, press Enter.

 Configuration object . . . . . > LINUX1        Name, generic*, *ANYNW...
               + for more values
 Type . . . . . . . . . . . . . > *NWS          *NWS, *NWI, *LIN, *CTL...
 Status . . . . . . . . . . . . > *ON           *ON, *OFF, *RESET...
 Range  . . . . . . . . . . . .   *NET          *NET, *OBJ
 Vary on wait . . . . . . . . .   *CFGOBJ       *CFGOBJ, *NOWAIT, 15-180 (sec)
 Asynchronous vary off  . . . .   *NO           *NO, *YES
 Reset  . . . . . . . . . . . .   *NO           *NO, *YES
 Resource name  . . . . . . . .                Name, *ALL
               + for more values
 Reset configuration file . . . .  *NO          *NO, *YES
 Forced vary off  . . . . . . . .  *NO          *NO, *YES, *LOCK
 Start TCP/IP interfaces  . . . .  *YES         *NO, *YES
 Submit multiple jobs . . . . . .  *NO          *NO, *YES
 Job description  . . . . . . . .  QBATCH       Name
   Library  . . . . . . . . . .     *LIBL       Name, *LIBL
                                                                      Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 2-38 Varying on a hosted partition

An alternative is to use the WRKCFGSTS *NWS command, where you see an overview of all hosted Linux partitions (together with all Integrated xSeries Server for iSeries servers), and you can vary them on or off using options 1 and 2. See Figure 2-39.

```
                       Work with Configuration Status                 M01
                                                             10/12/01  15:28:33
Position to  . . . . .               Starting characters

Type options, press Enter.
  1=Vary on   2=Vary off   5=Work with job   8=Work with description
  9=Display mode status    13=Work with APPN status...

Opt  Description       Status               -------------Job--------------
     SUSE              ACTIVE
     REDHAT            ACTIVE
     TURBO             ACTIVE


Parameters or command
===>
F3=Exit   F4=Prompt   F12=Cancel   F23=More options   F24=More keys
```

*Figure 2-39 The WRKCFGSTS *NWS display*

Tip: When the vary on of the NWSD fails, check the QSYSOPR messages for an error message stating MSGCPDB1AD and find the reason code. To see these messages, use the following command on the hosting partition:

```
DSPMSG QSYSOPR
```

You can find the explanation for the reason code in Table 2-1 on page 82.

Starting a partition from SST

The other option to start a partition is to use SST. Start the System Service Tools using the STRSST command. Choose option **5** (Work with system partitions) and then choose option **2** (Work with partition status).

Then use options **1**, **3**, **7**, or **8** to start and stop the partitions. An example of this is illustrated in Figure 2-40.

```
                         Work with Partition Status
                                                              System:   M01
Type options, press Enter.
  1=Power on      3=IPL restart   7=Delayed power off   8=Immediate power off
  9=Mode normal  10=Mode manual  11=Mode auto          12=Mode secure
  A=Source A      B=Source B      C=Source C            D=Source D

     Partition             IPL     IPL                 Sys IPL  Reference
Opt  Identifier  Name      Source  Mode     State      Action   Codes
       0         PRIMARY   B       Manual   On         IPL
       1         AS01B     B       Auto     On         IPL
       2         AS01C     B       Auto     On         IPL
       3         LINUX01   B       Manual   On         Hold
       4         REDHAT    A       Manual   On         Hold
       5         TURBO     B       Manual   On         Hold









F3=Exit   F5=Refresh                    F10=Monitor partition status
F11=Work with partition configuration   F12=Cancel   F23=More options
```

Figure 2-40 Starting and stopping partitions from SST

2.5 Virtual I/O

Virtual I/O is discussed in 1.4.5, “Virtual I/O” on page 11, and in previous discussions in this chapter. It is I/O provided by a hosting OS/400 partition.

If you do not have any LAN adapters that can be dedicated to Linux, you can use the virtual LAN to pass Ethernet traffic to and from the Linux partition and to your network via an OS/400 adapter. The configuration is slightly more complex than the configuration of a native LAN adapter. It increases the number of hops enroute to the destination since OS/400 essentially becomes a router that forwards traffic in and out of its OS/400 adapter interface.

2.5.1 Virtual LAN configuration

Virtual LAN is available to OS/400 partitions as well as Linux partitions (both hosted and non-hosted Linux partitions). Data can be transferred between the two partitions in this manner. This might be the only means of communicating to and from the server if directly attached LAN adapters were not allocated. It has been noted elsewhere that this is not a physical adapter or LAN connection. But literally it is a *virtual connection* meaning that it is really emulating an Ethernet LAN. This is not important to the user, since it configures exactly like a physical Ethernet. It has bandwidth similar to a 1 Gb Ethernet, and as many as 16 LANs can be configured.

The most efficient configuration requires the following four steps:

1. Enable the same virtual LAN port during partitioning for the guest partition and the partition to which it will communicate. We will assume virtual LAN port 0 for a guest partition and the primary partition. See “Setting up the virtual LAN” on page 36 for information on how to do this.
2. Define an IP addressing scheme and route. See “Connecting your guest partition to a LAN” on page 53.
3. Configure OS/400. This means you need to create a line description and set up the appropriate interfaces and routes.
4. Configure Linux. This consists of defining an interface, eth0 in our example, and a default gateway. Interactively this is done by means of the **`ifconfig`** command and the **`route`** command from a Linux command line.

Attention: In the initial release of Linux on iSeries, the virtual Ethernet devices were numbered in a (Linux) non-standard way. For example, a connection to virtual LAN 7 would be shown by the kernel as veth7.

While this redbook was being written, this changed to the standard nomenclature. The first device found by the kernel will always be called *veth0*, the second one is *veth1*, and so on, regardless of the number of the virtual LAN to which a device is connected.

Creating an Ethernet line description for virtual LAN

Creating an Ethernet line description for virtual LAN is one of the mandatory steps in using the virtual LAN between OS/400 and Linux partitions. Therefore, this is covered in advance to the distinct network approaches of connecting the partitions as covered in “Connecting your guest partition to a LAN” on page 53.

This configuration allows a guest partition to communicate with an OS/400 partition using virtual LAN. To configure a new Ethernet line description to support virtual LAN, complete the following steps:

1. At the OS/400 command line, type

   ```
   WRKHDWRSC *CMN
   ```

   Then press the Enter key.
2. From the Work with Communication Resources display, select option **7** (Display resource detail) next to the appropriate virtual LAN Ethernet port. The Ethernet port identified as a 268C is the virtual LAN resource. There will be one for each virtual LAN that is connected to the partition, if there are more than one.
3. From the Display Resource Detail display, scroll down to find the port address. The port address corresponds to the virtual LAN you selected during the configuration of the partition.

4. From the Work with Communication Resources display, enter option **5** (Work with configuration descriptions) next to the appropriated virtual LAN Ethernet port and press the Enter key.
5. From the Work with Configuration Descriptions display, select option **1** (Create). Enter the name of the line description, and press the Enter key.
6. In the Create Line Description Ethernet (CRTLINETH) display, provide the following information:

   ```
   RSRCNAME (from step 2)
   LINESPEED(1G)
   DUPLEX(*FULL)
   ```

   Press the Enter key. On the following screen, change Maximum frame size to 8996, and the transfer rate of data across the virtual LAN is improved. Press the Enter key.
7. On the Work with Configuration Description display, a message is displayed that the line description has been created.

Important: Setting a very large maximum framesize is mostly useful for internal communication over the virtual LAN. If packets are also routed to other networks (Ethernet or others) that use a smaller maximum framesize, a value should be used that fits all in order to avoid packet fragmentation.

Connecting your guest partition to a LAN

There are a number of choices for connecting your guest LPAR environment to a LAN. The simplest configuration is for Linux to use a direct-attach LAN adapter to communicate with an external LAN, because you would not have to perform any TCP/IP configuration on OS/400.

Virtual LAN creates a high-speed virtual Ethernet segment that can be used to connect logical partitions in a physical iSeries server. This LAN segment is separate from any real LAN with which the system might have a connection. A virtual LAN is made up of a virtual line description and an OS/400 TCP/IP interface. The Linux TCP/IP interface then has its own IP address but uses the virtual network device for its hardware.

If a guest partition is only connected to a virtual LAN segment, and you want the guest partition to communicate with systems on an external LAN, you need to bridge TCP/IP traffic between the OS/400 external LAN and the virtual OS/400 LAN segment. For a logical flow of the IP packets, see Figure 2-41.

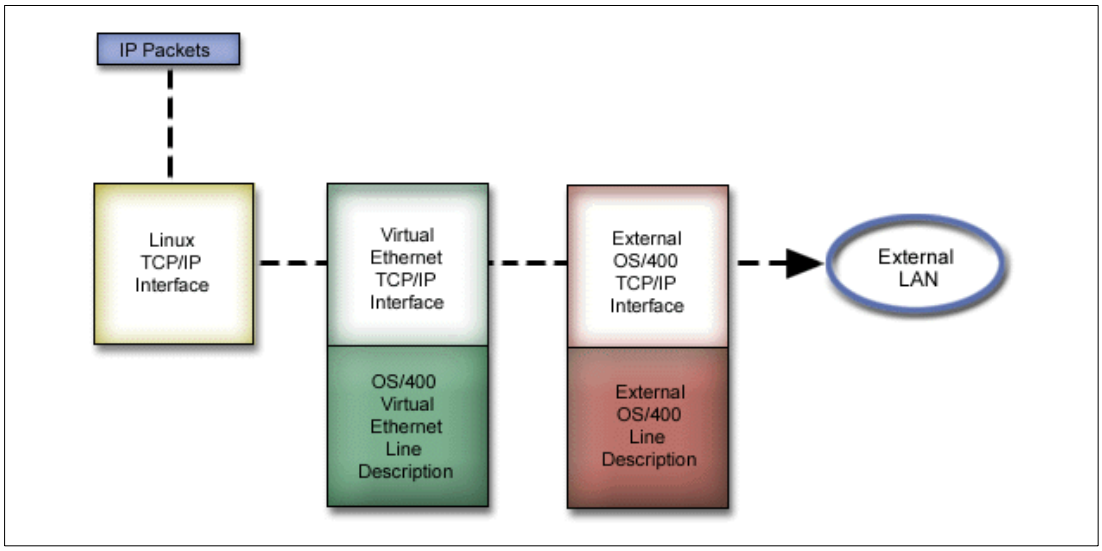


Figure 2-41 A logical flow of the IP packets between an external LAN and virtual LAN

IP traffic initiated by the guest partition goes from the Linux network interface to the virtual OS/400 interface. If the virtual interface is associated with the external interface, the IP packets can continue on to the external interface and toward its destination.

There are three methods for bridging the external and virtual LAN segments. Each method has nuances that make it more feasible based on your knowledge of TCP/IP and your environment. Choose from one of the following methods:

- **Proxy ARP**: This method is a transparent subnetting to associate the guest partition with an external interface. If you have the necessary IP addresses, we recommend using this approach.
- **Network address translation (NAT)**: OS/400 packet filtering can be used to route traffic between a partition and the outside network.
- **TCP/IP routing**: Standard TCP/IP routing is used to route traffic to the new virtual LAN segment in the same way you would define routing to any other LAN segment. This requires updating routing information throughout your network.

Proxy ARP method

The proxy ARP method uses a technique commonly known as *transparent subnetting*. If you choose to use the proxy ARP method, you must have a firm understanding of subnetting and TCP/IP. You need to obtain a contiguous block of IP addresses that are routable by your network. You subnet this block of IP addresses. Then you assign one to the virtual TCP/IP interface and one to the TCP/IP connection in your guest partition as shown in Figure 2-42.

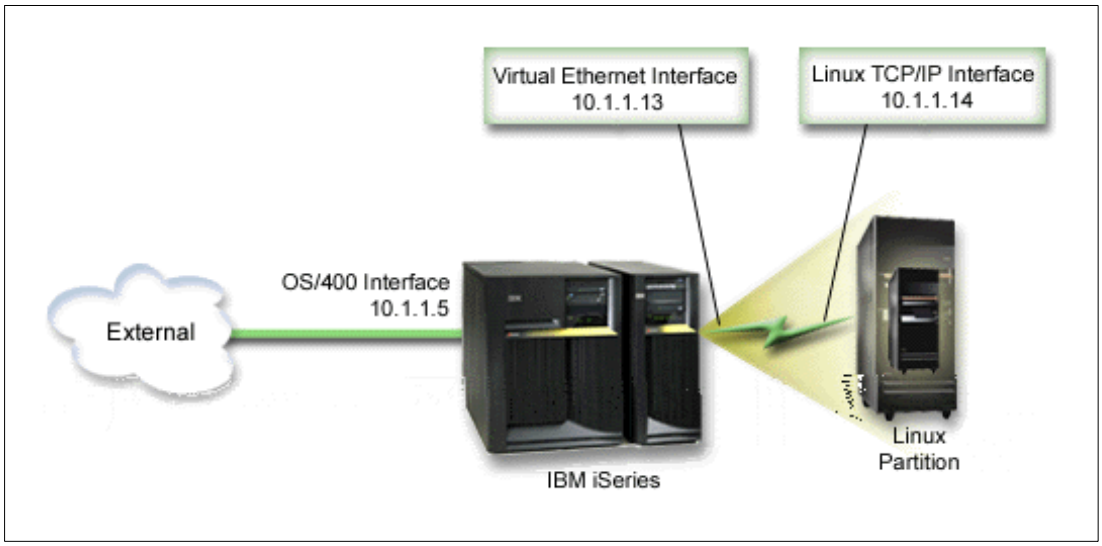


Figure 2-42 Proxy ARP example

In this example, a contiguous block of four IP addresses (10.1.1.12 through 10.1.1.15) is used. Since it is a block of four IP address, the subnet mask for these addresses is 255.255.255.252. To configure a setup similar to this, follow these steps:

1. Obtain a contiguous block of IP addresses that are routable by your network. Since there is only one guest partition, you only need four IP addresses. The fourth segment of the first IP address in the block must be divisible by four. The first and last IP addresses of this block are the subnet and broadcast IP addresses and are unusable. The second and third IP addresses can be used for a virtual TCP/IP interface and a TCP/IP connection in your guest partition. For this example, the IP address block is 10.1.1.12 through 10.1.1.15 with a subnet mask of 255.255.255.252.

   You also need a single IP address for your external TCP/IP address. This IP address does not have to belong to your block of contiguous addresses, but it must be within the same original subnet of addresses as your block. In the example, the external IP address is 10.1.1.5 with a subnet mask of 255.255.255.0.

2. Create an OS/400 TCP/IP interface for your hosting partition. In this example, you would assign 10.1.1.5 as an IP address, and the subnet mask would be 255.255.255.0.

3. Create an Ethernet line description for your virtual LAN. Note the port of the hardware resource, because you will need it when you set up TCP/IP in your guest partition. In the case of the above example, assume the hardware resource is CMN05 with a port of 0, and although arbitrary, the line description name is ETH0.

   ```
   CRTLINETH LIND(ETH0) RSRCNAME(CMN05) LINESPEED(1G) DUPLEX(*FULL)
   ```

4. Add a TCP/IP interface for your Ethernet line description. Following the example, you could use the following command:

   ```
   ADDTCPIFC INTNETADR('10.1.1.13') LIND('ETH0') SUBNETMASK('255.255.255.252')
   LCLIFC('10.1.1.5')
   ```

   Note: If the fourth segment of your virtual TCP/IP interface is greater than the fourth segment of your Proxy ARP IP address, then you need to set AUTOSTART to *NO on this interface.

   OS/400 starts its TCP/IP interfaces in numerical order. You must start the proxy ARP interface before you start any virtual TCP/IP interfaces.

5. Turn on IP datagram forwarding. This allows the OS/400 TCP/IP interfaces to pass IP packets between each other.

   ```
   CHGTCPA IPDTGFWD(*YES)
   ```

6. Start your TCP/IP interfaces. You can use a command like:

   ```
   STRTCPIFC INTNETADR(yourIPaddress)
   ```

   For the example, you would issue commands like the following example to start the external proxy ARP interface:

   ```
   STRTCPIFC INTNETADR('10.1.1.5')
   ```

   Or to start the virtual Ethernet interface, you could enter the command:

   ```
   STRTCPIFC INTNETADR('10.1.1.13')
   ```

7. On your guest partition, set up networking using the instructions or tools provided by your Linux distribution. Be sure to use the correct IP address, subnet mask, port, and router IP address. During the TCP/IP setup in Linux, you need to know your net or network device. The network device is always ETH, plus the port number assigned when you configured an Ethernet line description.

   In the example, the values for these parameters are:

Interface IP address	10.1.1.14
Subnet mask	255.255.255.252
Gateway IP address	10.1.1.13
Network device	ETH0

8. To verify network communications, ping the virtual LAN interface and a host on the external network from your guest partition. Then from OS/400, ping the virtual LAN interface and Linux interface.

Network address translation (NAT)

NAT can route traffic between your guest partition and the external network using virtual LAN. This particular form of NAT is called *static NAT*, and it allows both inbound and outbound IP traffic to and from the guest partition. Other forms of NAT like masquerade NAT may also work if your guest partitions do not need to receive traffic initiated by external clients. Like the TCP/IP routing and proxy ARP methods, you can take advantage of your existing OS/400 network connection. Since you will use IP packet rules, you must use Operations Navigator to create and apply your rules.

Figure 2-43 shows an example of using NAT to connect your Linux partition to an external network. The 10.1.1.X network represents an external network, while the 192.168.1.X network represents the virtual Ethernet LAN.

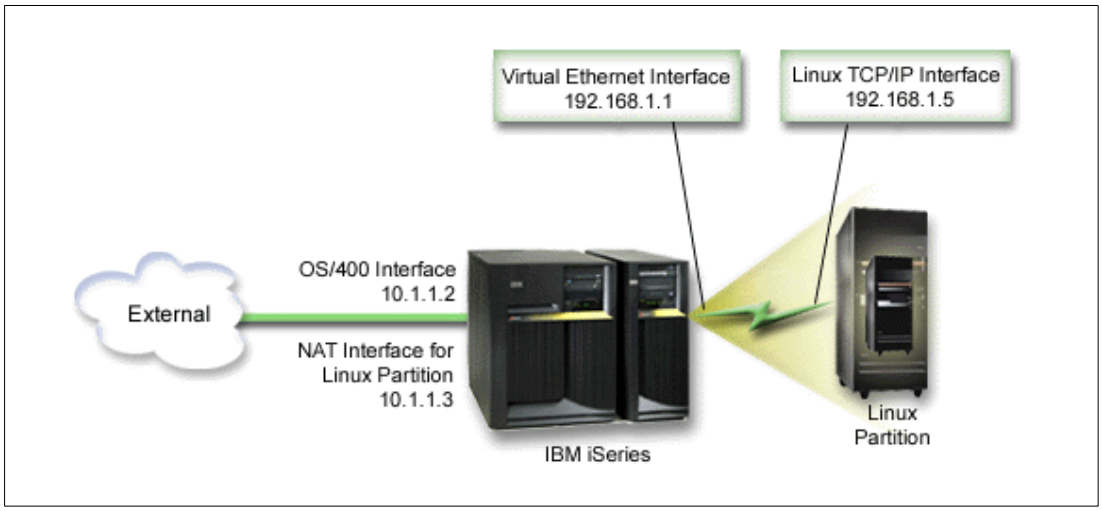


Figure 2-43 Network address translation example

In this example, any existing TCP/IP traffic for the iSeries hosting partition runs over the 10.1.1.2 interface. The 10.1.1.3 interface translates the Linux partition's traffic. The guest partition uses the virtual interface (192.168.1.1) and its own interface (192.168.1.5) to communicate with the hosting partition.

To make static NAT work, you need to first set up your OS/400 and Linux TCP/IP communications. Then you will create and apply some IP packet rules. Use the following procedure:

1. During the configuration of your guest partition, make sure you selected the option to create virtual LAN. This is described in 2.3.2, “Configuring a guest partition” on page 23.
2. Create an Ethernet line description for your virtual LAN. Note the port of the hardware resource, because you will need it when you set up TCP/IP in your guest partition. In the case of the above example, assume the hardware resource is CMN05 with a port of 0, and although arbitrary, the line description name is ETH0.

   ```
   CRTLINETH LIND(ETH0) RSRCNAME(CMN05) LINESPEED(1G) DUPLEX(*FULL)
   ```

3. Add a TCP/IP Interface for your virtual line description. Following the example, you could use:

   ```
   ADDTCPIFC INTNETADR('192.168.1.1') LIND('ETH0') SUBNETMASK('255.255.255.0')
   ```

   Start your new TCP/IP interface using the following command for this example:

   ```
   STRTCPIFC INTNETADR(yourIPaddress) or STRTCPIFC INTNETADR('192.168.1.1')
   ```

4. On your guest partition, set up networking using the instructions or tools provided by your Linux distribution. Be sure to use the correct IP address, subnet mask, port from step 2, and gateway IP address.

   During the TCP/IP setup in Linux, you need to know your network device. The network device is always ETH, plus the port number from step 2.

   In the example, note the following values:

   Interface IP address `192.168.1.5`
   Subnet mask `255.255.255.0`
   Gateway IP address `192.168.1.1`
   Network device `ETH0`

5. Create another TCP/IP interface that connects to the external network. It should use the same line description as your existing external TCP/IP interface. This interface will

eventually perform the address translation for your partition. Check to make sure your new interface communicates with the external LAN properly. In the case of the example, this interface has an IP address of 10.1.1.3 on a line description called ETHLINE.

6. Start the external TCP/IP interface:

   ```
   STRTCPIFC INTNETADR('10.1.1.3')
   ```

7. You should now verify that your virtual Ethernet connection works. From the guest partition, ping the Linux gateway, and from OS/400, ping the guest partition. If the ping is successful, continue.

8. Turn on IP datagram forwarding. This allows the OS/400 TCP/IP interfaces to pass IP packets between each other.

   ```
   CHGTCPA IPDTGFWD(*YES)
   ```

9. Connect to the hosting partition with Operations Navigator. You must not connect to the hosting partition with the NAT interface that you just created.

10. Navigate your way to Packet Rules. Use the Packet Rules interface to write at least three rules to enable static NAT. You need to create two New Defined Address rules and a New Mapped Address rule.

    a. In the Packet Rules window, create a new rules file by selecting **File->New File**.

    b. In the New Rules file menu, right-click **Defined Addresses** and select **New Defined Address**.

    c. Enter an address name, the IP address of the guest partition, and a type of **Trusted**. For the example, you would enter:

       Address Name LINUXPART
       Defined Address IP address where your IP address equals 192.168.1.5
       Type Trusted

    d. In the New Rules file menu, right-click **Defined Addresses** and select **New Defined Address**.

    e. Enter an address name, the IP address of the guest partition, and a type of **Border**. For the example, you would enter:

       Address Name SHELL
       Defined Address IP address where your IP address equals 10.1.1.3
       Type Border

    f. Expand the **Address Translation** menu item from the New Rules file menu window.

    g. Right-click **Mapped Addresses** and select **New Mapped Address**.

    h. Enter the Mapped address name, the To address name, and the line name. You can leave Journalling set to **off**.

       For the example, you would enter:

       Mapped address name LINUXPART
       To Address name SHELL
       Line ETHLINE
       Journalling OFF

    i. Verify your rules by selecting **File->Verify**.

    j. Save your rules file.

    k. Upon successful verification, select **File->Activate** from the menu. Your static NAT rules are now active.

11. To test outbound communications, ping an external host from your guest partition. Then from that external host, ping your guest partition to test inbound communications.

TCP/IP routing

You can also route traffic to your guest partitions through your iSeries server with various routing techniques. This solution is not difficult to configure on your iSeries but depending on the topology of your network, it may not be practical to implement. See the Figure 2-44.

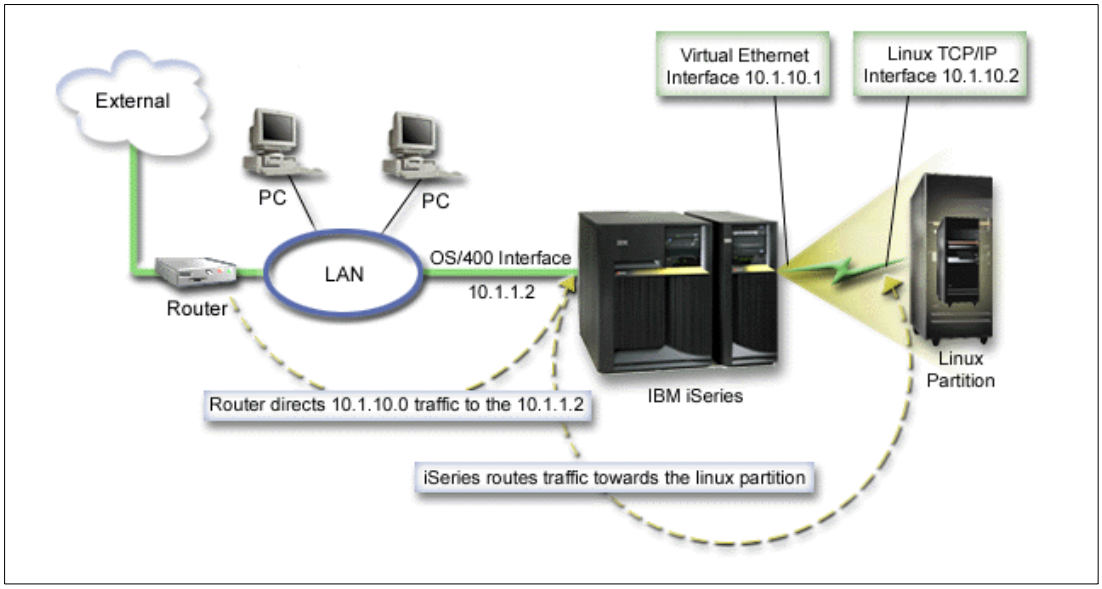


Figure 2-44 TCP/IP routing example

The existing TCP/IP interface (10.1.1.2) connects to the LAN. The LAN is connected to remote networks with a router. The Linux TCP/IP interface is addressed as 10.1.10.2 and the virtual LAN TCP/IP interface as 10.1.10.1. In OS/400, if you turn on IP Datagram Forwarding, the iSeries will route the IP packets to and from the guest partition. When you define your Linux TCP/IP connection, the router address must be 10.1.10.1.

The difficulty with this type of routing is getting the IP packets to the iSeries. In this scenario, you could define a route on the router so that it passes packets destined to the 10.1.10.0 network to the 10.1.1.2 interface. That works great for remote network clients. It would also work for the local LAN clients (clients connected to the same LAN as the iSeries) if they recognize that same router as their next hop. If they do not, then each client must have a route that directs 10.1.10.0 traffic to the iSeries 10.1.1.2 interface; therein starts the impracticability of this method. If you have hundreds of LAN clients, then you have to define hundreds of routes.

To configure a setup similar to this, follow these steps:

1. During the configuration of your guest partition, make sure you selected to create a virtual LAN. This is described in 2.3.2, “Configuring a guest partition” on page 23.
2. Create an Ethernet line description for your virtual LAN. Note the port of the hardware resource, because you will need it when you set up TCP/IP in your guest partition. In the case of the above example, assume the hardware resource is CMN05 with a port of 0, and although arbitrary, the line description name is ETH0.

   ```
   CRTLINETH LIND(ETH0) RSRCNAME(CMN05) LINESPEED(1G) DUPLEX(*FULL)
   ```

3. Add a TCP/IP Interface for your virtual line description. Following the example, you could use:

   ```
   ADDTCPIFC INTNETADR('10.1.10.1') LIND('ETH0') SUBNETMASK('255.255.255.0')
   ```

Start your new TCP/IP interface using the following command for the example:

```
STRTCPIFC INTNETADR(yourIPaddress) or STRTCPIFC INTNETADR('10.1.10.1')
```

4. On your guest partition, set up networking using the instructions or tools provided by your Linux distribution. Be sure to use the correct IP address, subnet mask, port, and gateway IP address. During the TCP/IP setup in Linux, you need to know your net or network device. The network device is always ETH, plus the port number from when you configured an Ethernet line description.

   In the example, use:

   Interface IP address `10.1.10.2`
   Subnet mask `255.255.255.0`
   Gateway IP address `10.1.10.1`
   Network device `ETH0`

5. Turn on IP datagram forwarding. This allows the OS/400 TCP/IP interfaces to pass IP packets between each other:

   ```
   CHGTCPA IPDTGFWD(*YES)
   ```

2.5.2 Using the virtual CD-ROM

The virtual CD-ROM is under /dev/viocd0 when using a non-devfs kernel and is found under the /dev/cdroms directory when using a devfs kernel. It is listed as /dev/cdroms/cdrom0 or cdrom1 and so on. This would be cdroms on the hosting partition. To use the CD-ROM in this case, you need to mount the device as follows:

- For a devfs kernel:

  ```
  mount /dev/cdroms/cdrom0 /mnt/cdrom
  ```

- For a non-devfs kernel:

  ```
  mount /dev/viocd0 /mnt/cdrom
  ```

Refer to 3.2.3, “iSeries-specific devices” on page 91, for more information on accessing these devices.

Restricting access to CD-ROM

There is a parameter in the NWSD that can restrict the optical or tape devices that can be used by Linux. See Figure 2-45 and Figure 2-46. When running CHGNWSD from the OS/400 command line, you see the display shown in Figure 2-45, which shows the options for restricting devices. The help text for this prompt is helpful for clarifying what the parameters are about.

```
                    Change Network Server Desc (CHGNWSD)

Type choices, press Enter.

TCP/IP route configuration:
  Route destination  . . . . . .   *NONE
  Subnet mask  . . . . . . . . .
  Next hop . . . . . . . . . . .
               + for more values
TCP/IP local host name . . . . .   *NWSD

TCP/IP local domain name . . . .   *SYS



TCP/IP name server system  . . .   *SYS
               + for more values
Restricted device resources  . .   *NONE         Name, *SAME, *NONE, *ALL...
               + for more values
Synchronize date and time  . . .   *NO           *SAME, *TYPE, *YES, *NO
                                                                      More...
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 2-45 Changing the network server description

When prompting on the Restricted device resources (RSTDDEVRSC) parameter, by setting the cursor to that input field and pressing F4, you see the display shown in Figure 2-46. The values that can be entered for this option are given here. The name parameter is the name of the device such as OPT01.

```
                  Specify Value for Parameter RSTDDEVRSC

Type choice, press Enter.

Type . . . . . . . . . . . . . :   NAME
Restricted device resources  . .   *SAME

Single Values
  *SAME
  *NONE
  *ALL
Other Values
  *ALLTAPE
  *ALLOPT









F3=Exit   F5=Refresh   F12=Cancel   F13=How to use this display   F24=More keys
```

Figure 2-46 Prompting the Restricted device resources (RSTDDEVRSC) parameter

2.5.3 Using virtual tape

Virtual tape shows up under /dev/viotape, where you see all OS/400 tape drives on the hosting partition by default. The devices available to Linux can be restricted in the same way as the optical drive in Figure 2-45 and Figure 2-46. An example of a command to backup a directory to tape would be:

- To archive a directory /etc to tape, enter:

  ```
  tar cvf /dev/viotape /etc
  ```

- To extract them, use (restore):

  ```
  tar xvf /dev/viotape
  ```

Important: Check documentation for supported tape drives. Remember to format the tape as a non-label tape.

See 3.10, “Backup and recovery” on page 121, for an in-depth discussion about backup and recovery commands.

2.6 Linux native adapters and disk (hosted or non-hosted partitions)

Fundamentally the procedure is the same to create a non-hosted or hosted partition. By definition, a non-hosted partition does not use virtual resources and therefore does not require a hosting partition to be defined when the partition is created. The major configuration task is removing resources from a partition and allocating them to a Linux partition. The fact that direct attached IOAs are being used is not unique to a non-hosted partition. Therefore, the considerations in the following sections are true also when allocating native I/O for use in a hosted partition.

2.6.1 Allocating resources

Adapters in Linux plug into the bus and communicate directly to device drivers. In contrast, iSeries adapters are connected to an input/output processor (IOP). The IOP provides the ability to offload tasks from the system processor to the IOP tasks such as identifying and communicating with the hardware. This is one of many features on the iSeries that contributes to its stability, reliability, and ease of use. Linux does not know about IOPs; therefore, it is necessary to emulate the PCI bus structure that Linux knows about to allow Linux to operate as it would in other environments. The result is what is known as “direct attach” adapters. These are adapters without an IOP attached.

With the new PCI technology, there are new rules for configuring the various features. There is more flexibility in the placement of IOPs and IOAs, but conversely it is more important to be aware of the rules associated with the configuration. There are *hard rules* that are enforced by the configurator and *soft rules* that qualify various configurations. There are some specific rules with regard to the placement and management of the direct attached cards. There are specific issues with card placement on the first bus of the CEC. This is particularly important to note with regard to Model 270. See “Hardware planning” on page 290 for more information.

Until the card is removed from iSeries control, it is governed by the normal rules for IOPs and IOAs in that box. When it is allocated to Linux, it will be ignored by OS/400 partitions and belong to a Linux partition as a direct attached IOA.

Adapter status

Adapters that are not attached to an IOP cannot be identified by the iSeries server, although it will recognize the fact that hardware is installed in the slot. Therefore, we must preface this discussion by briefly noting how the system denotes empty slots, as opposed to those that it cannot identify, and how it tags those adapters that belong to a Linux partition.

There are three slot states of concern:

- **Empty position**: The slot is empty. A slot that is not physically plugged is seen in the service tools as an *empty position*.
- **Occupied position**: A slot that has card in it, but is not attached to an IOP is seen as an *occupied position*, signifying that the system knows that there is a card there but cannot identify it. This might be due to the fact that there is physically no IOP installed or because it was removed from the IOP. In our case, it would be because it was either removed from the IOP in preparation for adding it to the Linux partition or because it was removed from the Linux partition presumably to return it to OS/400. In some scenarios, it might be the result of “stale” data that has yet to be cleaned up.
- **Exact device name**: With current PTFs, the correct device names are displayed. If the device shows up as *Generic Adapter*, then check for additional PTFs.

To reach this screen, follow these steps:

1. Go to SST (use STRSST).
2. Choose **5** (Work with system partitions).
3. Choose **1** (Display partition information).
4. Type **3** (Display allocated I/O resources).

   Then a screen similar to the example in Figure 2-47 is displayed.

```
                      Display Allocated I/O Resources
                                                          System:   AS03
System partition(s) to display . . . .  *ALL  *ALL, 0-1
Level of detail to display . . . . . .  *ALL  *ALL, *BUS, *IOP, *IOA, *DEV

Par                                              Serial      Part
ID   Description                    Type-Model   Number      Number
         Communications Chann       605A-001     00-1072020
     System Bus           23        283B-        38-0301080   04N4723
      Communications IOP            2890-001     10-81042    0000023L4306
       Communications IOA           2890-001     10-81042    0000023L4306
        Communications Port         2744-001     10-1010042  0000023L4288
        Communications Port         2744-001     10-82017    0000023L4288
        Virtual Port                6B00-001     10-81042    000008193654
     System Bus           24        283B-        38-0301080   04N4723
  1  LINUX    Partition              9406-270     10-5HYMM
     System Bus           23        283B-        38-0301080   04N4723
      Multiple Function IOA         2763-
     System Bus           24        283B-        38-0301080   04N4723
                                                                    More...
* Indicates load source.
F3=Exit   F5=Refresh          F6=Print   F9=Toggle empty pos
F10=Display logical address   F11=Display partition status   F12=Cancel
```

Figure 2-47 Adapter status

Adding and removing adapters

Before any resources can be removed from a partition and added to a Linux partition, the bus on which the resources are located must be *shared*. The guest partition cannot own a shared bus. A shared bus is the opposite of a dedicated bus. A dedicated bus means that all the resources on that bus are dedicated to the partition that owns the bus. A bus can not have its ownership changed while other partitions are using the resources. To change bus ownership, follow these steps:

1. Go to SST (use STRSST).
2. Choose **5** (Work with system partitions).
3. Choose **3** (Work with partition configuration). Then the display shown in Figure 2-48 appears.
4. Enter **5** before the partitions upon which you want to change the bus ownership. Press the Enter key.

```
                     Work with Partition Configuration
                                                            System:   SEATTLE
Type option, press Enter.
  1=Change partition name      2=Change partition processing resources
  3=Add I/O resources          4=Remove I/O resources
  5=Change bus ownership type  6=Select load source resource

        Partition
Option  Identifier  Name
          0         PRIMARY
          1         LINUX1
          2         LINUX2
          3         LINUX3









F3=Exit   F5=Refresh                       F9=Work with shared processor pool
F10=Work with Virtual LAN configuration    F11=Work with partition status
F12=Cancel                                 F23=More options
```

Figure 2-48 Working with the partition configuration to change bus ownership

5. Type **1** or **2** before the bus whose status you want to change as shown in Figure 2-49.

```
                        Change Bus Ownership Type
                                                              System:   SEATTLE
Level of detail to display . . . . . .  *ALL  *ALL, *BUS, *IOP, *IOA, *DEV

Partition identifier . . . . . . . . . . . . . . :  0
Partition name . . . . . . . . . . . . . . . . . :  PRIMARY

Type options, press Enter.
  1=Own bus dedicated  2=Own bus shared

     I/O Resource                                    Owning  Ownership
Opt  Description                    Type-Model      Par ID  Type
 2   System Bus            1         282D-             0     2
      Combined Function IOP   *<     284C-001          0     1
       Workstation IOA               2746-001          0     1
        Display Station              3487-0HC          0     1
       Communications IOA            2771-001          0     1
        Communications Port          2771-001          0     1
        Communications Port          2771-001          0     1
       Communications IOA            2744-001          0     1
        Communications Port          2744-001          0     1
                                                                        More...
F3=Exit   F9=Toggle empty pos   F10=Display serial/part numbers   F12=Cancel
```

Figure 2-49 Changing bus ownership

The process of allocating any of the adapters is essentially the same for any type of adapter, but there is a particular concern with allocating DASD. If the DASD are configured, the system will allow you to remove them, although it gives you warning messages. If you continue, there will be a problem that will require an IPL to repair it. If you intend to allocate DASD to a Linux partition, first verify that they are unconfigured and that device parity protection has been stopped. It they are not, you need to IPL to DST and remove them from the ASP.

There are two ways to remove and assign adapters. One is to use the LPAR configuration manager, and the other is to use concurrent maintenance.

- The first method we discuss is to use the LPAR configuration manager, which is the preferred method, because it cleans up stale configuration data. We strongly recommend you use this method for everything that needs to be done. There are two exceptions when it is necessary to use concurrent maintenance, which is discussed later. Using either method, the IOP must be removed in order to add the IOA. This may not be practical where an IOA is owned by a combined function IOP or a multifunction IOP that has other devices including DASD attached to it.

  For newly installed systems or systems acquiring and configuring new hardware, this is not a problem. The steps to remove an adapter using the Work with system partitions option are outlined here:

  a. Remove the IOP from the OS/400 partition. Start SST (STRSST), and select option **5** (Work with system partitions).

  b. Select option **3** (Work with Partition Configuration), as shown in Figure 2-50.

```
                    Work with Partition Configuration
                                                          System:   SEATTLE
Type option, press Enter.
  1=Change partition name      2=Change partition processing resources
  3=Add I/O resources          4=Remove I/O resources
  5=Change bus ownership type  6=Select load source resource

        Partition
Option  Identifier  Name
   4      0         PRIMARY
          1         LINUX1
          2         LINUX2
          3         LINUX3









F3=Exit   F5=Refresh                      F9=Work with shared processor pool
F10=Work with Virtual LAN configuration   F11=Work with partition status
F12=Cancel                                F23=More options
```

Figure 2-50 Work with Partition Configuration

c. Choose the partition that you want to remove resources from and type a **4** in front of it. Press the Enter key, and the display in Figure 2-51 appears.

```
                           Remove I/O Resources
                                                          System:   SEATTLE
Level of detail to display . . . . . .  *ALL  *ALL, *BUS, *IOP, *IOA, *DEV

Partition identifier . . . . . . . . . . . . . :  0
Partition name . . . . . . . . . . . . . . . . :  PRIMARY

Type options, press Enter.
  1=Remove  2=Remove and clear hardware resource(s)

     I/O Resource                                  Serial      Part
Opt  Description                    Type-Model     Number      Number
     System Bus            1        282D-         E8-0261078   24L1460
  2   Combined Function IOP   *<    284C-001      E8-0261078   24L1460
       Workstation IOA              2746-001      10-0244401   0000021H5497
        Display Station             3487-0HC      00-*******
       Communications IOA           2771-001      10-0257321   0000021P4151
        Communications Port         2771-001      10-0257321   0000021P4151
        Communications Port         2771-001      10-0257321   0000021P4151
       Communications IOA           2744-001      10-61036     0000023L4288
                                                                        More...
* Indicates load source.
F3=Exit   F9=Toggle empty pos   F10=Display logical address   F12=Cancel
```

Figure 2-51 Removing adapters

d. Find the IOP that you want to remove, and complete these steps:

i. Type **1** in front of the I/O resource if the resource needs to return to the primary or hosting partition.

ii. Type **2** in front of the I/O resource to remove the resource permanently, and delete the resource description.

The display shown in Figure 2-52 confirms your choice.

```
                        Confirm Remove I/O Resources
                                                              System:   SEATTLE
Press Enter to confirm your choice to remove the following
  I/O resources from this partition.  Removals can only be
  made at a bus or I/O processor level.
Press F12 to return to change your choice.

  Partition identifier . . . . . . . . . . . . . :  0
  Partition name . . . . . . . . . . . . . . . . :  PRIMARY

I/O Resource                                 Serial      Part
Description                    Type-Model    Number      Number
 Combined Function IOP         2843-001      10-0271195  0000004N5095
  Workstation IOA              2746-001      10-9265123  0000021H5497
  Communications IOA           2744-001      10-0167047  0000023L4288
   Communications Port         2744-001      10-0167047  0000023L4288
  Multiple Function IOA        2748-001      10-0145071  0000004N2255
   Disk Unit                   6718-050      68-613E2    09L3932
   Disk Unit                   6718-050      68-6194D    09L3932
   Disk Unit                   6718-050      68-664D5    09L3932
                                                                  More...
* Indicates load source.
F10=Display logical address   F12=Cancel
```

Figure 2-52 Confirming the removal of IOP

e. If there are resources attached the multi-function IOP that are active, you will receive a warning that allows you to cancel and take care of the busy resource. Otherwise the display is returned notifying you that the resource was successfully removed. See Figure 2-53.

```
                              Remove I/O Resources
                                                              System:   SEATTLE
Level of detail to display . . . . . .  *ALL  *ALL, *BUS, *IOP, *IOA, *DEV

Partition identifier . . . . . . . . . . . . . . :  0
Partition name . . . . . . . . . . . . . . . . . :  PRIMARY

Type options, press Enter.
  1=Remove  2=Remove and clear hardware resource(s)

     I/O Resource                                    Serial      Part
Opt  Description                      Type-Model     Number      Number
     System Bus            1          282D-          E8-0261078   24L1460
      Combined Function IOP   *<      284C-001       E8-0261078   24L1460
       Workstation IOA                2746-001       10-0244401  0000021H5497
        Display Station               3487-0HC       00-*******
       Communications IOA             2771-001       10-0257321  0000021P4151
        Communications Port           2771-001       10-0257321  0000021P4151
        Communications Port           2771-001       10-0257321  0000021P4151
       Communications IOA             2744-001       10-61036    0000023L4288
                                                                            More...
* Indicates load source.
F3=Exit   F9=Toggle empty pos   F10=Display logical address   F12=Cancel
Remove I/O resource(s) was successful.
```

Figure 2-53 Notification of successful adapter removal

f. When the IOP is removed, return to the Work with Partition Configuration display. Type **3** in front of the partition to which you want to add the resource. See Figure 2-54.

```
                      Work with Partition Configuration
                                                              System:   SEATTLE
Type option, press Enter.
  1=Change partition name      2=Change partition processing resources
  3=Add I/O resources          4=Remove I/O resources
  5=Change bus ownership type  6=Select load source resource

        Partition
Option  Identifier  Name
          0         PRIMARY
          1         LINUX1
          2         LINUX2
  3       3         LINUX3









F3=Exit   F5=Refresh                      F9=Work with shared processor pool
F10=Work with Virtual LAN configuration   F11=Work with partition status
F12=Cancel                                F23=More options
```

Figure 2-54 Adding I/O resources to a Linux partition

g. If the bus is not shared, you need to use option **3** (Use bus shared) on the bus and option **1** (Own dedicated) on the IOA. Otherwise the only option required would be to dedicate the adapter. See Figure 2-55.

```
                          Add I/O Resources
                                                          System:   SEATTLE
Level of detail to display . . . . . *ALL  *ALL, *BUS, *IOP, *IOA, *DEV

Partition identifier . . . . . . . . . . . . . :  3
Partition name . . . . . . . . . . . . . . . . :  LINUX3

Type options, press Enter.
  1=Own dedicated  2=Own bus shared  3=Use bus shared

     I/O Resource                              Serial      Part
Opt  Description                  Type-Model   Number      Number
     System Bus            1      282D-        E8-0261078   24L1460
 3   System Bus            2      282D-        E8-0261078   24L1460
      Combined Function IOP       2843-001     10-0271195  0000004N5095
       Workstation IOA            2746-001     10-9265123  0000021H5497
       Communications IOA         2744-001     10-0167047  0000023L4288
        Communications Port       2744-001     10-0167047  0000023L4288
       Communications IOA         2838-001     10-0264282  0000021H5460
        Communications Port       2838-001     10-0264282  0000021H5460
 1     Multiple Function IOA      2748-001     10-0145071  0000004N2255
                                                                   More...
F3=Exit   F9=Toggle empty pos   F10=Display logical address   F12=Cancel
```

Figure 2-55 Adding an adapter and sharing the bus

The next display (Figure 2-56) confirms the selection process.

```
                       Confirm Add I/O Resources
                                                        System:   SEATTLE
Press Enter to confirm your choice to add the following
  I/O resources to this partition.
Press F12 to return to change your choice.

  Partition identifier . . . . . . . . . . . . . :  3
  Partition name . . . . . . . . . . . . . . . . :  LINUX3

     I/O Resource                              Serial      Part
Opt  Description                  Type-Model   Number      Number
 3   System Bus            2      282D-        E8-0261078   24L1460
 1     Multiple Function IOA      2748-001     10-0145071  0000004N2255
 1      Disk Unit                 6718-050     68-613E2    09L3932
 1      Disk Unit                 6718-050     68-6194D    09L3932
 1      Disk Unit                 6718-050     68-664D5    09L3932
 1      Disk Unit                 6718-050     68-46776    09L3932
 1      Device Services           283F-001     38-02299    0000004N2472
 1      Disk Unit                 6718-050     68-62318    09L3932
 1      Disk Unit                 6718-050     68-6128C    09L3932


F10=Display logical address   F12=Cancel
```

Figure 2-56 Confirming the selection of resources

h. There is a message that the resource was successfully added and the resource is seen on the partition that it was added to (Figure 2-57).

```
                      Display Allocated I/O Resources
                                                            System:   AS03
System partition(s) to display . . . .  *ALL  *ALL, 0-1
Level of detail to display . . . . . .  *ALL  *ALL, *BUS, *IOP, *IOA, *DEV

Par                                              Serial      Part
ID   Description                    Type-Model   Number      Number
         Communications Chann       605A-001     00-1072020
     System Bus           23        283B-        38-0301080   04N4723
      Communications IOP            2890-001     10-81042    0000023L4306
       Communications IOA           2890-001     10-81042    0000023L4306
        Communications Port         2744-001     10-1010042  0000023L4288
        Communications Port         2744-001     10-82017    0000023L4288
        Virtual Port                6B00-001     10-81042    000008193654
     System Bus           24        283B-        38-0301080   04N4723
  1  LINUX    Partition              9406-270     10-5HYMM
     System Bus           23        283B-        38-0301080   04N4723
      Multiple Function IOA         2763-
     System Bus           24        283B-        38-0301080   04N4723
                                                                    More...
* Indicates load source.
F3=Exit   F5=Refresh          F6=Print   F9=Toggle empty pos
F10=Display logical address   F11=Display partition status   F12=Cancel
```

Figure 2-57 Displaying the allocated resources

i. Sometimes it is confusing whether the adapter is the correct one and whether it was assigned correctly. A very useful function on the Display Partition Information screen is option 5 (Display system I/O resources). The F10 key toggles to the logical address as shown in Figure 2-58.

```
                      Display Allocated I/O Resources
                                                            System:   AS03
System partition(s) to display . . . .  *ALL  *ALL, 0-1
Level of detail to display . . . . . .  *ALL  *ALL, *BUS, *IOP, *IOA, *DEV

Par                                    Type-
ID   Description                       Model     Logical Address
        Communications Chann           605A-001  2/  1/0/ 21-1/  /14/ 7/ 0/ 0/
     System Bus           23           283B-     2/ 23/ /   - /  /  /  /  /  /
      Communications IOP                2890-001  2/ 23/0/ 18- /  /  /  /  /  /
       Communications IOA               2890-001  2/ 23/0/ 18-1/  /14/ 2/  /  /
        Communications Port             2744-001  2/ 23/0/ 18-1/  /14/ 2/ 4/  /
        Communications Port             2744-001  2/ 23/0/ 18-1/  /14/ 2/ 6/  /
        Virtual Port                    6B00-001  2/ 23/0/ 18-1/  /14/ 2/15/  /
     System Bus           24           283B-     2/ 24/ /   - /  /  /  /  /  /
  1  LINUX    Partition                 9406-270
     System Bus           23           283B-     2/ 23/ /   - /  /  /  /  /  /
      Multiple Function IOA             2763-     2/ 23/0/ 17- /  /  /  /  /  /
     System Bus           24           283B-     2/ 24/ /   - /  /  /  /  /  /
                                                                        More...
* Indicates load source.
F3=Exit   F5=Refresh              F6=Print   F9=Toggle empty pos
F10=Display serial/part numbers   F11=Display partition status   F12=Cancel
```

Figure 2-58 Using F10 to toggle to the logical address

j. Toggling F10 shows the status and ownership information. Finally, it displays the using partitions. This display option is very useful in sorting out partition resource information. Refer to Figure 2-59.

```
                      Display Allocated I/O Resources
                                                            System:   AS03
System partition(s) to display . . . .  *ALL  *ALL, 0-1
Level of detail to display . . . . . .  *ALL  *ALL, *BUS, *IOP, *IOA, *DEV

Par                                                Serial      Part
ID   Description                       Type-Model  Number      Number
        Communications Chann           605A-001    00-1072020
     System Bus           23           283B-       38-0301080   04N4723
      Communications IOP                2890-001    10-81042    0000023L4306
       Communications IOA               2890-001    10-81042    0000023L4306
        Communications Port             2744-001    10-1010042  0000023L4288
        Communications Port             2744-001    10-82017    0000023L4288
        Virtual Port                    6B00-001    10-81042    000008193654
     System Bus           24           283B-       38-0301080   04N4723
  1  LINUX    Partition                 9406-270    10-5HYMM
     System Bus           23           283B-       38-0301080   04N4723
      Multiple Function IOA             2763-
     System Bus           24           283B-       38-0301080   04N4723
                                                                        More...
* Indicates load source.
F3=Exit   F5=Refresh          F6=Print   F9=Toggle empty pos
F10=Display logical address   F11=Display partition status   F12=Cancel
```

Figure 2-59 Verifying the resource status

k. The last step is to return to the Work with Partition Configuration display and add back the IOP to the OS/400 partition. Any additional adapters that were attached to it will be returned with it.

- The second way to remove and assign adapters is to use *concurrent maintenance*. This option is accessed by way of the Hardware Service Manager and is intended for experts. There are two specific instances when it is necessary to use this, but if you do not converse much with the options or are not comfortable using them, we recommend that you contact your customer engineer.

  The specific instances that might be encountered are:

  – You want to remove an IOA and return it to the OS/400 partition
  – You have a multi-function IOP that has an adapter that you want to remove without removing the IOP

  This normally occurs when a DASD IOA is attached to the IOP. If the DASD are configured, it will be necessary to IPL to DST and unconfigure them. But, if you had a communication adapter that you wanted to remove, this would be one option. There is almost no reason to do this on a new system or if you are acquiring new resources for a Linux partition. If you are redistributing resources, then it may be necessary. You must consider consulting your customer engineer if you need to use this approach.

  The procedure is outlined in the following steps and corresponding displays:

  a. Start a service tool (STRSST) as shown in Figure 2-60.

  b. Choose option **7** (Hardware service manager).

```
                        Start a Service Tool

Warning: Incorrect use of this service tool can cause damage
to data in this system.  Contact your service representative
for assistance.

Select one of the following:

     1. Product activity log
     2. Trace Licensed Internal Code
     3. Work with communications trace
     4. Display/Alter/Dump
     5. Licensed Internal Code log
     6. Main storage dump manager
     7. Hardware service manager




Selection


F3=Exit         F12=Cancel         F16=SST menu
```

Figure 2-60 Choosing the HSM to use concurrent maintenance

This option accesses the display shown in Figure 2-61.

c. Select option **1** (Packaging hardware resources (systems, frames, cards,...)) as shown in Figure 2-61.

```
                          Hardware Service Manager

Attention:  This utility is provided for service representative use only.

   System unit . . . . . :   9406-820 10-5310M
   Release . . . . . :   V5R1M0 (1)

Select one of the following:

   1. Packaging hardware resources (systems, frames, cards,...)
   2. Logical hardware resources (buses, IOPs, controllers,...)
   3. Locate resource by resource name
   4. Failed and non-reporting hardware resources
   5. System power control network (SPCN)
   6. Work with service action log
   7. Display label location work sheet
   8. Device Concurrent Maintenance

Selection


F3=Exit      F6=Print configuration      F9=Display card gap information
F10=Display resources requiring attention      F12=Cancel
```

Figure 2-61 Packaging option to access concurrent maintenance

d. The display in Figure 2-62 appears. Find the bus that has the resources and type **9** in front of it.

```
                       Packaging Hardware Resources

                                  Local system type . . . . :  9406
                                  Local system serial number:  10-5310M

Type options, press Enter.
  2=Change detail    3=Concurrent Maintenance    4=Remove    5=Display Detail
  8=Associated logical resource(s)    9=Hardware contained within package

Opt Description                     Type-Model  Frame ID  Resource Name
    System                           9406-820              SYS01
     System Unit             +       9406-820        1     FR01









F3=Exit    F5=Refresh    F6=Print    F8=Exclude non-reporting resources
F9=Reserve frame space        F10=Non-reporting resources
F11=Display SPCN system information   F12=Cancel   F13=Unresolved locations
There are resources with unresolved locations.  Press F13 to see list.
```

Figure 2-62 Displaying hardware associated with the bus

e. The display shown in Figure 2-63 appears. Choose **3** (Concurrent maintenance) for the required IOA.

```
                      Packaging Hardware Resources
                                                              Frame ID:  1
Type options, press Enter.
  2=Change detail    3=Concurrent Maintenance    4=Remove    5=Display Detail
  8=Associated logical resource(s)    9=Hardware contained within package

                                                Resource    Card   Device
Opt Description                      Type-Model Name        Pos    Pos
      Communications IOA              2771-001  P39         C04
 3    Multiple Function IOA           2748-001  P21         C05
      Workstation IOA                 2746-001  P36         C06
      Occupied Position                                     C07
      Communications Port     ?       2838-001  P44         C07
      Occupied Position                                     C08
      Multiple Function IOA   ?       2748-001  P28         C08
      Multiple Function IOA   ?       2768-001  P37         C09
      Communications Port             2744-001  P40         C10
      Workstation IOA                 2746-001  P38         C11
                                                                    More...

F3=Exit     F5=Refresh     F6=Print     F7=Include empty positions
F8=Exclude non-reporting resources      F10=Non-reporting resources
F12=Cancel                              F13=Unresolved locations
```

Figure 2-63 Selecting the IOA to remove

f. The display in Figure 2-64 appears. In this example, we selected a DASD IOA that previously should have been verified that the DASD was not configured. Now relinquish the IOA from the controlling resource. In this case, the Combined function IOP owns the adapter; type **9** in front of the IOP.

```
                  Hardware Resource Concurrent Maintenance
                                                              Frame ID:  1
Type options, press Enter.
  2=Toggle LED blink off/on            5=Display detail
  8=Associated logical resource(s)     9=Work with controlling resource

                                                Power       Card   Device
Opt Description                      Type-Model Status      Pos    Pos
 9   Multiple Function IOA         >  2748-001   On          C05




F3=Exit       F5=Refresh      F6=Print      F8=Display resource names
F9=Power off domain   F10=Power on domain   F11=In-use resources   F12=Cancel
```

Figure 2-64 Concurrent maintenance IOA

g. The display in Figure 2-65 appears. At this point, the adapter should appear as being in “occupied position”.

```
                      Work with Controlling Resource
                                                                Frame ID:  1
Selected resource:
                                              Resource    Card   Device
    Description                    Type-Model  Name        Pos    Pos
    Multiple Function IOA           2748-001   P21         C05

Type options, press Enter.
  5=Display detail     6=Relinquish from     7=Assign to
  8=Associated packaging resource(s)

                                                             Resource
Opt Description                    Type-Model  Status          Name
    Combined Function IOP        *> 284C-001   Operational     CMB01









F3=Exit      F5=Refresh      F6=Print      F12=Cancel
```

Figure 2-65 Working with controlling resource

h. The adapter can be removed from the partition by using the Work with Partition Configuration display (Figure 2-66). Type **4** in front of the partition from which to remove the resources.

```
                     Work with Partition Configuration
                                                          System:   SEATTLE
Type option, press Enter.
  1=Change partition name      2=Change partition processing resources
  3=Add I/O resources          4=Remove I/O resources
  5=Change bus ownership type  6=Select load source resource

        Partition
Option  Identifier  Name
  4       0         PRIMARY
          1         LINUX1
          2         LINUX2
          3         LINUX3









F3=Exit   F5=Refresh                      F9=Work with shared processor pool
F10=Work with Virtual LAN configuration   F11=Work with partition status
F12=Cancel                                F23=More options
```

Figure 2-66 Selecting the partition from which to remove resources

i. When removing resources, there are two options as shown in Figure 2-67:

- If you will remove the resource and permanently assign it to another partition, choose **2** (Remove and clear hardware resource(s)).
- If you are removing a tape drive or CD-ROM to temporarily assign it to a partition, then select **1** (Remove) to keep the configuration information for that resource. Using this option does not preclude returning the resource. It merely allows you to return it to the same previous configuration.

Notice in the same figure that there are three occupied slots and identifying information is not displayed with the resources. You need to verify the resources by displaying the logical address and verifying to what it corresponds.

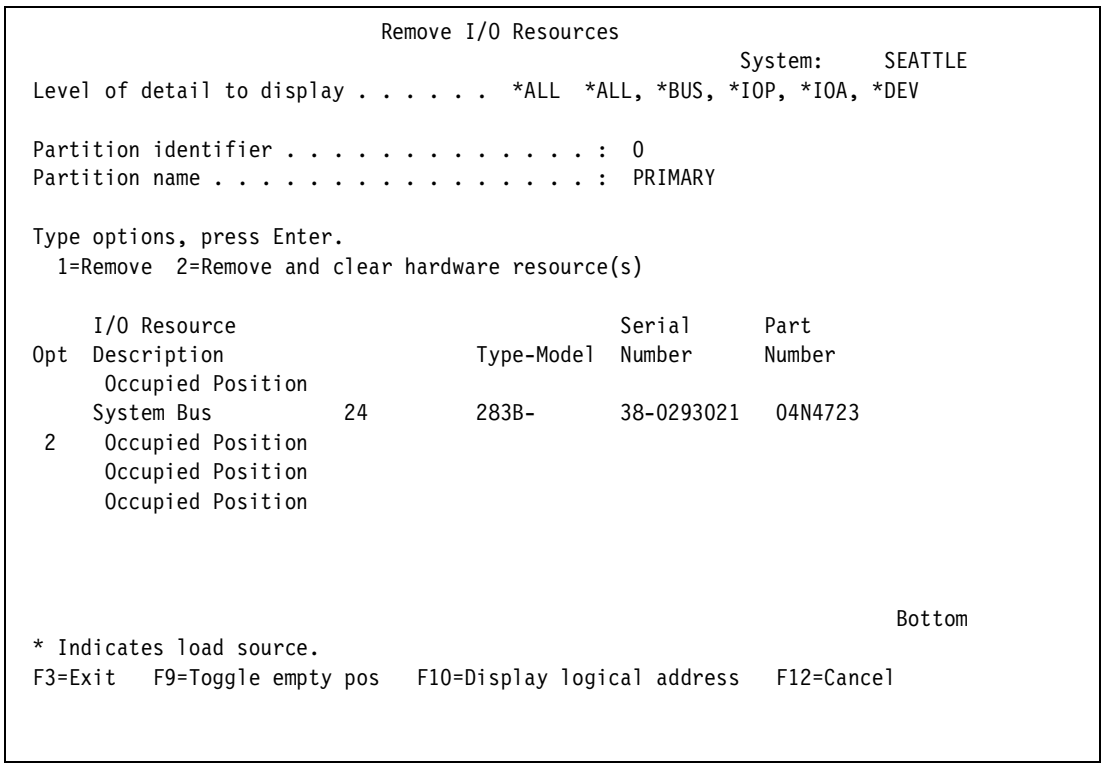


Figure 2-67 Removing resources

The display in Figure 2-68 allows you to verify that this is what you want to do.

```
                        Confirm Remove I/O Resources
                                                              System:   SEATTLE
Press Enter to confirm your choice to remove the following
  I/O resources from this partition.  Removals can only be
  made at a bus or I/O processor level.
Press F12 to return to change your choice.

  Partition identifier . . . . . . . . . . . . . . :  0
  Partition name . . . . . . . . . . . . . . . . . :  PRIMARY

I/O Resource                                    Serial      Part
Description                     Type-Model      Number      Number
 Occupied Position

* Indicates load source.
F10=Display logical address   F12=Cancel
```

Figure 2-68 Confirming the removal of a resource

j. Press Enter to confirm your choice.

k. To add the resources that we just removed, you would use the LPAR configuration manager and choose **Work with partition configuration**. On the display shown in Figure 2-69, place **3** next to the partition to which to add the resource.

```
                      Work with Partition Configuration
                                                              System:   SEATTLE
Type option, press Enter.
  1=Change partition name      2=Change partition processing resources
  3=Add I/O resources          4=Remove I/O resources
  5=Change bus ownership type  6=Select load source resource

         Partition
Option   Identifier  Name
           0         PRIMARY
  3        1         LINUX1
           2         LINUX2




F3=Exit   F5=Refresh                       F9=Work with shared processor pool
F10=Work with Virtual LAN configuration    F11=Work with partition status
F12=Cancel                                 F23=More options
```

Figure 2-69 Choosing a partition to add resources to LINUX1

l. Press Enter and the display in Figure 2-70 appears. The only option that applies to an IOA is option **1** (Own dedicated). Select this option.

```
                              Add I/O Resources
                                                                System:   SEATTLE
Level of detail to display . . . . . .  *ALL  *ALL, *BUS, *IOP, *IOA, *DEV

Partition identifier . . . . . . . . . . . . . :  1
Partition name . . . . . . . . . . . . . . . . :  LINUX1

Type options, press Enter.
  1=Own dedicated  2=Own bus shared  3=Use bus shared

     I/O Resource                                  Serial      Part
Opt  Description                    Type-Model     Number      Number
     System Bus           23         283B-         38-0293021   04N4723
     System Bus           24         283B-         38-0293021   04N4723
 1    Occupied Position

F3=Exit   F9=Toggle empty pos   F10=Display logical address   F12=Cancel
```

Figure 2-70 Adding an IOA

m. You can then confirm your choice as shown in Figure 2-71.

```
                         Confirm Add I/O Resources
                                                           System:   SEATTLE
Press Enter to confirm your choice to add the following
  I/O resources to this partition.
Press F12 to return to change your choice.

  Partition identifier . . . . . . . . . . . . . :  1
  Partition name . . . . . . . . . . . . . . . . :  LINUX1

     I/O Resource                                  Serial      Part
Opt  Description                    Type-Model     Number      Number
 1    Occupied Position

F10=Display logical address   F12=Cancel
```

Figure 2-71 Confirming the addition of an IOA

n. After you confirm your choice, navigate to the main configuration screen (Figure 2-72).

```
                        Work with System Partitions
                                                              System:   SEATTLE
Attention:  Incorrect use of this utility can cause damage
to data in this system.  See service documentation.

  Number of partitions  . . . . . :  3
  Partition manager release . . . . . :  V5R1M0 L000

  Partition identifier  . . . . . :  0
  Partition name  . . . . . :  PRIMARY   *

Select one of the following:

     1. Display partition information
     2. Work with partition status
     3. Work with partition configuration
     4. Recover configuration data
     5. Create a new partition

Selection


F3=Exit   F12=Cancel
```

Figure 2-72 Displaying partition information

o. Select option **1** (Display partition information) to confirm the successful addition. Choose option **3** (Display allocated I/O resources). The display in Figure 2-73 appears.

```
                        Display Partition Information
                                                              System:   SEATTLE
Select one of the following:

     1. Display partition status
     2. Display partition processing configuration
     3. Display allocated I/O resources
     4. Display available I/O resources
     5. Display system I/O resources
     6. Display partition operating environment
     7. Display communication options
     8. Display secondary partition reference code history
     9. Display guest environment host information
    10. Display guest environment console log




Selection
    3

F3=Exit   F12=Cancel
```

Figure 2-73 Displaying allocated resources

p. Select **3** (Display allocated I/O resources). Selected adapters are displayed on the bus (Figure 2-74).

```
                      Display Allocated I/O Resources
                                                              System:   AS03
System partition(s) to display . . . .  *ALL  *ALL, 0-1
Level of detail to display . . . . . .  *ALL  *ALL, *BUS, *IOP, *IOA, *DEV

Par                                                Serial      Part
ID   Description                      Type-Model   Number      Number
         Communications Chann         605A-001     00-1072020
     System Bus           23          283B-        38-0301080   04N4723
      Communications IOP              2890-001     10-81042    0000023L4306
       Communications IOA             2890-001     10-81042    0000023L4306
        Communications Port           2744-001     10-1010042  0000023L4288
        Communications Port           2744-001     10-82017    0000023L4288
        Virtual Port                  6B00-001     10-81042    000008193654
     System Bus           24          283B-        38-0301080   04N4723
  1  LINUX    Partition                9406-270     10-5HYMM
     System Bus           23          283B-        38-0301080   04N4723
      Multiple Function IOA           2763-
     System Bus           24          283B-        38-0301080   04N4723
                                                                        More...
* Indicates load source.
F3=Exit   F5=Refresh          F6=Print   F9=Toggle empty pos
F10=Display logical address   F11=Display partition status   F12=Cancel
```

Figure 2-74 Displaying the IOA that was added

Adding new adapters

The previous examples used existing resources. New adapters are automatically allocated by an upstream IOP on a bus owned by OS/400. If an IOP is not present, the adapter is identified as *occupied position*.

The default owner of the resource would be the owner of the bus. If the bus is owned by the *guest partition*, then no further action is required. If the bus is owned by OS/400, then the adapter must be removed, and added to the Linux partition. Before adding a new IOA, remove an “empty slot” and add it to the Linux partition. This should be done to insure that an IOP does not automatically allocate the card causing extra steps to remove it from the IOP.

Reassigning adapters

To reassign an adapter to an IOP, remove it from the guest partition. By using *concurrent maintenance* under the Hardware Service Manager, the adapter is assigned to an IOP. On a system IPL, normal rules prevail and the adapter is assigned to the IOP.

2.6.2 Native SCSI support

Native SCSI support is implemented through a Linux SCSI driver that has direct access to the a PCI card, and thereby allows for communication with DASD devices. The driver, which is called `ibmsis`, is proprietary (partly binary only), but distributable. It is part of the distributions and is available at in the Developer Resources section at:
http://www.ibm.com/servers/eserver/iseries/linux

The driver is accompanied by a utility to configure direct attached DASD called **sisconfig**, which presents an OS/400-like screen to configure disks within Linux. In addition, the **sisupdate** utility is available to update IOA and device firmware. For more information, see Chapter 4, “Advanced administration and development” on page 137.

The SCSI driver itself is ibmsis.o. The following Linux drivers are also required:

- **sd.o**: SCSI DASD driver
- **st.o**: SCSI Tape driver
- **sr.o**: SCSI CD-ROM driver
- **scsi_mod.o**: SCSI driver

Important: When using native disks, they must not have parity configuration turned on. If they are new disks and are being added to the iSeries, then do not add parity protection to them. If they were being used in an ASP that had parity turned on, then go to DST and remove parity before loading Linux. With SuSE, the parity can be unconfigured with **sisconfig**, which pops up during the installation.

2.6.3 Native LAN adapters

If you have multiple LAN adapters and can dedicate one or more to Linux, you might consider doing so. A dedicated adapter eliminates the extra hop that must be taken when using the virtual LAN to communicate with the network. There are additional routing and subnet issues that were described in the virtual LAN configuration. Once the hardware is allocated, it is a simple matter to configure in Linux, since OS/400 does not know of the existence of the hardware. The configuration for the native adapter is contained within Linux. There are no OS/400 objects to configure, so it is strictly a matter of using the **ifconfig** and **route** commands for basic configuration of Linux networking. For more details, see 3.4, “Working with network devices (virtual and direct)” on page 101, or refer to the distributors’ documentation on how to use the configuration tools that are delivered with your distribution.

2.6.4 Configuring for a non-hosted partition

The current distributions contain the **ibmsis.o** device driver that is loaded in the installation process. To boot from the native DASD, it is necessary to use the Linux **dd** command to put the kernel *with* an attached RAM disk into /proc/iSeries/mf/[A|B]/vmlinux. Refer to the distributors installation instructions because they might have already done this automatically for you.

2.7 Troubleshooting

If the installation completed satisfactorily, the problem is likely to be with Linux itself. Linux is supported by the distributor from whom the code was acquired, IBM, or via various Internet forums.

When doing problem determination, consider the following tips:

- Problems creating a guest partition?
  - The problems most likely to occur in this area are adding or removing resources. Consider these points:

- Who owns the bus that the resource is located on, and is that bus owned shared?
- If you are using Work with System Partitions (recommended), have you removed the IOP? Using this method, the IOP must be removed and then re-added after the resource was removed.
- Is the IOP a combined function IOP or a multi-function IOP? If so, are there other resources that will be affected by removing the IOP, for example, DASD?
- If you are removing DASD, are they configured? If not, they need to be removed from the ASP (unconfigured).
- Are you allocating a slot to install a new IOA, or are you removing an existing IOA?

► Is a NWSD present?

Use the WRKNWSD or WRKCFGSTS *NWS command.

– If yes and it is type *GUEST, continue.
– If no, see the definition and discussion in 2.4.3, “Creating an initial network server description (CRTNWSD)” on page 38.

► Is NWSD active?

– If yes, go to “Virtual LAN” on page 84.
– If no:

i. Check the QSYSOPR messages for error message CPDB1AD and the return code. To see these messages, use the following command on the hosting partition:

```
DSPMSG QSYSOPR
```

Find the explanation for the reason code in Table 2-1.

Table 2-1 Reason codes for message CPDB1AD

Reason Code	Explanation
00000001	*NWSSTG was specified as the IPL source but no storage space was found.
00000002	The partition specified in the PARTITION parameter was not found.
00000003	The partition specified in the PARTiTION parameter is not a GUEST partition.
00000004	There is already a NWSD in this OS/400 partition that is active and using the partition specified in the PARTITION parameter of the NWSD.
00000005	The partition specified in the PARTITION parameter of the NWSD is powered on (perhaps through the LPAR configuration interface or from another OS/400 partition).
00000006	The partition is set to boot from a stream file (stmf), and for some reason, that didn’t work. You should note that the user performing the vary on needs READ access to the IPL STMF parameter.
00000007	The NWSD is set to boot from a storage space (NWSSTG), but for some reason, the kernel couldn’t be found (no type 0x41 partition? not marked bootable).
00000008	The partition does not start. There are a variety of reasons why the partition does not start. You could have a corrupt kernel or the processor feature code does not support the shared processor pool. If the kernel and processor are not the problem, then start reviewing the information for the partition and the SRCs.
00000009	The partition identified as the hosted partition is not configured.
00000010 00000011 00000080	Contact your next level of support to find a proper solution to this problem.

Reason Code	Explanation
00001088 00001089 0000108A	The kernel does not appear to be valid. This error is frequently caused if you do not FTP the kernel in binary mode.
0000108B 0000108C	The kernel does not appear to be compatible with the version of OS/400 in the primary partition.
000010A3 000010A9 000010AA	There is an insufficient amount of processor assigned to the partition, or there are not enough shared processors available.
000010A4 000010A5	There is an insufficient amount of memory available for the partition.
000010AE	This error occurs on systems that support dedicated processors when either you have specified a shared processor for a Linux partition or you have the QPRCMLTTSK system value set to 1.

ii. Verify that the appropriate storage spaces are linked. Use the WRKNWSSTG command.

iii. Verify that the IPL source and IPL path are correct.

iv. Verify that the host partition is set in the partition configuration.

v. Are there IPL parameters defined? If so, verify that these are correct.

vi. Check the secondary partition SRCs using the display secondary partition reference code history screen. Please note that the sequence of reference codes in Figure 2-75 is a normal bring up, and the PROG XXXX codes are the kernel initialization. PROG FFFF means that all is well.

```
              Display Secondary Partition Reference Code History
                                                            System:   SEATTLE
Secondary partition(s) to display . . . . .  *ALL  *ALL,   1-3       04/05/01
Number of reference codes to display  . . .  200   1-200             09:20:26

Partition
Identifier  Name      Reference Codes  Date      Time
  1         LINUX1    PROG FFFF        04/04/01  12:55:38
                      PROG 3EAB        04/04/01  12:55:37
                      PROG 3EAB        04/04/01  12:55:37
                      PROG 3EAB        04/04/01  12:55:37
                      PROG 0301        04/04/01  12:55:37
                      PROG 0205        04/04/01  12:55:37
                      PROG 0105        04/04/01  12:55:37
                      PROG 0105        04/04/01  12:55:37
                      C200 82FF        04/04/01  12:55:37
                      C200 8200        04/04/01  12:55:37
                      C200 81FF        04/04/01  12:55:37
                      C200 8120        04/04/01  12:55:37
                      C200 8110        04/04/01  12:55:37
                                                                    More...
F3=Exit   F5=Refresh   F6=Print   F8=Clear   F9=Include reference code detail
F10=Monitor reference code history          F12=Cancel
```

Figure 2-75 Displaying reference codes for Linux1 partition

vii. Also from the LPAR configuration manager, display the Guest “console” log as shown in Figure 2-76.

```
                    Display Guest Environment Console Log
                                                                 System:   SEATTLE
Partition(s) to display . . . . . . . . . . .  *ALL  *ALL,   1-3       04/05/01
Number of lines to display  . . . . . . . . .  999   1-999             09:27:12

Par
ID   Log
  1  Mapping load area - physical addr = 0, absolute addr = 18000000
     Load area size 32768K
     HPT absolute addr = 24000000, size = 4096K
     D-cache line size = 128  (log = 7)
     I-cache line size = 128  (log = 7)
     mf.c: iSeries Linux LPAR Machine Facilities initialized
     Registering c00d5abc in reg c029c810
     Registering c00d62b0 in reg c029c818
     Total memory: 264241152 bytes
     Progress: Ý0300¨ - MMU:hash init
     Progress: Ý0105¨ - hash:enter
     iSeries_hashinit: added 24576 hptes to existing mapping
     Progress: Ý0205¨ - hash:done
     Progress: Ý0301¨ - MMU:mapin
                                                                      More...
F3=Exit   F5=Refresh   F8=Clear   F10=Monitor log   F24=More keys
Already at top of area.
```

Figure 2-76 Displaying the Linux console log

viii. Was the hosted partition varied on from the virtual service panel in SST? If so shut it down from the same panel and vary on the NWSD.

ix. Do you have a corrupted or missing kernel?

- Virtual LAN

  Most of the issues with virtual LAN will be with the configuration since it is not actual Ethernet media, but a vehicle for relaying Ethernet messages.

  a. Is the interface up in Linux? Run the **ifconfig** command from the virtual console. If it is not, check /var/logs/messages, run the **dmesg** command, or correct the configuration.

  b. Can you ping from Linux to OS/400? This generally is a matter of having both sides on the same virtual port (see the discussion in 2.5.1, “Virtual LAN configuration” on page 52) and having an IP address configured for each host (Linux and OS/400).

  c. From the hosting partition, can you ping OS/400 and Linux? If you can only ping the OS/400 system or the interface for the virtual LAN, there is a subnetting problem. Perhaps the network address is used for the virtual LAN interface.

  d. Can you ping from Linux through the LAN adapter on OS/400? If you can ping across the virtual LAN but not to the OS/400 adapter, the route on OS/400 is not defined correctly. The **traceroute** utility may help resolve this. There are both OS/400 and Linux versions that may be helpful.

  e. Can you ping from the external LAN to Linux? If you can ping to the virtual LAN interface but not to the Linux interface, it is probably a routing problem. Again the **traceroute** utility may help.

  f. Is the OS/400 IP forwarding turned on?

  g. Is the Ethernet LIND status varied on or off? It may not be autostarted.

h. Can you see the virtual adapter in Hardware Service Manager?

i. Are the interfaces for OS/400 and virtual LAN active? Use CFGTCP, select option **1**, and select F11 (Display line information) as shown in Figure 2-77.

```
                        Work with TCP/IP Interfaces
                                                              System:   SEATTLE
Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display   9=Start   10=End

     Internet         Subnet           Interface
Opt  Address          Mask             Status

     x.x.x.x          255.255.255.0    Active
     x.x.x.x          255.255.255.248  Active
     127.0.0.1        255.0.0.0        Active

                                                                        Bottom
F3=Exit      F5=Refresh   F6=Print list   F11=Display line information
F12=Cancel   F17=Top      F18=Bottom
```

Figure 2-77 Verifying the interface status

j. Has Linux been configured and was the appropriate script file altered to start the interface correctly at boot up?

- Virtual console

  What Telnet client are you using? OS/400 is not recommended. Use PuTTY if using a Microsoft Windows PC.

- Native I/O problems

  What is the status of the IOAs? Use the display partition information or concurrent maintenance.

  - Problems removing/adding adapters?
  - Is the bus owned and dedicated to OS/400?
  - Are resources in use?
  - Are you trying to remove an IOP that is supporting other devices?
  - Are you trying to remove DASD that are configured?

3

Linux administration

This chapter discussed basic Linux administration. The main topics that are covered include:

- How devices appear in the Linux system and how to handle them
- Locating files in the Linux directory structure
- Working with disk partitions
- Working with network adapters
- Managing users and groups
- Keeping track of what's running on a system
- Where the log files are stored for when something goes wrong
- File permissions and how to handle them
- National language support
- Backing up and restoring the system or parts of the system

3.1 General Linux concepts

To Linux, everything is a file. Configuration parameters are stored in text files in the /etc directory. Devices are seen as files located in /dev. And, information about the system can be found in files in the /proc directory.

Important: The file system and almost everything else in Linux is case sensitive. This is a very big difference as opposed to OS/400 where case does not generally matter. Make sure that you keep an eye on this, and use exactly the same case as in our examples.

3.1.1 Configuration files

It is no coincidence that configuration files are the first topic to be discussed in this section of this book. Practically all configuration of programs in Linux can be done by editing text files located in /etc or in the subdirectories below. To display a textfile on the screen, use **`cat filename`**, for example:

```
cat /etc/passwd
```

Use the **`less filename`** command to display those files screen by screen. Type **`q`** to exit **`less`**.

3.1.2 Editing files using text editors

More and more system administration tasks can be done using configuration tools (as explained later in this section). But when working with Linux, you cannot avoid the fact of editing text files. To do scripting, edit configuration files, or change preferences, you must inevitably edit text files. This can be hard in the beginning, but after a while, you appreciate it because it enables you to change the behavior of almost everything just as you want it.

Important: Try to break the barrier of being able to edit files as soon as you can. It will help you very much when working and administering your Linux system. Try to spend just one hour using a text editor, and then continue.

Quite a few text editors are available in Linux. More experienced UNIX users should probably use editors such as **`vi`** or **`emacs`**. Users who have never used one of these editors should probably use **`pico`**.

`pico`, **`vi`**, and **`emacs`** are editors in text mode, so you don't need X Windows to run them, which is quite important, because from time to time a text editor is needed to get the X Windows system running.

Here is a short introduction on several editors. Again, take a few moments to learn at least one editor.

pico

Probably the most basic editor that is generally available is **`pico`**. It might be a good idea to start working with **`pico`**, until you become somewhat familiar with the concept of editing files in text mode. For more information on its use, refer to the following Web sites:

- http://www.indiana.edu/~ucspubs/b103/
- http://www.uwsg.iu.edu/usail/external/riceinfo/UNIX12/unix12.html

vi

A commonly used and advanced editor is **vi**. The **vi** variant that you will find on nearly all distributions is **vim** (vi improved), and when you encounter **vi** it's usually **vim**.

Online help in **vi** is accessible by typing **:help** or by starting the hands-on tutorial using the **vimtutor** command executed on the Linux command line.

A good Vim Howto is available on the Web at:
http://www.linuxdoc.org/HOWTO/Vim-HOWTO.html

You can also find some quick reference documents on all the **vi** controls at:

- http://www.linuxdoc.org/HOWTO/Vim-HOWTO-11.html
- http://www.cs.umr.edu/unixinfo/general/packages/viguide.html

Print out one of these references so you can use them while working with **vi**.

To turn on syntax highlighting in **vi** (to help you edit scripts, for example), run this from your command line:

```
echo syntax on >> ~/.vimrc
```

If you happen to use a graphical environment, you can as well use the graphical version of **vim**, which is called **gvim**. And for beginners, there is an "easy" mode that can be started with the command **evim**, which does not discriminate between command and insert mode (you are always in insert mode).

emacs

Another very advanced editor is **emacs**. It is much bigger than **vi** to load, so it takes longer but it has more things built in. In a way, it's an Integrated Development Environment (IDE) on its own with commands to compile programs or run version control programs like CVS.

For help with **emacs**, refer to:
http://www.lib.uchicago.edu/keith/tcl-course/emacs-tutorial.html

For a reference list of controls, go to: http://www.indiana.edu/~ucspubs/b131/

3.1.3 Configuration tools

Several configuration tools are available to simplify administration on a Linux system by integrating all administration tasks into one graphical user interface. X Windows and text-based tools can be found. Depending on the distribution you've installed, LinuxConf, YaST, or something similar is available. In this chapter, however, the standard Linux commands are used to do the administration. Feel free to try one of these tools.

For more information about LinuxConf, check out: http://www.solucorp.qc.ca/linuxconf/

3.1.4 The man command

When you need help, the easiest way to access it is by running the **man** command to show the *manpages*. For help on **useradd**, type:

```
man useradd
```

In the same way, retrieve information about the **man** command by running:

```
man man
```

Sometimes, you need to explicitly specify a section number to solve ambiguities. For example, the term *passwd* has two meanings: one as the name of a command and the other as the file with name passwd. Here is a list of section numbers and their description:

1. User commands
2. System calls
3. Library calls
4. Devices
5. File formats
6. Games
7. Miscellaneous information
8. System administration

For help about the **passwd** command, run:

```
man 1 passwd
```

For help on the /etc/passwd file, run:

```
man 5 passwd
```

For more information on the **man** command and to get help and documentation, refer to Chapter 16, “Help information” on page 273.

3.2 Devices

This section focuses on the iSeries Linux-specific devices, mainly virtual devices, and a short explanation of how they are used and configured.

3.2.1 Linux devices

Every device that can be used on a Linux system is represented by a special file located in the /dev directory, or in some cases in the /proc file system. On a PC, for example, you can find the /dev/hda file, which refers to the first hard disk drive found on its IDE controller. The /dev/hda1 file refers to the first partition on that hard disk drive.

The special files in /dev can also represent a software mechanism, such as /dev/null, which represents the garbage bin. If you don’t want to see the output that a command produces, you simply send it to /dev/null such as in:

```
ls > /dev/null
```

Of course the example above is not useful as such, but it can be very handy in scripts.

3.2.2 Creating devices

The contents of this directory are built one of two ways. They can be created by the **mknod** command, mostly used by scripts or system utilities. A user almost never has to use the **mknod** command, since it is done at installation time.

The other way is that devices in the kernel create entries in /dev themselves, but they can only do that when the kernel is compiled with *devfs* support. See 4.3, “Building the kernel” on page 154, for more information.

3.2.3 iSeries-specific devices

Comparing the /dev directory on an iSeries Linux system and an ordinary Linux system, you can find some differences, especially the virtual devices use specific drivers IBM added to the Linux kernel. These iSeries-specific additions to the kernel will be made public in the near future as a contribution to the open source community.

It should be noted that IBM does not control when kernel modifications will be accepted by the Linux kernel maintainers, nor when the third parties will have a distribution available. It is possible that for some period of time the iSeries specific support may be made available as patches as opposed to being fully integrated into the kernel source tree.

When summarizing these types of devices, the assumption is made that the kernel is compiled with devfs support. The main difference (from a user point of view) is that the /dev file systems without the devfs support is only one level deep, without any directory structures. The devfs kernel module creates a hierarchy in which all the devices have their own location, so that you get a better view of these devices.

In Table 3-1, there is a list of the most current devices, each with their location in the devfs file system. For more details about how to use them, refer to the sections on each device in Table 3-1.

Table 3-1 List of devices and their location in the devfs file system

Device description	Location in the devfs file system
Disks (direct attached)	► /dev/hd/disc0/disk ► /dev/hd/disc1/disk Partitions are called /dev/hd/disc0/part1 and /dev/hd/disc0/part2, etc.
Disks (virtual)	► /dev/viod/disc0 = /dev/discs/disc0 (link to /dev/viod/disc0) ► /dev/viod/disc1 = /dev/discs/disc1 (link to /dev/viod/disc1)
CD drives (direct attached and virtual)	► /dev/cdroms/cdrom0 ► /dev/cdroms/cdrom1
Tape drives (direct attached)	► /dev/tapes/tape0 ► /dev/tapes/tape1
Tape drives (virtual)	► /dev/vt0 ► /dev/vt1
LAN interfaces (direct attached)	Check out /proc/net/dev for list. These devices are not located in /dev. ► eth0 *(Ethernet)* ► eth1 ► tr0 *(token ring)* ► tr1
LAN interfaces (virtual)	Check out /proc/net/dev for list. These devices are not located in /dev. ► eth0 *(virtual Ethernet)* ► eth1

As noted, not all Linux systems that exist use the devfs file system. Table 3-2 contains a list of the names they will have in a non-devfs system.

Note: These file names are just conventions and tend to be different in each distribution. For an official list, look at the kernel source tree in Documentation/devices.txt.

Table 3-2 List of devices and their location in the non-devfs file system

Device description	Location in the non-devfs file system
Disks (direct attached through SCSIdev)	► /dev/sda ► /dev/sdb Partitions are called /dev/sda1 and /dev/sdb3, etc.
Disks (virtual)	► /dev/hda ► /dev/hdb ► /dev/hda1 *for disk partitions* when using IDEemulation
CD drives (virtual)	► /dev/viocd0 ► /dev/viocd1
CD drives (direct attached)	► /dev/scda ► /dev/scdb
Tape drives (direct attached)	► /dev/tap0 (normally available in all distributions) ► /dev/mt0 ► /dev/mtl0 In some distributions, depending on the name (mt, mtl, mtr), the tape is opened in a different mode (rewind, append, etc).
Tape drives (virtual)	► /dev/viotape0 ► /dev/viotape1
LAN interfaces (direct attached)	Check out /proc/net/dev for list. These devices are not located in /dev. ► eth0 *(Ethernet)* ► eth1 ► tr0 *(token ring)* ► tr1
LAN interfaces (virtual)	Check out /proc/net/dev for list.These devices are not located in /dev. ► eth0 *(virtual Ethernet)* ► eth1

Virtual console

This virtual console provides the console function for a Linux system. The console is the input/output terminal used by the system directly (for example, without using TCP/IP) and is how programs like the installation program communicate with the user prior to networking being configured. It is also required to troubleshoot the system. For example, if a disk is found to be damaged during the boot process, the system administrator is often asked to repair the disk using the console before the boot is allowed to proceed. In that case, Linux does not boot until an answer is received, and this answer can only be provided using the virtual console.

In the devfs file system, the console is device */dev/viocons/0*. It is a character device driver with major number 100. The most common way to connect to the console is by opening a Telnet session to port 2301 on the primary partition or the hosting OS/400 partition. Refer to 2.4.6, “Connecting to the virtual Linux console” on page 44, for more details on how to connect to this virtual console.

Virtual disk

Virtual disk is also called *virtual direct access storage device* (*virtual DASD*).

Example 3-1 shows how the disks appear in /dev. You have two network server storage spaces attached to the Linux system. They each have a corresponding entry in the /dev/viod directory.

Example 3-1 Device names of virtual disks using devfs

```
/dev
   /viod
      /disc0
         /disc
         /part1
         /part2
      /disc1
         /part1
   /discs
      /disc0          This is a symbolic link to the /dev/viod/disc0 directory.
      /disc1          This is a symbolic link to the /dev/viod/disc1 directory.
```

Example 3-2 shows you the device names of virtual disks and their partitions, without using devfs. Since IDE emulation can be used, you get the same situation as in Linux on a normal PC, where there are no subdirectories, just filenames in a one-level hierarchy. IDE emulation is definitely recommended for compatibility and transparency reasons.

Example 3-2 Device names of virtual disks without using devfs, but using IDE emulation

```
/dev
   /hda               The first virtual disk
   /hda1              The first partition of first virtual disk
   /hda2              The second partition of first virtual disk
   /hdb              The second virtual disk
```

Virtual Ethernet

The virtual Ethernet adapters provide access to virtual LAN configured in the LPAR configuration. They actually *do not* show up in the /dev directory. To see what adapters are available, look in the /proc/net/dev file (use **`cat /proc/net/dev`**).

Virtual CD

The virtual CD provides access to the hosting OS/400 partition CD drive. There is a parameter on the NWSD that can restrict Linux from accessing CD drives, but by default it can see all CD drives on the hosting partition.

With devfs support, these devices show up under the /dev/cdroms directory, for example `/dev/cdroms/cdrom0`. To use a CD-ROM, it must be mounted as well. For example, to mount it under the /mnt/cdrom directory, use either of the following commands:

```
mount /dev/cdroms/cdrom0 /mnt/cdrom
mount /cdrom
```

Use the second command above only if /etc/fstab has an entry for the CD-ROM.

Virtual tape

The virtual tape provides access to the hosting OS/400 partition tape drive, and the tape has to be varied off in the host partition. A parameter on the NWSD can restrict Linux from accessing tape drives, but by default, Linux can see all tape drives on the hosting partition. Refer to 3.10, “Backup and recovery” on page 121, for more information on how to use this.

With devfs support, these tape devices show up under the /dev/viotape directory, for example, `/dev/viotape/tape0`.

Direct attached I/O

The direct attached I/O devices show up exactly the same as on an ordinary Linux system and, therefore, it is more intuitive to set up, especially for direct attached LAN. For example, disks would show up as /dev/discs/disc0, /dev/discs/disc1, etc. Native LAN cards show up as devices named `eth0` or `tr0`, depending on whether they are Ethernet or token ring adapters.

Tools like **`fdisk`** and **`ifconfig`** are actually regular Linux commands, and therefore, can be used on all devices whether they are direct attached or virtual. See 3.3.3, “Partitioning the added disk” on page 97, and 3.4, “Working with network devices (virtual and direct)” on page 101, for examples about these two commands.

3.2.4 Device name changes

Linux has, so far, undergone fairly frequent changes by iSeries standards. In particular, the nomenclature for devices may change. In later kernels that will be released after this book is published, for example, it appears that virtual Ethernets will be named the same as physical Ethernets. That is, they will all have *ethxx* type names, and *vethxx* will disappear.

What are the implications? Well, when you upgrade the kernel, your application may be unaffected, but your system administration (and, your boot process) may fail. You may need to use the 2301 console to reconfigure your *ethxx* devices, because they may now describe different entities than they formerly did.

You may want to create an additional Linux partition to “role play” this. You do not need to install all your applications – just a minimal copy of your distribution (perhaps using less than two gigabytes of total disk) and configure it for the same physical resources. If you are hosted and have the space, you may be able to simply duplicate the network storage object.

With the new partition created, simply vary off the production version, give its resources to the test version and vary on (IPL) the test version in its place using the new kernel from a stream file. Monitor the boot with the 2301 console and figure out which devices are going to change and to what. This may require manual editing of some files, such as /etc/fstab (for disk resources) and any files that define the virtual and physical LANs (often located under /etc/sysconfig/network-scripts).

Another method, which may be simpler, is available to those with shared processors and only hosted virtual disks, with enough space to manage it:

1. Create the new partition (NWSD, NWSSTG).
2. Give it half the CPU power of the old Linux logical partition, leaving the old one at half strength as well.

3. Install and boot both partitions, leaving the physical LAN with the old partition. Assign a temporary IP address to the new partition and make sure it shares a common virtual LAN with the old partition.
4. Export all the important data (/home, /var/lib/mysql, /etc) via NFS from the old partition.
5. Judiciously copy that information into the new configuration.
6. Vary both off.
7. Move the physical LAN devices from the old partition to the new partition.
8. Move any physical disks from the old partition to the new partition. Then you need to determine (by some trial mountings) how to edit /etc/fstabs and include them in normal operation.
9. Reboot the new partition and sort out where the LAN devices end up (typically, the virtual Ethernets would be unchanged with this method).
10. Move the CPU resources to the new partition. Leave the old partition's virtual disk around for a while in case you missed something on the copy (you can even unlink it from the old partition and link it into the new partition somewhere).

Some kernel upgrades do not affect the existing device naming. It is important to know which is which. But, if you accidentally install a kernel with name changes, the usual result is a failed boot or a loss of network connectivity. Make sure the old kernel is still available. New kernels should typically be tried from a stream file for this reason.

3.3 Handling virtual and direct attached hard disks

In 3.2.1, “Linux devices” on page 90, we discussed how devices, such as virtual DASD, show up in the Linux system. This section covers how to handle these virtual disks. All concepts explained here also work for the direct I/O disks. Simply replace the filenames of the virtual disks in /dev with names of native DASD. For details on naming in /dev, see “Virtual disk” on page 93 and “Direct attached I/O” on page 94.

The following scenario explains how to add an extra virtual disk to your Linux partition. It discusses creation, disk partitioning, formatting, and mounting.

3.3.1 Looking at partition information of hard disks

When talking about partitions here, we refer to “disk partitioning” as it is known on PCs, *not* logical partitioning.

To partition these virtual disks, use a Linux command such as **fdisk** or **cfdisk**. The latter certainly is the most user-friendly one, but **fdisk** is more powerful. Now look at the virtual disk /dev/viod/disc0/disc using **fdisk**, as shown in Example 3-3. Note that this disk has two partitions named */dev/viod/disc0/part1* and */dev/viod/disc0/part2*, as you can see in Example 3-1. (For non-devfs systems, this would be **fdisk /dev/hda**.)

Example 3-3 Virtual disk partitions

```
[root@ihavenoname disc1]# fdisk /dev/viod/disc0/disc

Command (m for help): m                 Type m first to display a list of options.
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
```

```
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)

Command (m for help): p              Type p to print the list of partitions.

Disk /dev/viod/disc0/disc: 255 heads, 63 sectors, 196 cylinders
Units = cylinders of 16065 * 512 bytes

               Device Boot    Start       End    Blocks   Id  System
/dev/viod/disc0/disc1             1       193   1550241   83  Linux
/dev/viod/disc0/disc2   *       194       196     24097+  41  PPC PReP Boot
```

The first partition has a partition ID of 83, which denotes it is a Linux ext2 partition. The second partition is of type PPC PreP Boot, which is an IPL source for the kernel. If your Linux system uses this partition to boot (and it probably does if it shows up), definitely don't delete this partition. The list of all partition IDs can be retrieved using the **l** option in **fdisk** as illustrated in Example 3-4. Exit fdisk using option **q** (quit without saving changes).

Example 3-4 List of all known partition types

```
Command (m for help): l

 0  Empty           17  Hidden HPFS/NTF 5c  Priam Edisk     a6  OpenBSD
 1  FAT12           18  AST Windows swa 61  SpeedStor       a7  NeXTSTEP
 2  XENIX root      1b  Hidden Win95 FA 63  GNU HURD or Sys b7  BSDI fs
 3  XENIX usr       1c  Hidden Win95 FA 64  Novell Netware  b8  BSDI swap
 4  FAT16 <32M      1e  Hidden Win95 FA 65  Novell Netware  c1  DRDOS/sec (FAT-
 5  Extended        24  NEC DOS         70  DiskSecure Mult c4  DRDOS/sec (FAT-
 6  FAT16           3c  PartitionMagic  75  PC/IX           c6  DRDOS/sec (FAT-
 7  HPFS/NTFS       40  Venix 80286     80  Old Minix       c7  Syrinx
 8  AIX             41  PPC PReP Boot   81  Minix / old Lin db  CP/M / CTOS / .
 9  AIX bootable    42  SFS             82  Linux swap      e1  DOS access
 a  OS/2 Boot Manag 4d  QNX4.x          83  Linux           e3  DOS R/O
 b  Win95 FAT32     4e  QNX4.x 2nd part 84  OS/2 hidden C:  e4  SpeedStor
 c  Win95 FAT32 (LB 4f  QNX4.x 3rd part 85  Linux extended  eb  BeOS fs
 e  Win95 FAT16 (LB 50  OnTrack DM      86  NTFS volume set f1  SpeedStor
 f  Win95 Ext'd (LB 51  OnTrack DM6 Aux 87  NTFS volume set f4  SpeedStor
10  OPUS            52  CP/M            93  Amoeba          f2  DOS secondary
11  Hidden FAT12    53  OnTrack DM6 Aux 94  Amoeba BBT      fd  Linux raid auto
12  Compaq diagnost 54  OnTrackDM6      a0  IBM Thinkpad hi fe  LANstep
14  Hidden FAT16 <3 55  EZ-Drive        a5  BSD/386         ff  BBT
16  Hidden FAT16    56  Golden Bow
```

3.3.2 Creating and adding a virtual disk

Create a new partition using the CRTNWSSTG command on the hosting OS/400 partition, as illustrated in Figure 3-1.

```
                    Create NWS Storage Space (CRTNWSSTG)

Type choices, press Enter.

Network server storage space . . > LINUX2ALT     Name
Size . . . . . . . . . . . . . . > 1800          *CALC, 1-64000 megabytes
From storage space . . . . . . .   *NONE         Name, *NONE
Format . . . . . . . . . . . . . > *OPEN         *NTFS, *FAT, *FAT32, *OPEN
Auxiliary storage pool ID  . . .   1             1-99
Text 'description' . . . . . . .   *BLANK


                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 3-1 Creating a network server storage space

Then you attach it to the NWSD, as illustrated in Figure 3-2. Note that the partition must be varied off to do this.

```
                    Add Server Storage Link (ADDNWSSTGL)

Type choices, press Enter.

Network server storage space . .   LINUX2ALT     Name
Network server description . . . > REDHAT71      Name
Dynamic storage link . . . . . .   *NO           *NO, *YES
Drive sequence number  . . . . .   *CALC         3-18, *CALC


                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display        F24=More keys
```

Figure 3-2 Adding a network server storage space to a Linux partition

3.3.3 Partitioning the added disk

Attention: Use care when partitioning disks. If you change partitioning information for the disk that you need to boot your Linux partition, the data may be damaged if you don't know exactly what you're doing.

To stay distribution independent, we use **fdisk** and **cfdisk** (where **cfdisk** is perhaps easier for unexperienced users). However, refer to your distributors specific documentation concerning this topic.

After you add a link to this NWSSTG, a new disk appears as /dev/viod/disc1. You have to repartition the newly added virtual disk, as shown in Example 3-5.

Example 3-5 Partitioning a virtual disk

```
[root@ihavenoname disc1]# fdisk /dev/viod/disc1/disc

Command (m for help): p

Disk /dev/viod/disc1/disc: 255 heads, 63 sectors, 229 cylinders
Units = cylinders of 16065 * 512 bytes

               Device Boot    Start       End    Blocks   Id  System
/dev/viod/disc1/disc1             1       228   1831378+   6  FAT16

Command (m for help): d                  Delete the existing partition
Partition number (1-4): 1

Command (m for help): p

Disk /dev/viod/disc1/disc: 255 heads, 63 sectors, 229 cylinders
Units = cylinders of 16065 * 512 bytes

               Device Boot    Start       End    Blocks   Id  System

Command (m for help): n                  Create a new partition of 1 GB
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-229, default 1): (press Enter)
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-229, default 229): +1000M

Command (m for help): p

Disk /dev/viod/disc1/disc: 255 heads, 63 sectors, 229 cylinders
Units = cylinders of 16065 * 512 bytes

               Device Boot    Start       End    Blocks   Id  System
/dev/viod/disc1/disc1             1       128   1028128+  83  Linux

                                         Because the partition already has ID 83, you
                                         don't need to change that anymore. Otherwise you
                                         can do that with the option t.

Command (m for help): w                  Write all changes to the partition table.
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: If you have created or modified any DOS 6.x
partitions, please see the fdisk manual page for additional
information.
Syncing disks.
[root@ihavenoname disc1]#
```

3.3.4 Creating the file system (formatting the disk partition)

You now format the partition, using **mkfs.ext2**, as illustrated in Example 3-6.

Example 3-6 Formatting the new partition

```
[root@ihavenoname /root]# mkfs.ext2 /dev/discs/disc1/part1
mke2fs 1.19, 13-Jul-2000 for EXT2 FS 0.5b, 95/08/09
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
128768 inodes, 257032 blocks
12851 blocks (5.00%) reserved for the super user
First data block=0
8 block groups
32768 blocks per group, 32768 fragments per group
16096 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376

Writing inode tables: done
Writing superblocks and filesystem accounting information: done
[root@ihavenoname /root]#
```

3.3.5 Mounting a disk partition

In Linux, the whole file system is represented in one tree. To write or read data on a disk partition, you have to mount it, so an empty directory (for example, /mnt/second) in the file system is linked to the mounted partition. The directory structure residing on the newly mounted partition is therefore completely displayed under that directory.

To mount a disk partition, use the **mount** command, as shown in Example 3-7. In this case, first create a directory named /mnt/second in which you mount the new formatted partition.

Example 3-7 Mounting a partition

```
[root@ihavenoname /root]# cd /mnt
[root@ihavenoname /mnt]# ls
[root@ihavenoname /mnt]# mkdir second
[root@ihavenoname /mnt]# mount /dev/viod/disc1/part1 /mnt/second
[root@ihavenoname /mnt]# cd second
[root@ihavenoname second]# ls
lost+found
```

3.3.6 Unmounting a disk partition

Currently, no files are created in this partition. Only the lost+found directory is there, which is the way a Linux system keeps corrupted data after, for example, a power down without first unmounting the disks. This unmounting procedure is done automatically when properly shutting down the Linux system or can be done manually by running **umount** with the mounted directory as an argument (or the device name in /dev, as you prefer), for example:

```
umount /mnt/second
```

This command unmounts the partition mounted under the /mnt/second directory. Unmounting can only be done when no files in this partition are being used.

3.3.7 Automatically mounting a disk partition at boot time

You describe information about file systems used in your system in the file /etc/fstab. Each file system is described on a separate line. One record has six fields that are separated by tabs or spaces:

```
device mountpoint fstype options frequency passno
```

See manpage of fstab (**man fstab**) and mount (**man mount**) for details of each field. For information on editing files, see 3.1.2, “Editing files using text editors” on page 88.

Example 3-8 shows the /etc/fstab file.

Example 3-8 The /etc/fstab file

```
/dev/viod/disc0/part1  /            ext2     defaults          1  1
none                   /proc        proc     defaults          0  0
none                   /dev/pts     devpts   gid=5,mode=620    0  0
/dev/cdroms/cdrom0     /cdrom       iso9660  ro,user,noauto    0  0
bauer5:/home           /home        nfs      exec,dev,suid,rw  0  0
LABEL=/P782XRLC        /mnt/first   ext2     defaults          1  2
/dev/viod/disc1/part2  swap         swap     defaults          0  0
/dev/viod/disc1/part1  /mnt/second  ext2     defaults          1  2
```

The first line specifies the root (/) file system using the device /dev/viod/disc0/part1, and it needs to be dumped. The command **fsck** is executed at reboot time. See 3.5.2, “Maintenance of the file system” on page 103, to learn about **fsck**.

The second and the third line show the proc file system and devpts file system respectively.

The forth line describes the CD-ROM device to be mounted. The three options mean:

- This file system is read-only (`ro`).
- Normal users can mount this file system (`user`).
- This file system should not be mounted at boot time or by the **mount -a** command (`noauto`).

The fifth line specifies the /home directory is a mount point of the directory /home on the host bauer5 via the Network File System (NFS).

The four options are:

- Users can execute binary programs on this file system (`exec`).
- The system interprets character or block special devices on this file system (`dev`).
- This file system allows the user to set user-identifier bit or group-identifier bit (`suid`).
- Normal users can read and write files on this file system (`rw`).

The sixth line illustrates how to use a label in the first field. This file system will be mounted on the /mnt/first directory. This file system needs to be dumped, without running **fsck** at reboot time.

The seventh line specifies swap partition.

The last line specifies local drive partition to be mounted on the directory /mnt/second. This file system also needs to be dumped, without running **fsck** at reboot time.

There are many options for each supported type of file system. Use **man mount** for more details.

3.3.8 Moving a virtual disk between partitions

When storing data on virtual disks, you can easily move them from one Linux logical partition to another logical partition. Simply disconnect a virtual disk from one partition, connect it to the other, and mount it there.

It should also be possible to mount FAT, FAT32, and NTFS disk partitions originating from an Integrated xSeries for iSeries Server into a Linux partition.

3.4 Working with network devices (virtual and direct)

As discussed in “Virtual Ethernet” on page 93, network adapters appear in the /proc/net/dev file (use **cat /proc/net/dev**). The same information is formatted more readable by running **ifconfig**, such as shown in Example 3-9, but that normally only shows the active interfaces. If you run the **ifconfig -a** option, you see information about all devices that are active at the moment.

When native LAN cards are available, they show up as *eth0* or *tr0* depending on whether they are Ethernet or token ring adapters.

Attention: When you modify network settings, make sure you do this from the console, because you could lock out your own session if it uses the network adapter.

Example 3-9 LAN devices on a Linux system

```
[root@bauer4 /root]# ifconfig -a
lo        Link encap:Local Loopback
          LOOPBACK  MTU:16176  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0

eth0     Link encap:Ethernet  HWaddr 02:01:FF:00:FF:02
          inet addr:192.168.140.2 Bcast:192.168.140.255  Mask:255.255.255.248
          UP BROADCAST RUNNING MULTICAST  MTU:9000  Metric:1
          RX packets:77293 errors:0 dropped:0 overruns:0 frame:0
          TX packets:88539 errors:0 dropped:1 overruns:0 carrier:0
          collisions:0 txqueuelen:100
```

Using the **ifconfig** command, you can also modify settings for the interfaces. To modify an IP address, set the network mask and bring the interface up. Use the following command:

```
ifconfig eth0 192.168.140.2 netmask 255.255.255.248 up
```

To bring an interface up and down use the commands **ifconfig <device> up/down**, as shown in Example 3-10.

Example 3-10 Bringing LAN interfaces down and up again

```
[root@bauer4 /root]# ifconfig
eth0     Link encap:Ethernet  HWaddr 02:01:FF:00:FF:02
          inet addr:10.5.177.99  Bcast:10.255.255.255  Mask:255.255.255.248
          UP BROADCAST RUNNING MULTICAST  MTU:9000  Metric:1
          RX packets:77430 errors:0 dropped:0 overruns:0 frame:0
          TX packets:88637 errors:0 dropped:1 overruns:0 carrier:0
          collisions:0 txqueuelen:100

[root@bauer4 /root]# ifconfig eth0 down
[root@bauer4 /root]# ifconfig

[root@bauer4 /root]# ifconfig eth0 up

[root@bauer4 /root]# ifconfig
eth0     Link encap:Ethernet  HWaddr 02:01:FF:00:FF:02
          inet addr:10.5.1.7  Bcast:10.5.1.7  Mask:255.255.255.248
          UP BROADCAST RUNNING MULTICAST  MTU:9000  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100

[root@bauer4 /root]#
```

Try using **man ifconfig** for more information about this command. Routing information is set up using the **route** command. Simply run **route** to find out what current settings are. In case the Linux partition has problems reaching a DNS server, try **route -n**. Otherwise, the route command will take a very long time before giving routing information because it first tries to resolve IP addresses to hostnames.

Tip: The **ifconfig** and **route** commands are available in each Linux system. However, we advise that you look for distributor-specific documentation that explains how to set this up using graphical or semi-graphical user interfaces.

For more information how to set up routing via OS/400 to access the Linux partition over virtual LAN, see 2.5.1, “Virtual LAN configuration” on page 52.

3.5 Directory hierarchy

This section discusses the directory hierarchy of the Linux system.

3.5.1 Standard of directory structure

Many distributors have created their own distribution. These distributions have many common features and structures, but there are some differences indeed. Especially differences with regard to the file system can cause big problems when application vendors want to port their products among distributions. Furthermore, when users decide to use a new distribution, sometimes they have questions like:

- Where are Web server configuration files located?
- Where is the mail spool directory?
- What is this file used for?

The File system Hierarchy Standard (FHS) solves these problems. See http://www.pathname.com/fhs/ to get to know the standard hierarchy.

Tip: Use the **find** or **locate** command to find files, as explained in 16.5, “Useful tools” on page 277.

3.5.2 Maintenance of the file system

The **fsck** utility checks a file system’s correctness and validity. The system runs **fsck** automatically at boot time, so that any errors are detected (and hopefully corrected) before the system is used. The automatic checking only works for the file systems that are mounted automatically at boot time (see /etc/fstab). Use **fsck** manually to check other file systems. The **fsck** utility must only be run on unmounted file systems, never on mounted file systems (with the exception of the read-only root during startup). This is because it accesses the raw disk and can therefore modify the file system without the operating system realizing it.

Example 3-11 shows running **fsck**.

Example 3-11 Running fsck

```
# fsck -t ext2 /dev/viod/disc1/part2
Parallelizing fsck version 1.19 (13-Jul-2000)
e2fsck 1.19, 13-Jul-2000 for EXT2 FS 0.5b, 95/08/09
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Check reference counts
Pass 5: Checking group summary information

/dev/viod/disc1/part2: clean, 23833/128768 files, 148393/257032 blocks
```

Use **man fsck** to see what options can be used other than -t.

3.6 Users and groups

This section covers management of users and groups.

3.6.1 Multi-user system

With Linux, as a multi-user operating system, you can log on using different user accounts simultaneously. On a Linux PC, this is usually done through the use of the concurrent virtual consoles.

Note: The name *virtual console* here is something different than in iSeries Linux. In iSeries Linux, you only have one virtual console that you connect to using Telnet to port 2301. On PCs, you press Ctrl-Alt-F1 for the first virtual console and Ctrl-Alt-F6 for the last console.

On an iSeries Linux system, however, multiple users usually connect to so called *pseudo terminals* using a secure shell client (ssh) or Telnet. They are prompted for username and password for identification, quite similar to a 5250 emulation to OS/400.

A graphical session to the iSeries Linux can be established as well using an X Windows server. This is discussed in Chapter 5, “X Windows and OpenOffice” on page 165.

On Linux, users are identified by a *username*, and that username has a one-to-one mapping with a unique number called a *user ID*. A user is always a member of at least one group, but can be a member of several supplementary groups. These usernames and groups can also be used to give specific access to files, commands, or directories. See 3.8, “Security issues” on page 112, for more information.

3.6.2 Server-oriented system

On a server-oriented system, such as a system that is solely used for serving Web pages and as a firewall, user profiles are used to run specified tasks. It is quite common to run server daemons, such as ftp or http daemons, with special user profiles so you have better tracking on which process corresponds to which server daemon and files. (A *daemon* could be compared to a server job in OS/400 that runs in *batch mode*.)

Quite often, an account is created with a home directory that points to the html files in a Web server. That way a webmaster can maintain all files, but can’t modify crucial settings.

3.6.3 The root user

There is one user with maximal authority named *root*. You could say this superuser corresponds to the QSECOFR user profile on a regular iSeries server, although no connection between them exists. QSECOFR can only be used in the OS/400 system, and *root* is specifically for the Linux system.

When you log on to the Linux console, you are identified as *root*. You use the *root* profile to install programs. However, we strongly discourage you from using *root* as an account to generally use the system as an end user.

3.6.4 Managing users

This section discusses adding, deleting, and modifying user accounts using standard Linux commands. Refer to 3.6.6, “Layout of /etc/passwd” on page 107, for editing users straight into the /etc/passwd file, but this section lets these commands do that job for you.

Attention: While it is possible to make changes in /etc/passwd with an editor, we do not recommend doing this on a multi-user system. While you edit the file, a user could change their default shell. When you end your editor session, you would plainly overwrite their change. You should instead use the various commands that are provided for changing data in /etc/passwd. These commands, which include **useradd**, **userdel**, **usermod**, **chsh**, **chage**, and **chfn**, take care of file locking for /etc/passwd, so it cannot become corrupted.

If you ever have to edit /etc/passwd directly, you should not type vi /etc/passwd, but use the special command **vipw** for that. This sets the appropriate locks to prevent file corruption. Likewise, **vipw -s** can be used to edit /etc/shadow.

Tip: The commands **useradd** and **userdel** can be customized via the shell scripts useradd.local and userdel.local.

For complete information on all commands used here, check the man pages. This is only an overview of the most important options.

Adding users with useradd

One of the methods to create users is by using the **useradd** command. Its syntax is:

```
useradd [-c comment] [-d home_dir] [-m [-k skeleton_dir]] [-e expire_date]
[-f inactive_time] [-g initial_group] [-G group[,...]] [-p passwd] [-s shell] [-u uid [-o]]
[-n] [-r] sadams
```

The most commonly used options are explained in greater detail:

-c Comment (usually the full name of the user).

-d The home directory of the new user. If not specified, this is */home/sadams*.

-g The default group for the user by name or by GID. The group must exist.

-G Additional groups to which the user should belong.

-s Basic shell, for example, program to run at login time.

-u Numeric user ID associated with the user name. If this value is not specified, the system associates the next free user ID>=500. This option comes in handy if you have existing users in OS/400, and want to create users with matching user IDs. This is important for write access to files over a Network File System (see 12.4, “User IDs” on page 245).

-m Create the home directory of the user (usually called /home/<username>), if it does not exist. Apart from creating the directory, copy a skeleton of default configuration files into it from /etc/skel.

-p The encrypted password, as returned by crypt(3), to be set initially. The default is to disable the account.

For example, to add a user account named *sadams*, issue the command:

```
useradd -m -c 'Sam Adams' -u 522 sadams
```

By default, this user's account is disabled. It can be enabled by root by setting a password, using the **passwd** command:

```
[root@bauer4 kernel]# passwd sadams
Changing password for user sadams
New UNIX password: somepassword (typed characters are not displayed)
Retype new UNIX password: somepassword (typed characters are not displayed)
passwd: all authentication tokens updated successfully
```

Normally a system is setup so that all users can access all programs (except, of course, the system commands, reserved for root). They can, however, only call these programs and write files in /tmp and their home directory. In general, many configuration files can be read by normal users (not *root*), but not altered. In most distributions, normal users cannot read critical files like log files.

You might want to change this by customizing authorities. For more information, see 3.8, “Security issues” on page 112, for information on how to do this.

The OS/400 analog to this command is Create User Profile (CRTUSRPRF).

Deleting users using userdel

The **userdel** command deletes a user. Its syntax is:

```
userdel [-r] sadams
```

It only has one option:

-r Files in the user's home directory are to be removed along with the home directory itself. Files located in other file systems have to be searched for and deleted manually.

To delete all files owned by that user (not only in its home directory), you can run a specific command. Files owned by the user can be found (before deleting the user) with:

```
find / -user user2
```

Deletion of all those files can be achieved with the following command (do this carefully when using **find** and **rm**):

```
find / -user user2 -exec rm -f {} \;
```

Use **man find** for more information about this powerful command.

The OS/400 equivalent to this command is Delete User Profile (DLTUSRPRF).

Modifying users with usermod

To change a user account, you can use the **usermod** command. Its syntax is:

```
usermod [-c comment] [-d home_dir [-m]] [-e expire_date] [-f inactive_time]
[-g intial_group] [-G group[,...]] [-l login_name] [-p passwd] [-s shell] [-u uid [-o]]
[-L|-U] sadams
```

The meaning of all options is the same as for the **useradd** command. Only two are special:

-L Lock a user account (disable it)
-U Unlock a user account (enable it)

The OS/400 equivalent of this command is Change User Profile (**CHGUSRPRF**).

3.6.5 Managing groups

This section discusses adding, deleting, and modifying groups using standard Linux commands. See 3.6.7, “Layout of /etc/group” on page 108, for editing groups straight into the `/etc/group` file. This section describes commands that do the job for you.

For complete information on all commands used here, check the man pages.

The primary group of a user cannot be deleted.

Modifying groups with groupmod

To change a group, use the **groupmod** command. The syntax is:

```
groupmod [-g <gid>] [-n <newname>] <groupname>
```

The options stand for:

-g Change the group ID.
-n Change the name of the group.

The following example changes the GID of group1 to 603:

```
groupmod -g 603 group1
```

Be aware that if you change the GID of a group, GID does not change the GID of the users and files. This can result in problems accessing files. For this reason, you should never change this GID.

3.6.6 Layout of /etc/passwd

You *could* manually manage users by altering the configuration file named /etc/passwd. However, as mentioned above, this is not recommended because interleaving read/write accesses (as they can occur in a multi-user system) can lead to corruption of the file. The reason why this can happen is because users might, for example, change their default shell with the **chsh** command, while the admin has /etc/passwd loaded into their editor. When the file is saved later, changes of other people would be overwritten.

Nevertheless, the structure of /etc/passwd deserves our attention to understand the mechanisms behind it. We provide a short readable introduction here; for the full information, you should consult the respective manpage by running:

```
man 5 passwd
```

Here *5* refers to the section about file formats. If *5* is not specified, you are possibly redirected to the man pages of the **passwd** command.

The /etc/passwd file contains a line for every user on the system. On each line, you will find:

```
username:password:user_id:group_id:comment:home:shell
```

Here is a description of the fields:

- **username:** The unique log-in name for the user.
- **password:** Historically, this is the password field, and it deserves some extra comments. Since /etc/passwd must be world readable, the passwords stored here could also be read by any user of the system. The passwords are scrambled with the **crypt(3)** command, but are still susceptible to a brute force attack, especially when weak passwords (like words that occur in a dictionary) are used. Most Linux distributions nowadays use so-called shadow passwords, which means that the password is actually stored in /etc/shadow, which is not readable for normal users. Applications commonly use the Pluggable Authentication Modules (PAM) library to check passwords. The password field in /etc/passwd contains a simple “x” when shadow passwords are used.
- **user_id:** Unique number for this user.
- **group_id:** The number that identifies the primary group to which the user belongs.
- **comment:** Usually used for the real name of the user, but can be any comment.

- **home:** The home directory of the user, mostly /home/username for ordinary users, but can be in /usr or /var for user profiles that are used to run processes.
- **shell:** The program to be run at login; usually a shell such as /bin/bash or /bin/sh. If set to a non-existing executable, the user is unable to login.

To add a user, you would normally use the **useradd** command. If you know what you are doing, however, you can simply add a line here.

If you edit /etc/passwd by hand, you should at least use the special command **vipw** to load /etc/passwd into the **vi** editor. This command ensures proper file locking while the file is being edited to protect it from corruption.

These days, the /etc/passwd file is no longer used to store encrypted passwords, because it is readable by all users on the system. Therefore, the file /etc/shadow is used, where all encrypted passwords are stored, and that file is only readable by *root*. This gives you better protection against users that use brute force methods to search for passwords. When a password of a user is stored in the /etc/shadow file, the password field of that user in /etc/passwd shows an “x”.

3.6.7 Layout of /etc/group

You can manually manage groups by altering the configuration file named /etc/group. To add a group, add a line to the /etc/group file. A line in this file has the following layout:

```
groupname:password:group_id:user_list
```

The fields are described here:

- **groupname:** Unique name for the group
- **password:** Usually left blank
- **group_id:** A unique ID for the group
- **user_list:** A comma-separated list of user names associated with this group. This list is only used to make users part of multiple groups

For extra information, consult the manpages by running:

```
man 5 group
```

The /etc/group file should not be edited directly. Instead, you should use the command **vigr** that opens the file in the **vi** editor and locks the file against concurrent access. This should rarely be necessary, because the **groupadd**, **groupdel**, and **groupmod** commands should normally suffice.

3.6.8 Synchronization of iSeries users and Linux users

Currently there is no synchronization between user accounts on the iSeries Linux system and OS/400 user profiles. However, using NFS, you can have the systems with the same user ID between them. For more information about NFS user IDs, refer to 12.4, “User IDs” on page 245.

3.7 Monitoring the system

All programs that run in a Linux system are processes. A process has a unique *process ID* (PID). This can be compared to what jobs are in an OS/400 system.

3.7.1 The top command

The **top** command lists all the processes that are currently running on a system. By default, it refreshes the screen every five seconds, and processes are sorted with the one using the most processor time on top. Once **top** is started, it takes commands with single keystrokes, such as:

q Quit top
h Show the help screen
k Kill a process by giving its PID
s Set delay in seconds between updates

The display of the **top** command is illustrated in Example 3-12. At the top of the screen, you can see other system information, such as the load, the number of processes, the percentage of idle processor time.

Example 3-12 Most important processes running using top

```
5:27pm  up 1 day,  3:08,  7 users,  load average: 2.38, 2.42, 2.35
155 processes: 152 sleeping, 3 running, 0 zombie, 0 stopped
CPU states: 56.8% user, 37.7% system,  5.4% nice,  0.0% idle
Mem:    62360K av,   61368K used,     992K free,       0K shrd,    1368K buff
Swap:   68504K av,   68452K used,      52K free                   33364K cached

  PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME COMMAND
10337 root      20   0 12492  12M  2220 R    42.8 20.0   0:10 cc1
 5498 reguser   15   0  3716 1860  1648 S    41.5  2.9  18:28 Xvnc
10302 reguser   16  10  1344 1344  1148 R N  10.8  2.1   0:13 wander
10339 itso      19   0  1116 1116   872 R     4.1  1.7   0:00 top
10338 root       9   0  1112 1112   716 S     1.0  1.7   0:00 as
10336 root       9   0  2200 2200   588 S     0.5  3.5   0:00 cpp0
    1 root       9   0   104   52    52 S     0.0  0.0   0:04 init
    2 root       8   0     0    0     0 SW    0.0  0.0   0:00 keventd
    3 root       9   0     0    0     0 SW    0.0  0.0   0:31 kswapd
    4 root       9   0     0    0     0 SW    0.0  0.0   0:00 kreclaimd
    5 root       9   0     0    0     0 SW    0.0  0.0   0:00 bdflush
    6 root       9   0     0    0     0 SW    0.0  0.0   0:00 kupdate
    7 root      -1 -20     0    0     0 SW<   0.0  0.0   0:00 mdrecoveryd
   73 root       9   0     0    0     0 SW    0.0  0.0   0:00 khubd
  816 root       9   0   132   56    56 S     0.0  0.0   0:01 syslogd
  841 root       9   0   620    4     4 S     0.0  0.0   0:00 klogd
  855 rpc        9   0    84    4     4 S     0.0  0.0   0:00 portmap
```

3.7.2 The ps, kill, and killall commands

More advanced management is done using the **ps** and **kill** commands. **ps** lists the processes. Which processes are shown depends on the parameters given to **ps**.

To list all the processes of the system, run **ps aux**, as shown in Example 3-13.

Example 3-13 Using ps aux to list the system processes

```
[root@bauer4 /root]# ps aux
USER       PID %CPU %MEM   VSZ  RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.2  1364  592 ?        S    07:29   0:00 init
root         2  0.0  0.0     0    0 ?        SW   07:29   0:00 [keventd]
```

```
root         3  0.0  0.0     0    0 ?        SW   07:29   0:00 [kswapd]
root         4  0.0  0.0     0    0 ?        SW   07:29   0:00 [kreclaimd]
root         5  0.0  0.0     0    0 ?        SW   07:29   0:00 [bdflush]
root         6  0.0  0.0     0    0 ?        SW   07:29   0:00 [kupdate]
root        11  0.0  0.2  1592  648 ?        S    07:29   0:00 /sbin/devfsd /dev
root       165  0.0  0.2  1492  680 ?        S    07:29   0:00 syslogd -m 0
root       170  0.0  0.4  1888 1108 ?        S    07:29   0:00 klogd -2
daemon     203  0.0  0.2  1460  652 ?        S    07:29   0:00 /usr/sbin/atd
root       213  0.0  0.5  3096 1332 ?        S    07:29   0:00 /usr/sbin/sshd
root       244  0.0  0.3  1856  780 ?        S    07:29   0:00 crond
xfs        298  0.0  1.5  5676 3960 ?        S    07:29   0:00 xfs -droppriv -daemon
root       319  0.0  0.4  2864 1144 ?        S    07:29   0:00 xinetd
root       321  0.0  0.2  1340  508 viocons/0 S    07:29   0:00 /sbin/mingetty --noclear viocons/0
root       350  0.0  0.5  3596 1484 pts/0    S    08:51   0:00 login -- root
root       351  0.0  0.6  2816 1524 pts/0    S    08:51   0:00 -bash
root       505  0.0  0.3  1856  776 ?        S    11:01   0:00 CROND
root       506  0.0  0.3  2140 1004 ?        S    11:01   0:00 /bin/bash /usr/bin/run-parts /etc/cron.hour
root       508  0.0  0.2  1780  680 ?        S    11:01   0:00 awk -v progname=/etc/cron.hourly/sysstat
root       509  0.0  0.3  2124  976 ?        S    11:01   0:00 /bin/sh /usr/lib/sa/sa1 600 6
root       511  0.0  0.2  1348  580 ?        S    11:01   0:00 /usr/lib/sa/sadc 600 6 /var/log/sa/sa29
root       761  0.0  0.5  2800 1472 ?        S    11:23   0:00 -bash
root       783  0.1  1.8 10212 4768 ?        S    11:23   0:00 gnome-terminal
root       784  0.0  0.2  1520  644 ?        S    11:23   0:00 gnome-pty-helper
root       785  0.0  0.5  2796 1468 pts/2    S    11:23   0:00 bash
root       802  0.0  0.3  2724  816 pts/2    R    11:24   0:00 ps aux
```

To terminate a process, use **kill PID**, where PID is the process number of the process to kill. If a process stopped responding, you can do a **kill -9 PID** to force it to stop immediately.

killall processname simply kills all processes with a given process name, for example, to kill the ping process that's running:

```
[root@bauer4 /root]# killall ping
[1]+  Terminated              ping localhost
```

In the following example, you can see two **ping** commands running. The first one is killed quite easily, but the other does not want to stop. Therefore, the **killall -9 command** is used to force it to stop.

```
[root@bauer4 /root]# ps aux
USER       PID %CPU %MEM   VSZ  RSS TTY      STAT START   TIME COMMAND
...
root       826  0.0  0.2  1664  684 pts/2    T    11:37   0:00 ping localhost 7
root       827  0.0  0.2  1664  684 pts/2    S    11:38   0:00 ping localhost
root       829  0.0  0.3  2724  816 pts/2    R    11:38   0:00 ps aux
[root@bauer4 /root]# killall ping
[2]-  Terminated              ping localhost
[root@bauer4 /root]# killall ping
[root@bauer4 /root]# ps aux
USER       PID %CPU %MEM   VSZ  RSS TTY      STAT START   TIME COMMAND
...
root       826  0.0  0.2  1664  684 pts/2    T    11:37   0:00 ping localhost 7
root       848  0.0  0.3  2724  816 pts/2    R    11:38   0:00 ps aux
[root@bauer4 /root]# killall -9 ping
[1]+  Killed                  ping localhost 7
[root@bauer4 /root]# ps aux
USER       PID %CPU %MEM   VSZ  RSS TTY      STAT START   TIME COMMAND
...
root       848  0.0  0.3  2724  816 pts/2    R    11:38   0:00 ps aux
```

3.7.3 System logs

System log files contain a lot of information about what has been going on in a Linux system. There are several of them, and they are all located in the */var/log* directory. It could be compared to the QSYSOPR messages in OS/400, which are displayed with the DSPMSG QSYSOPR command. However, in Linux, all messages are logged to these files (even the good ones). The most important of these log files is */var/log/messages*.

Use the **less** command to display these log files, so you can look at it screen by screen, for example when executing the following command:

```
[root@bauer4 kernel]# less /var/log/messages
```

You can see something like Example 3-14.

For a short list of controls as a quick reference:

Page Down	To go down
Page Up	To go up
q	To quit less
g	To go to the top
G	To go to the bottom
/<string>	To search for <string>
n	To search for the next occurrence
N	To search backwards

To see the last lines of a text file as well as the log files, use the **tail filename** command.

Example 3-14 Part of /var/log/messages

```
Mar 25 14:38:58 bauer4 kernel: Linux NET4.0 for Linux 2.4
Mar 25 14:38:58 bauer4 kernel: Based upon Swansea University Computer Society NET3.039
Mar 25 14:38:58 bauer4 kernel: Starting kswapd v1.8
Mar 25 14:38:58 bauer4 kernel: viocons: init2 - open path to primary
Mar 25 14:38:58 bauer4 kernel: viopath: open, setting sinst 32771, tinst 32771
Mar 25 14:38:58 bauer4 kernel: viocons major is 100
Mar 25 14:38:58 bauer4 kernel: viocons: init2 - open cons to hosting
Mar 25 14:38:58 bauer4 kernel: viopath: got monitor INT event
Mar 25 14:38:58 bauer4 kernel: viocons: init2 - open cons to primary
Mar 25 14:38:58 bauer4 kernel: viocons: init2 - initializing tty
Mar 25 14:38:58 bauer4 kernel: Kernel command line:
Mar 25 14:38:58 bauer4 kernel: pty: 256 Unix98 ptys configured
Mar 25 14:38:58 bauer4 kernel: block: queued sectors max/low 166946kB/55648kB, 512 slots per queue
Mar 25 14:38:58 bauer4 kernel: viod: iSeries Virtual DASD vers 1.01, major 96, max disks 64, hosting
partition 0
Mar 25 14:38:58 bauer4 kernel: viod: opened path to hosting partition 0
Mar 25 14:38:58 bauer4 kernel: viod: Currently 2 disks connected
Mar 25 14:38:58 bauer4 kernel: Partition check:
Mar 25 14:38:58 bauer4 kernel:  vioda: p1 p2
Mar 25 14:38:58 bauer4 kernel: viod: Disk 00 size 1537M, sectors 63, heads 255, cylinders 196, sectsize 512
Mar 25 14:38:58 bauer4 kernel: viod: Disk 00 partition 01 start sector 63, # sector 3100482
Mar 25 14:38:58 bauer4 kernel: viod: Disk 00 partition 02 start sector 3100545, # sector 48195
Mar 25 14:38:58 bauer4 kernel:  viodb: p1 p2
Mar 25 14:38:58 bauer4 kernel: viod: Disk 01 size 1796M, sectors 63, heads 255, cylinders 229, sectsize 512
Mar 25 14:38:58 bauer4 sshd: sshd startup succeeded
Mar 25 14:38:58 bauer4 kernel: viod: Disk 04 partition 01 start sector 63, # sector 2056257
Mar 25 14:38:58 bauer4 kernel: viod: Disk 04 partition 02 start sector 2056320, # sector 530145
Mar 25 14:38:58 bauer4 kernel: iSeries Real Time Clock Driver v1.0
Mar 25 14:38:58 bauer4 kernel: NET4: Linux TCP/IP 1.0 for NET4.0
Mar 25 14:38:58 bauer4 kernel: IP Protocols: ICMP, UDP, TCP, IGMP
```

```
Mar 25 14:38:58 bauer4 kernel: IP: routing cache hash table of 2048 buckets, 16Kbytes
Mar 25 14:38:58 bauer4 kernel: TCP: Hash tables configured (established 16384 bind 16384)
Mar 25 14:38:58 bauer4 kernel: NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
Mar 25 14:38:59 bauer4 kernel: devfs: v0.102 (20000622) Richard Gooch (rgooch@atnf.csiro.au)
Mar 25 14:38:59 bauer4 kernel: devfs: boot_options: 0x0
Mar 25 14:38:59 bauer4 kernel: (viodasd.c:689) DBG: check_change
Mar 25 14:38:59 bauer4 kernel: VFS: Mounted root (ext2 filesystem) readonly.
Mar 25 14:38:59 bauer4 kernel: Mounted devfs on /dev
Mar 25 14:38:59 bauer4 kernel: Freeing unused kernel memory: 160k init
Mar 25 14:38:59 bauer4 kernel: Unable to find swap-space signature
Mar 25 14:38:59 bauer4 kernel: (viodasd.c:689) DBG: check_change
Mar 25 14:38:59 bauer4 kernel: Unable to find swap-space signature
Mar 25 14:38:59 bauer4 rc: Starting keytable:  succeeded
Mar 25 08:38:51 bauer4 rc.sysinit: Mounting proc filesystem:  succeeded
Mar 25 08:38:51 bauer4 sysctl: net.ipv4.ip_forward = 0
```

3.8 Security issues

This section discusses some security issues especially related to server systems.

3.8.1 Changing file ownership (chown and chgrp)

Every file on UNIX systems belongs to an owner and to a group. The owner is characterized by its user ID, where the group is determined by the group ID. The command to change ownership is **chown**, for example:

```
chown mac file
```

Similarly, there is the **chgrp** command that can be used to change the owning group like this:

```
chgrp users file
```

However, the same can be achieved with the **chown** command, using:

```
chown .users file
```

In addition, the two actions can be combined like this:

```
chown mac.users file
```

Note that only *root* can arbitrarily change a file's owner. It would be surprisingly harmful for security if any user could create a file that belongs to another user, especially a file that belongs to root.

3.8.2 Access control

The system administrator can define access control policies by using file permissions. The file permission consists of nine bits that determine access permissions to the file, plus three special bits. This mechanism allows the administrator to define access permissions for three classes of users: the file owner, the owning group, and the rest of the world. The system administrator as well as the file owner can modify these permission bits by using the **chmod** utility.

3.8.3 Permission bits

When a user runs an **ls -l** command on a file, the first ten characters of output represent the file mode field. See Example 3-15.

Example 3-15 Example of ls -l

```
$ ls -l
total 8
drwxrwxr-x    2 itsousr1 itsousr1     4096 Mar 29 16:32 dir1
-rw-rw-r--    1 itsousr1 itsousr1       15 Mar 29 16:33 file1.txt
```

Table 3-3 shows the interpretation of the file mode characters “r”, “w”, and “x” for a file.

Table 3-3 Permissions for a file

Character	Mode name	Description
r	read	Can look at the contents of the file
w	write	Can change or delete the contents of the file
x	execute	Can execute the file as a command (for shell scripts, normal users also need r)

Table 3-4 shows the interpretation of the file mode characters “r”, “w”, and “x” for a directory.

Table 3-4 Permissions for a directory

Character	Mode name	Description
r	read	Can find out what files are in the directory
w	write	Can create and remove files from the directory (x is also needed)
x	execute	Can access files in the directory

In addition to the three sets of permissions listed above, a file’s permissions have three special components that affect only executable files (programs) and, on some systems, directories:

- Set the process's effective user ID to that of the file’s owner upon execution (called the *setuid bit*). No effect on directories.
- Set the process's effective group ID to that of the file upon execution (called the *setgid bit*). For directories, put files created in the directory into the same group as the directory, regardless of the group in which the user who creates them exists.
- The traditional meaning for files was to hold the program image in memory so it will load more quickly when it runs (that’s where the name “sticky bit” comes from). In Linux, the sticky bit on files is ignored. For directories, prevent users from removing or renaming a file in a directory unless they own the file or the directory, this is called the *restriction deletion flag* for the directory.

3.8.4 Setting permissions

The **chmod** utility changes the permissions of each given file according to mode operators, which can be either a symbolic representation of changes to make or an octal number that represents the bit pattern for the new permissions.

Symbolic mode

The format of a symbolic mode is

```
chmod [ugoa...][[+-=][rwxXstugo...]...][,...] files...
```

Multiple symbolic operations can be given, separated by commas. Using the **chmod** command with symbolic notation, you specify changes relative to the existing permissions on a file or directory by adding or deleting permissions.

Table 3-5 shows the first set of parameters that select which users' access to the file will be changed.

Table 3-5 Target parameters of chmod

Character	Meaning
u	File owner
g	Group
o	All others
a	User, group, and all others

Table 3-6 shows the second set of parameters that select whether permissions are to be removed, added, or set.

Table 3-6 Operation parameters of chmod

Operator	Meaning
-	Remove specified permissions
+	Add specified permissions
=	Clears selected permission field and sets it to the mode specified

Table 3-7 shows the third set of parameters that select the permissions.

Table 3-7 Permission parameters of chmod

Character	Meaning
r	To set read permission
w	To set write permission
x	**File**: To set execute permission **Directory**: To set search permission
X	**File**: To set execute permission only if the file already has execute permission for some user **Directory**: To set search permission
s	To set user or group ID on execution
t	**File**: To stay permanently on the swap device **Directory**: To prevent users from removing or renaming a file in a directory unless they own the file or the directory
u	To set the permissions that the user who owns the file currently has
g	To set the permissions that other users in the file's group have
o	To set the permissions that other users not in the file's group have

To remove execute permission for the file owner and change group and for others to write permission of the file1.txt file, you would use **chmod** as shown here:

```
$ chmod u-x,go+w file1.txt
```

Numeric mode

A numeric mode is from one to four octal digits (0-7), derived by adding up the bits with values 4, 2, and 1. Any omitted digits are assumed to be leading zeros.

Table 3-8 shows the octal values of each symbolic notation.

Table 3-8 Octal value of symbolic notations

Translate binary to octal				
User	**s**	**r**	**w**	**x**
Octal value	4000	400	200	100
Group	**s**	**r**	**w**	**x**
Octal value	2000	40	20	10
Others	**t**	**r**	**w**	**x**
Octal value	1000	4	2	1

Table 3-9 shows how to get numeric values of the symbolic notation “rwsrw-r-x”.

Table 3-9 Symbolic - Binary - Octal

	Special	**User**	**Group**	**Other**
Symbolic Notation	--s------	rwx	rw-	r-x
Binary (permission = 1) (lack of permission = 0)		111	110	101
Octal	4000	400+200+100=700	40+20+0=60	4+0+1=5
Numeric value	**4765**			

Example 3-16 shows how to use octal digits from one to four.

Example 3-16 Example of using octal digits of chmod

```
$ ls -l
total 16
-rwxr-xr-x    1 itsousr1 itsousr1    14521 Mar 29 18:26 a_program
$ chmod 5 a_program
$ ls -l
total 16
-------r-x    1 itsousr1 itsousr1    14521 Mar 29 18:26 a_program
$ chmod 75 a_program
$ ls -l
total 16
----rwxr-x    1 itsousr1 itsousr1    14521 Mar 29 18:26 a_program
$ chmod 755 a_program
$ ls -l
total 16
-rwxr-xr-x    1 itsousr1 itsousr1    14521 Mar 29 18:26 a_program
$ chmod 7775 a_program
```

```
$ ls -l
total 16
-rwsrwsr-t    1 itsousr1 itsousr1    14521 Mar 29 18:26 a_program
```

3.8.5 Ext2 partition specific access control

In addition to traditional UNIX-like file permissions, the ext2 file system has a special access control list. See Table 3-10.

Table 3-10 Access control list attributes of the ext2 file system

Character	Description
a	To set to be open only in append mode for writing
c (unimplemented)	To set to be automatically compressed on the disk
d	To set to not be a candidate for backup when the dump program is run
i	To set disabled to modify. It cannot be deleted or renamed; no link can be created to this file and no data can be written to the file
s (unimplemented)	To set to be deleted, its blocks are zeroed and written back to the disk
S	To set to be modified, the changes are written synchronously on the disk
u (unimplemented)	To set to be deleted, its contents are saved. This allows the user to ask for its undeletion

You can display these attributes by the **lsattr** command and can change by the **chattr** command except for “a” and “i” (these can be executed by the administrator).

3.8.6 Pluggable Authentication Modules (PAM)

In the past, the problem was that each time a new authentication scheme was developed, it required all the necessary programs (**login**, **ftpd** etc...) to be rewritten to support it. That’s when PAM came up. The official document for PAM is located on the Web at:
http://www.opengroup.org/tech/rfc/rfc86.0.html

What PAM is

PAM is a unified authentication scheme used for most Linux systems. PAM allows you to change your authentication methods and requirements on the fly. It also allows you to encapsulate all local authentication methods without recompiling any of the applications.

Linux-PAM is designed to provide the system administrator with a great deal of flexibility in configuring the privilege granting applications of their system. The local configuration of those aspects of system security controlled by Linux-PAM is contained in one of two places:

- The /etc/pam.conf single system file
- The /etc/pam.d/ directory

For an exact specification of /etc/pam.conf or the files the /etc/pam.d/ directory, refer to:
http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam-4.html

Using /etc/pam.conf

Example 3-17 shows the /etc/pam.conf file. This file specifies the authentication method of an FTP server.

Example 3-17 Example of the /etc/pam.conf file

```
# ftp authorization
# Server Type      Control     Module                          Arguments
ftp      auth      required    /lib/security/pam_listfile.so \
       item=user  sense=deny file=/etc/ftpusers onerr=succceed
ftp      auth      required    /lib/security/pam_unix_auth.so
ftp      auth      required    /lib/security/pam_shells.so
ftp      account   required    /lib/security/pam_unix_acct.so
ftp      session   required    /lib/security/pam_unix_session.so
```

Basically, a line in /etc/pam.conf looks like this:

```
service-name   module-type   control-flag   module-path   arguments
```

The fields are described here:

- **service-name:** The name of the service associated with this entry. Frequently the service name is the conventional name of the given application, for example: **ftpd**, **rlogind**, **su**, etc.
- **module-type:** One of four types of modules (auth, account, session, password).
- **control flag:** Used to indicate how the PAM library will react to the success or failure of the module it is associated with.
- **module-path:** The path name of the dynamically loadable object file; the pluggable module itself.
- **arguments:** A list of tokens that are passed to the module when it is invoked.

As noted before, for a complete specification, refer to the Web site at:
http://www.opengroup.org/tech/rfc/rfc86.0.html

Using /etc/pam.d

The syntax of each file in /etc/pam.d/ is similar to that of the /etc/pam.conf file and is made up of lines in the following form:

```
module-type   control-flag   module-path   arguments
```

The only difference is that the service-name is not present. The service-name is the name of the given configuration file. For example, /etc/pam.d/rlogin contains the configuration for the rlogin service.

Example 3-18 shows the /etc/pam.d/rlogin file.

Example 3-18 The /etc/pam.d/rlogin file

```
auth       required     /lib/security/pam_securetty.so
auth       sufficient   /lib/security/pam_rhosts_auth.so
auth       required     /lib/security/pam_pwdb.so shadow nullok
auth       required     /lib/security/pam_nologin.so
account    required     /lib/security/pam_pwdb.so
password   required     /lib/security/pam_cracklib.so
password   required     /lib/security/pam_pwdb.so shadow nullok use_authtok
session    required     /lib/security/pam_pwdb.so
```

The first line specifies to check whether an *rlogin* request comes from the permitted terminal.

The second line says that the authentication is sufficient to permit the user to log in without the password check when a request hostname and user pair is listed in /etc/hosts.equiv or $HOME/.rhosts.

The other lines specify that when no password login authentication fails, the same method as local login is applied.

/etc/pam.conf versus /etc/pam.d/

We recommend that you use the directory structure configuration (/etc/pam.d). The reasons why are explained on the Web at:
http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam-4.html

Actually, which method to adopt depends on the distributor's policy.

3.9 National language support

This section discusses how to support different languages and different character sets on the system.

3.9.1 Locale

The *locale model* is a basic concept that was introduced into ISO C (ISO/IEC 9899:1990). In the locale model, the behavior of some C functions depend on the environment variables related to locale. POSIX also determines some standards around internationalization (i18n). Almost all of POSIX and ISO C standards are included in X/Open Portability Guide (XPG4) and all of them in XPG5. Note that XPG5 is a part of UNIX specifications version 2.

3.9.2 Locale names

The locale names are based on Appendix E of the POSIX 1003.1 standard. The `<locale>` string is:

`<language>[_<territory>][.<codeset>][@option][,version]`

The `<language>` field is taken from ISO 639 (a code for the representation of names of languages). It is two characters wide and specified with lowercase letters only. For example, `en` is for English, `de` is for German, and `zh` is for Chinese.

The `<territory>` field is the two-letter code of ISO 3166 (a specification of representation of countries), if possible. It is two characters wide and specified with uppercase letters only. For example, `GB` is for United Kingdom, `KR` is for Republic of Korea (South Korea), and `JP` is for Japan.

The `<codeset>` field represents the encoding set of the character set. No standard exists for the codeset. If the `<codeset>` field is just a numeric specification, the number represents the number of the international standard describing the character set.

The `<modifier>` field is used in a site-specific manner. But the GNU C library uses an "@euro" modifier to specify the locale names in which the euro sign is available.

The `<version>` field may be unused for the Linux system.

Table 3-11 shows some examples of locale name.

Table 3-11 Examples of locale name

Language	Territory	Codeset	Locale name
English	United Kingdom	ISO 8859-1	en_GB
English	United States	ISO 8859-1	en_US
French	Canada	ISO 8859-1	fr_CA
French	France	ISO 8859-1	fr_FR
French	France	ISO 8859-15	fr_FR@euro
German	Germany	ISO 8859-1	de_DE
German	Switzerland	ISO 8859-1	de_CH
Japanese	Japan	EUC-JP	ja_JP.eucjp
Korean	Korea	EUC-KR	ko_KR.euckr
Simplified Chinese	Great China	GB18030	zh_CN.gb18030
Simplified Chinese	Hong Kong	GB2312	zh_HK
Traditional Chinese	Taiwan	EUC-TW	zh_TW.euctw
Traditional Chinese	Taiwan	BIG5	zh_TW.big5

Note that the locale names of “C” and “POSIX” are determined for the names for default behavior. For example, when an application needs to parse the output of the date command, it has to set the value of the environment variable to “C” or “POSIX”.

3.9.3 Locale categories and settings

The environment variables related to the locale model are divided into six categories (Table 3-12). Each of these categories can be set independently as an environment variable.

Table 3-12 Category of locale

Environment variable	Description
LC_COLLATE	Characters and strings collation sequences
LC_CTYPE	Character classification such as upper, lower, space, and so on
LC_MESSAGES	Output messages
LC_MONETARY	Monetary format information
LC_NUMERIC	Numeric format information
LC_TIME	Date and time conversion parameters

In addition to these six categories, there are two environment variables, LC_ALL and LANG. Application programs determine the locale on which they have to run in the following manner:

1. Consult the LC_ALL environment variable.
2. If LC_ALL is not available, consult the environment variable that is the same as the name of the locale category (for example, LC_MESSAGES).
3. If none of them are available, consult the LANG environment variable.
4. If LANG is not set, this is set to the value POSIX (LANG=”POSIX”).

A user should use the LANG variable to set their locale because some application may want to set the LC_ALL or LC_xxx variables to override a user's locale value temporarily (for example, startup shell script).

Example 3-19 shows setting the LANG variable and seeing a user's current locale settings by the **locale** command.

Example 3-19 Using the environment variables to set a locale

```
$ export LANG=ja_JP.eucJP
$ locale
LANG=ja_JP.eucJP
LC_CTYPE="ja_JP.eucJP"
LC_NUMERIC="ja_JP.eucJP"
LC_TIME="ja_JP.eucJP"
LC_COLLATE="ja_JP.eucJP"
LC_MONETARY="ja_JP.eucJP"
LC_MESSAGES="ja_JP.eucJP"
LC_PAPER="ja_JP.eucJP"
LC_NAME="ja_JP.eucJP"
LC_ADDRESS="ja_JP.eucJP"
LC_TELEPHONE="ja_JP.eucJP"
LC_MEASUREMENT="ja_JP.eucJP"
LC_IDENTIFICATION="ja_JP.eucJP"
LC_ALL=
```

Example 3-20 illustrates that the output messages of the **ls** command are translated into the language specified by the LANG environment variable.

Example 3-20 Translated messages of ls output

```
# ls
# export LANG=en_US
# ls aaa
ls: aaa: No such file or directory
# LANG=fr_FR
# ls aaa
ls: aaa: Aucun fichier ou répertoire de ce type
# LANG=de_DE
# ls aaa
ls: aaa: Datei oder Verzeichnis nicht gefunden
```

The first line means there is no file in the current directory.

The language is set to English (United States) by the **export** command.

Then the output of **ls aaa** (*aaa* is a file name that doesn't exist in this directory) is output in English.

The rest of the lines show the output messages are translated into French and German.

When you want to see the current installed locale names on your system, use the **locale -a** command. But the result of this command is usually a list of over one hundred locale names, so you may use this command with the pipe (|) before the **more** or **less** command.

```
$ locale -a | less
C
```

```
POSIX
af_ZA
ar_AE
ar_BH
ar_DZ
ar_EG
...
```

3.9.4 Locale definition file

The locale definition file is stored in the /usr/share/i18n/locales directory. The definition file contains all the information about its cultural conventions. The general information is in the top of the file, followed by multiple sections to describe each locale category.

Usually you don't need to create a new locale definition file from scratch, because over one hundred locales are provided already.

You can create your own locale object by using the **localedef** utility. This example shows you how to create another Japanese locale using UTF-8 characteristics:

```
$ cd /usr/share/i18n
$ localedef -i ./locales/ja_JP -f ./charmaps/UTF-8 -c ja_JP.UTF-8
```

Now, you get a new locale "ja_JP.UTF-8" at /usr/lib/locales/ja_JP.utf8. Note that the codeset part will be normalized to all lowercase characters without the "-" sign.

3.10 Backup and recovery

There are two approaches to backup and recovery. From the OS/400 side, you can save the network server storage spaces (NWSSTG) that correspond to the virtual disks in a Linux system. The other possibility is to use the Linux commands to do this, so you look at it from the side of the Linux system. Which method is best is not a simple question. It depends on the situation. Therefore, this section looks at both approaches.

You can find more information about backup strategies and automating the backup process, along with general information about backup and restore, on the Web at:
http://www.backupcentral.com/

3.10.1 Backup from the Linux side

Doing a backup from the Linux side is probably the best way to save the system. You can use the following standard Linux commands to save to files or to tape drives:

- Saving to tape drives could include virtual or direct attached tape drives. To do this, you do not need to power down the Linux system before doing a backup as opposed to doing a backup from OS/400. The examples given in this chapter all write to files, but as shown in "Virtual tape" on page 94 more specifically, tape drives are seen as plain files by the Linux system. Therefore, all commands given here also apply to tape drives.
- If you save to file, you could transfer the file to an OS/400 partition and then back up the file along with the OS/400 backup process. This transfer could happen using FTP or NFS, or you could mount the iSeries IFS using Samba.

By mounting (**mount** command) the required export into the OS/400 mount points, you can use the OS/400 SAV command to do file-level backups of the Linux data. Then you can use RST to restore even individual files as long as the UID and GID on OS/400 are set correctly. See Chapter 11, “Samba” on page 231, for more information about setting up Samba.

Backup to virtual tape drive

As mentioned in the beginning of this chapter, to Linux, almost everything is a file. Devices, such as tape drives (virtual and direct), are also seen as files in the /dev file system (see 3.2, “Devices” on page 90). All the examples given in the following sections are done to file. However, they simply replace the filename by the name of the device to which you want to backup, and you are backing up to that device.

Note: In all following examples, you can take the filename (to backup to or restore from), and replace it by the device name of the tape drive you want to use for backups. For example, for virtual tape, you will use `/dev/viotapeX`, where *X* is the device number. For native tape, you use `/dev/tapX`.

The Linux kernel

There are several ways the Linux kernel is loaded, depending on the location of the Linux kernel. This parameter can be set in the NWSD IPL source option. Refer to 2.4.3, “Creating an initial network server description (CRTNWSD)” on page 38, and 4.4, “Installing the kernel” on page 162, for more information about IPL sources. In most cases, it resides in a disk partition of the NWSSTG. It could boot up from a stream file in the iSeries IFS or from objects in the iSeries memory space.

Be sure to back up this IPL source also if you want a complete backup of your Linux system. Back up the stream file or run the SAVSYS command to save the A and B load sources when you use them.

Linux backup commands

In general, there are three possibilities to back up the system with Linux commands. The command names for backup and restore are the same, except for the **dump** command. Backups created with the **dump** command have to be restored with the **restore** command.

How you do this depends on:

- The kind of backup you want to create (quick backup or disaster recovery backup, logical or physical backup)
- The commands with which you are familiar

For an overview, see Table 3-13.

Table 3-13 Overview of the Linux backup commands

Characteristic/task	Dump/restore	GNU cpio	GNU tar
Complexity	Complicated, but good for multi-level backup	Mostly simple, needs find	Simple, needs find
Incremental backup	Yes, through levels	Yes, but has to use find	Yes
Multi-volume archive	Yes	Yes	Yes

Characteristic/task	Dump/restore	GNU cpio	GNU tar
How to find a list of archived files	Simple, index in front (restore -r)	Search entire file (cpio -it)	Search entire file (tar -t)
Find specific files	Interactive (supports `ls` and `cd` commands)	Complex, search entire file; wildcards are allowed	Complex, search entire file; no wildcards allowed
Backup protocol	Generate after backup with restore -t dumplog	cpio -v 2> cpiolog	tar cvf 2> tarlog
Restore archive with absolute path to different location	Always relative to current work directory	Limited functionality with cpio -I	Complicated, only by using chroot
Uses	System backup	System backup, transfer files between systems	Quick backup, transfer files between systems

The dump and restore commands

The **`dump`** command is the only command that directly supports incremental backup. The **`restore`** command has powerful functionality, such as interactive restore and indexing.

Backup using dump

To back up the file system, issue the **`dump`** command:

```
dump <level> unbBf blocking_factor records archive_file fs_to_save
```

The options are:

<level> A number from 0 to 9 (level 0 = full backup, 1 to 9 = incremental backups).

b The blocksize; the number of kilobytes per dump record.

B The number of dump records per volume; this option overrides the calculation of tape size based on length and density. Specify a very high number, for example 10000000, to inactivate tape size calculation when dumping to DASD (otherwise dump prompts for a new tape during backup).

u Update the /etc/dumpdates file after a successful dump.

n Notify the members of a specified group whenever dump requires attention.

f Write the backup to filename (**`f archive_file`**).

There are two other options used for informational purposes:

w List information on all file systems.

W List those file systems that need to be backed up (dumped).

When you run the **`dump`** command for the first time on a system, you have to create an empty /etc/dumpdates file. This file must be owned by *root*:

```
touch /etc/dumpdates
```

Dump writes a table of contents at the beginning of the archive file. This index is created before the backup data is written. It does not honor changes made while the dump task is active.

The *level* of the dump command specifies how often a full backup should be executed and how often an incremental backup should be executed. Use level 0 to run a full backup; use levels 1 to 9 for the incremental backups, where each level backs up whatever was changed since the last backup with the next lower level.

Table 3-14 presents a concept using incremental backup.

Table 3-14 A concept using incremental backup

Day	Level	Data backup
1	0	Full backup
2	3	All changes since day 1
3	2	All changes since day 1
4	5	All changes since day 2
5	4	All changes since day 2
6	7	All changes since day 3
7	6	All changes since day 3

This example is for a large system with lots of data:

- Once a week, a full backup is taken; in this example, we'll call this day 1.
- On day 2, a level 3 backup is taken that saves all changes since the last full backup on day 1.
- On day 3, the level 2 backup saves all changes done after day 1. In a recovery situation, after day 3 where you need to restore the latest files, you would only need the backups of day 1 and day 3.

Day 4 to 7 do the same switching of the backup levels as day 2 and 3. This enables you to have more than two backup levels, without having to restore up to 7 backups in case of a crash. At worst, you would need the levels 0, 2, 4, and 6 for a recovery on day 7.

The restore command

To restore data saved with the **dump** command, use the **restore** command:

```
restore [trxi] vbfy blocking_factor archive_file fs_to_restore
```

The restore type/function can be specified with the following options:

t Display the content of a backup
r Restore the entire archive file (recursive)
x Extract only listed files from the archive file
i Interactive restore

The restore behavior is influenced by the following options:

v The verbose option; displays detailed information during restore.

b Specify the blocking factor (block size) used for the **dump** command. If this option is not specified, **restore** tries to determine the block size dynamically.

f Specify the archive file, **-f archive_file**. If you do not specify the archive file, **restore** expects input from the default tape drive.

y Attempt to recover from read errors (skip over bad blocks).

Option **-r** requires starting with the level 0 archive file. You have to restore in *level order*; otherwise, the restore fails.

As the backup of the files is done with relative paths, you have to change to the directory to where you want to restore the file before starting the restore.

For option **-x**, specify the exact path and file name. To restore two files named *file1* and *file2* from the directory *dir1*, enter:

```
restore rbvfy 126 /archives/archive.file.dump ./dir1/file1 ./dir1/file2
```

To determine what's in the archive file, you can create a table of contents with:

```
restore tbfy 126 /archives/home.day.dump
```

For the interactive option of **restore**, call restore with **-i**. This provides a shell with limited functionality. You can use the following commands:

- **cd** for change directory
- **ls** to list files
- **pwd** to list the current directory name

In addition, you can issue restore-specific commands like **add**, **delete**, and **extract**.

To *add* a file to the restore list, use **add filename** or **add *pattern***. This marks the selected file with an asterisk (*) when displaying the file list with **ls**.

To *delete* a file from the restore list, use **delete filename** or **delete *pattern***. After marking all the files you want to restore, initiate the restore with the **extract** command.

Here is an example of a dump with an interactive restore session:

1. Dump the directory *openoffice60/user* in */tmp/dump.oo*:

```
[root@bauer4 second]# dump 0bBf 64 1000000 /tmp/dump.oo openoffice60/user
  DUMP: Date of this level 0 dump: Thu Mar 22 10:56:50 2001
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/viod/disc1/part1 (/mnt/second (dir /openoffice60/user)) to
/tmp/dump.oo
  DUMP: Label: none
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 1509 tape blocks on 0.00 tape(s).
  DUMP: Volume 1 started at: Thu Mar 22 10:56:51 2001
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: Closing /tmp/dump.oo
  DUMP: Volume 1 completed at: Thu Mar 22 10:56:51 2001
  DUMP: 1606 tape blocks (1.57MB) on 1 volume(s)
  DUMP: finished in less than a second
  DUMP: Date of this level 0 dump: Thu Mar 22 10:56:50 2001
  DUMP: Date this dump completed:  Thu Mar 22 10:56:51 2001
  DUMP: Average transfer rate: 0 KB/s
  DUMP: DUMP IS DONE
```

2. Create a table of contents to verify the dump:

```
[root@bauer4 second]# restore tbfy 64 /tmp/dump.oo
Dump   date: Thu Mar 22 10:56:50 2001
Dumped from: the epoch
Level 0 dump of /mnt/second (dir /openoffice60/user) on
bauer4.mydomain.com:/dev/viod/disc1/part1
Label: none
```

```
         2      .
     96577      ./openoffice60
     96578      ./openoffice60/user
     16097      ./openoffice60/user/config
     16100      ./openoffice60/user/config/javarc
     96579      ./openoffice60/user/config/registry
     96594
./openoffice60/user/config/registry/org.openoffice.Office.Filter.Calc.xml
     96750
./openoffice60/user/config/registry/org.openoffice.Office.Filter.Draw.xml
     96759      ./openoffice60/user/config/registry/org.openoffice.Inet.xml
     96760      ./openoffice60/user/config/registry/org.openoffice.ucb.Configuration.xml
...
```

3. Change into the destination directory for restore:

```
cd /tmp
```

4. Call the interactive restore, select a file to restore, and extract it from the dump file (all actions are in bold letters):

```
[root@bauer4 /tmp]# restore -if dump.oo
restore > ls
.:
openoffice60/

restore > cd openoffice60/
restore > ls
./openoffice60:
user/

restore > cd user/
restore > ls
./openoffice60/user:
autocorr/    basic/       database/    sofficerc    temp/        work/
autotext/    bookmark/    gallery/     sregistryrc  template/
backup/      config/      plugin/      store/       wordbook/

restore > add work
restore > ls
./openoffice60/user:
 autocorr/     bookmark/     plugin/       temp/
 autotext/     config/       sofficerc     template/
 backup/       database/     sregistryrc   wordbook/
 basic/        gallery/      store/       *work/

restore > extract
You have not read any tapes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
set owner/mode for '.'? [yn] n
restore > quit
```

5. Look at the restored directory:

```
[root@bauer4 /tmp]# ls -l
total 7624
-rw-r--r--    1 root     root      1638400 Mar 22 10:56 dump.oo
drwxr-xr-x    3 root     root         4096 Mar 20 16:43 openoffice60
-rw-r--r--    1 root     root      6130797 Mar 21 16:37 samba-2.0.7-20000425.ppc.rpm
```

The restore information is created in the current directory, and the selected directory for restore is *work/* in *openoffice/* of the current directory.

The cpio command

The archive utility **cpio** packs files together like tar does. It creates and reads file archives. The restore processing includes recovery from data corruption in an archive file.

Because cpio has many options, it is not easy to handle. You should practice using the **cpio** command so you know how to use it efficiently in the case of a recovery situation.

Backup using cpio

By default, **cpio** reads the list of files to be archived from standard input (stdin) and writes the archive to standard output (stdout):

```
cpio -o [aBcv] [-C block_value]
```

Create the list of files to be archived with either the **ls** or **find** command:

```
ls | cpio -oacvB > /archives/backup.cpio
find . -print | cpio -oacvB > /archives/backup.cpio
```

The **find** command is more powerful and can be used to do a partial archive.

It is important to archive the files that are *relative* to the *current working directory*. This provides greater flexibility when restoring files. Archives created with absolute paths can only be restored to the original path (except when using **chroot** or **cpio -I** commands).

If you have a script that determines which files need to be archived, you can pass the output file (by using pipelines) of this script to the **cpio** command using the following command example:

```
cat /tmp/filelist | cpio -oacvB > /archives/backup.cpio
```

For the cpio archive function (create a backup), use the **-o** flag with any of the following useful options (**cpio -o<options>**):

-a Reset the access time to the value before the backup executed. This is important if you have processes that rely on a real access time. With the backup, all files would otherwise get the time of the last backup as access time.

-c This is the default for the GNU **cpio** used by Linux. Specify the ASCII header format, because this is the most compatible format across platforms.

If you archive and restore only with GNU **cpio**, you can use the **-newc** option instead. This is the new ASCII header format that supports bigger file systems.

-v Print a list of files that are archived to standard error, because standard output is reserved for the archive file itself.

-B Block size for input and output; we recommend that you use 5120 bytes per record. The default is 512 bytes per record. You have to remember the block size for the restore process.

-C Like **-B**, but block size can be any positive integer (**-C<value>**).

-O Only available in the GNU **cpio**; it allows you to specify an output archive file instead of just writing to standard output (**-O filename**).

Restore using cpio

After the archive is done to restore a cpio archive, you need to know the exact block size of the archive (backup). Because Linux uses GNU cpio, the header format is detected by cpio automatically.

If you do not know how the archive was created, try to validate it by using the following command:

```
cpio -itv -C <block_size_you_expect> < /tmp/unknown.archive.cpio
```

For the **cpio** restore, use the **-i** flag with any of the following options:

```
cpio -i [options] [-C block_value][patterns]
```

Here is a detailed list of the options used to restore files:

t Generate a table of contents.

d Create directories as needed.

m Restore files with original modified-time instead of restore time.

u Unconditional overwrite all files.

“pattern” Restore files matching the pattern. Pattern can be repeated several times. Wildcards, like an asterisk (*), can be used at any position, including the first character. A pattern */myfile restores myfile from any path in the directory.

r Interactively rename files during restore.

When you only need specific files of the /archives/backup.cpio archive to be restored, you should first create a table of contents:

```
cpio -ipt < /archives/backup.cpio
```

To restore only the needed files, use the following command, where the pattern can be repeated several times and can contain wildcards:

```
cpio -iBdmuv “pattern1” “pattern2” < /archives/backup.cpio
```

To restore all files of the archive, change to the specific directory and use:

```
cpio -iBdmuv < /archives/backup.cpio
```

Incremental backup with cpio

Because cpio uses a file list to determine which files have to be archived, incremental backup is possible. With a little helper file, it is even possible to create different concept levels for the backups:

1. Start with a full backup of the directory to be archived, for example, an application directory named /appl:

   ```
   cd /appl
   ```

2. Create a dummy file in /appl to save the time of the backup:

   ```
   touch dummy.cpio.level.0
   ```

3. Back up the files of the directory and all its sub-directories:

   ```
   find . -print | cpio -oacvB > /archives/appl.level.0.cpio
   ```

4. For the next backup, create a level 2 backup, which leaves level 1 free for consolidation:

   ```
   touch dummy.cpio.level.2
   find . -newer dummy.cpio.level.0 -print | cpio -oacvB > /archives/appl.level.2.cpio
   ```

5. The **find** command with the option **-newer** looks for files that were modified more recently than the file specified, which in our case, was the dummy file dummy.cpio.level.0.

   Perform a level 3 and level 4 backup on the following days. Each backup refers to the dummy file level 1 (for example, level 4 refers to dummy.cpio.level.3). Now create a level 1 backup:

```
touch dummy.cpio.level.1
find . -newer dummy.cpio.level.0 -print | cpio -oacvB > /archives/appl.level.1.cpio
```

6. The appl directory itself indicates the sequence of the backups through the timestamps of the dummy files:

```
[root@linux6 /appl]# ls -l
total 0
-rw-r--r--   1 root     root            0 May 10 10:30 dummy.cpio.level.0
-rw-r--r--   1 root     root            0 May 17 10:33 dummy.cpio.level.1
-rw-r--r--   1 root     root            0 May 12 10:31 dummy.cpio.level.2
-rw-r--r--   1 root     root            0 May 14 10:29 dummy.cpio.level.3
-rw-r--r--   1 root     root            0 May 16 10:32 dummy.cpio.level.4
```

The GNU tar command

tar is an easy-to-use archive command that is used in everyday system work, not just for backup purposes. It should be used for quick backups.

tar is a more portable command than other backup commands that we have discussed so far. Even non-UNIX operating systems, such as Windows, can read **tar** archives, for example, with programs like WinZip.

Backup using the tar command

To create a backup with the **tar** command using the **-c** option, enter:

```
tar -cvpf /archives/backup.tar "pattern"
```

The options are:

c Create a new archive.

v Print information while archiving (verbose).

f Specify an output file name **-f archive.tar**.

p Save all file attributes, for example, file permission bits, ownership bit.

pattern Archive files matching the pattern. Wildcards are allowed except for the first character.

A command to back up the entire file system would look like this example:

```
tar -cvpf /archive/full-backup-mydate.tar -directory / -exclude=mnt -exclude=proc
-exclude=archives
```

Exclude all directories that are mounted temporarily, special file systems like /proc, and the directory containing the archives with the **-exclude** option of tar.

GNU **tar** provides the useful option **-W** to verify the archive after it was written:

```
tar -cvpWf /archives/backup.tar <pathname><filename>
```

A table of contents for a tar file can be created with:

```
tar -tf /archives/backup.tar > /archives/backup.tar.toc
```

Restore using the tar command

To restore an archive file with the **-x** option, type:

```
tar -xvpf /archives/backup.tar ["pattern"]
```

To preserve the original owner of the files, you have to call **tar** from a superuser ID. The options are:

x Restore from an archive file.

v Print information while restoring (verbose).

f Specify an archive name as input **-f archive.tar**.

p Restore all file attributes, for example, file permission bits, ownership bit.

pattern Restore files matching the pattern. The pathname and filename must *exactly* match the name in the archive file. Wildcards are not allowed. To work with wildcards during restore, use **tar** in combination with the **grep** command.

Incremental backup with tar

There are only limited ways to use **tar** to create an incremental backup. You can create a list of files to be archived with the **find** command. This can be done similar to the cpio incremental scenario in “Incremental backup with cpio” on page 128. The following example uses a kind of manual incremental backup based on the **find** command.

The **find** command can produce a list of files that were changed since a certain period of time (in this example, 24 hours):

```
find / -mtime -1 \! -type d -print > /tmp/mylist
```

All directory entries are excluded with **\! -type d**, and the output of **find** is written to /tmp/mylist. If NFS file systems are connected to your system, the mylist file also includes files of connected NFS file systems. If you want to exclude the NFS file systems from your backup, you have to edit mylist and delete the NFS-specific entries.

Specify the **-T** option of **tar** to read the file name list as input for the backup:

```
tar -cv -T /tmp/mylist -f /archives/backup.incremental.tar
```

Compressing with gzip

Tar files are not compressed. **gzip** is a tool that zips one file and creates a new zipped file with the extension .gz (and deletes the original file as well).

Use this command to zip tar files such as:

```
gzip tarfile.tar
```

This creates a file with the name tarfile.tar.gz, which sometimes is renamed to tarfile.tgz.

Unzipping can be done using **gunzip**.

To compress immediately when tarring or uncompress when untarring, you can specify the **-z** option to **tar**.

Compressing with bzip2

A newer compression tool exists, similar to **gzip** (with the same command options), called **bzip2**. While **bzip2** has a better compression ratio (about 20%), it takes more time to compress as well. **tar** now has an extra option called **-I** (may be changed to **-j** in the latest version), which uses **bzip2** compression/decompression, similar to the **-z** option for **gzip**.

bzip2 is not installed on all distributions, but it can be easily installed from an rpm package.

Important: See the usage of **tar** (by **tar --help** or **man tar**) to check which option (**-I** or **-j**) is used to specify **bzip2** compression.

3.10.2 Backup from the OS/400 side

If you look at backup from the side of OS/400, you can save the network server storage spaces (NWSSTG) that correspond to the virtual disks in a Linux system. The idea is more or less the same as backing up objects used by a Windows Network Server (Integrated PC Server, Integrated Netfinity Server, or Integrated xSeries Server for iSeries).

Two objects should be backed up:

- Network server description
- Network server storage space

These objects are saved as part of the iSeries server when you perform a full iSeries server backup. You can also specifically save the network server description and storage spaces that are associated with the Linux server on iSeries. Daily backup of the system drive is a good idea.

When doing a system save, be sure not to exclude user data, because these storage spaces are located in the internal file system (IFS) and would not be saved in that case. It is also wise to shut down network servers while doing the save.

The Linux kernel

There are several ways the Linux kernel is loaded, depending on the location of the Linux kernel. This parameter can be set in the NWSD IPL source option. For more information, see 2.4.3, “Creating an initial network server description (CRTNWSD)” on page 38, and 4.4, “Installing the kernel” on page 162. In most cases, it resides in a disk partition of the NWSSTG.

Important: Make sure you also back up this IPL source, in case it is not included as a *PReP Boot Partition* in the NWSSTG.

Backing up network server description

When you save the storage space objects that are associated with the Linux server on the iSeries, you do not really have to save the NWSD as well. However, it might be needed to save it. To save the NWSD, you use the Save Configuration (SAVCFG) command (illustrated in Figure 3-3):

1. On the iSeries command line, type:

   ```
   SAVCFG
   ```

2. Fill in the device name:
   - In the Device field, specify the device name if you are saving to media.
   - If you are saving to a save file, specify ***SAVF** and identify the name and library for the save file in the appropriate fields. If the savefile doesn’t exist yet, create it with the CRTSAVF command.
3. Press Enter for the iSeries server to save the NWSD configuration.

```
                        Save Configuration (SAVCFG)

Type choices, press Enter.

Device . . . . . > *SAVF         Name, *SAVF
             + for more values
Save file  . . . . . > LNXSAVCFG     Name
  Library  . . . . . >   QGPL        Name, *LIBL, *CURLIB

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display        F24=More keys
```

Figure 3-3 Saving a NWSD

This method backs up all configuration related objects. In a disaster recovery situation, you would restore all the configuration objects, which include the network server description for your Linux server. In some situations, for example when you want to create an identical Linux server on another iSeries (or the same machine), you need to specifically restore the NWSD.

To have the iSeries server automatically relink restored storage spaces in the Integrated File System to the appropriate NWSD, restore the NWSD after you restore the storage spaces. See 3.10.3, “Restore from the OS/400 side” on page 133, for more information.

Backing up network server storage spaces

To save network server storage spaces that are located in the system IFS directory named /QFPNWSSTG on the iSeries server, follow these steps:

1. If you are saving to tape, ensure that you have mounted a tape that is formatted for the iSeries server.
2. To prevent files being updated during the backup, shut down the Linux system. First, vary off the network server description by running:

   ```
   WRKCFGSTS *NWS
   ```

   Then vary off the NWS.
3. On the iSeries command line, type the SAV command and press F4 (illustrated in Figure 3-4).
4. Specify the device:
   - If you are saving the storage space to tape, specify the name of your tape device (for example, TAP01) in the Device field.
   - If you are saving the storage space to a save file instead of to tape, specify the path to the save file as the device. For example, to use a save file named LINUXBAK in library

QGPL, you would specify /QSYS.LIB/QGPL.LIB/LINUXBAK.FILE for the device. If the savefile doesn't exist yet, create it with the CRTSAVF command.

5. In the Name field under Objects, specify **/QFPNWSSTG/stgspc**, where *stgspc* is the name of the network server storage space. To add more than one storage space, you can specify them by entering a **+** in the "+ for more values" field.
6. Specify values for any other parameters that you want and press Enter to save the storage space.
7. Vary on the Linux server again, using the command:

```
WRKCFGSTS *NWS
```

```
                              Save Object (SAV)

Type choices, press Enter.

Device . . . . . . . . . . . . . > '/QSYS.LIB/QGPL.LIB/LNXSAV.FILE'

               + for more values

Objects:
  Name . . . . . . . . . . . . . > '/QFPNWSSTG/LINUX7'

  Include or omit  . . . . . . .   *INCLUDE      *INCLUDE, *OMIT
               + for more values
Directory subtree  . . . . . . .   *ALL          *ALL, *DIR, *NONE, *OBJ
Save active  . . . . . . . . . .   *NO           *NO, *YES, *SYNC









                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display        F24=More keys
```

Figure 3-4 Backup of a NWSSTG

3.10.3 Restore from the OS/400 side

To have an iSeries automatically relink restored storage spaces in the Integrated File System to the appropriate NWSD, restore the NWSD after you restore the storage spaces.

If you restore an NWSD of type *GUEST before restoring the predefined and user-defined storage spaces in the Integrated File System, you need to relink those storage spaces. You can do this by using the Add Network Server Storage Link (ADDNWSSTGL) command for each storage space that is associated with the NWSD:

```
ADDNWSSTGL NWSSTG(Storage_Name)NWSD(NWSD_Name)
```

Restoring network server description

In a disaster recovery situation, you would restore all the configuration objects, which include the network server description for your Linux server. In some situations, for example when you migrate to new Integrated xSeries Server hardware, you need to specifically restore the NWSD. To have the iSeries automatically relink storage spaces within the Integrated File System to the restored NWSD, restore those storage spaces first. To restore the NWSD, you use the Restore Configuration (RSTCFG) command:

1. On the OS/400 command line, type

   ```
   RSTCFG
   ```

   Press F4.

2. In the Objects field, specify the name of the NWSD (illustrated in Figure 3-5).
3. In the Device field, specify the device name if you are restoring from media. If you are restoring from a save file, specify ***SAVF** and identify the name and library for the save file in the appropriate fields.
4. Press Enter to have the iSeries restore the NWSD.

```
                        Restore Configuration (RSTCFG)

Type choices, press Enter.

Objects  . . . . . . . . . . . . > LINUX7D       Name, generic*, *ALL, *SRM
             + for more values
Device . . . . . . . . . . . . . > *SAVF         Name, *SAVF
             + for more values
Object types . . . . . . . . . . > *NWSD         *ALL, *CFGL, *CNNL, *COSD...
             + for more values
Save file  . . . . . . . . . . . > LNXSAVCFG      Name
  Library  . . . . . . . . . . . >   QGPL        Name, *LIBL, *CURLIB









                                                                        Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display        F24=More keys
```

Figure 3-5 Restoring a NWSD

Restoring network server storage spaces

Restore server storage spaces in the IFS with the Restore Object (RSTOBJ) command. If you are restoring from save media, ensure that you have mounted your media using the following steps.

1. If no server storage spaces currently exist on the system (none appear when you use the WRKNWSSTG command), you must create the /QFPNWSSTG directory before you can restore server storage spaces that you saved beneath that directory. This directory can be created manually or simply by creating an empty NWSSTG with these four steps:

   a. On the iSeries command line, type the following command to create a server storage space:

      ```
      CRTNWSSTG
      ```

      Then, press F4.

   b. Provide a name for the storage space.

   c. Use the minimal size allowed and specify the appropriate ASP.

d. Press Enter to create the storage space. OS/400 creates the storage space in the /QFPNWSSTG directory.

2. To restore the storage spaces, type **RST** and press F4. The display shown in Figure 3-6 appears.
3. In the Objects name field, specify **/QFPNWSSTG/*stgspc***, where *stgspc* is the name of the network server storage space.
4. Specify values for any other parameters that you want and press Enter to restore the storage space.

```
                              Restore Object (RST)

Type choices, press Enter.

Device . . . . . . . . . . . . . > '/QSYS.LIB/QGPL.LIB/LINUX7.FILE'

               + for more values

Objects:
  Name . . . . . . . . . . . . . > '/QFPNWSSTG/LINUX7'

  Include or omit  . . . . . .   *INCLUDE      *INCLUDE, *OMIT
  New object name  . . . . . .   *SAME

               + for more values
Directory subtree  . . . . . .   *ALL          *ALL, *DIR, *NONE, *OBJ
Output . . . . . . . . . . . .   *NONE



                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display        F24=More keys
```

Figure 3-6 Restoring a NWSSTG

3.11 Problem determination

This section provides some information to help you solve problems with Linux administration:

- At the time this redbook was written, the version of **xinetd** supplied with Red Hat is not very stable. Therefore, when encountering problems with login using Telnet, ftpd or httpd, upgrade the version of **xinetd**.

  The reported **xinetd** problem is fixed in the latest build. You can retrieve it via an anonymous FTP from ftp://ftp.wirespeed.com

  You can find it in the AS400 directory named xinetd-2.1.8.9pre14-6.ppc.rpm.

- Speed of the virtual Ethernet

  To enhance the speed of the virtual Ethernet, set the Maximum Frame Size of the line description (LIND) to 8996. Note that if you route connections to an external network card, like Ethernet, you should choose an appropriate Maximum Frame Size to avoid excessive fragmentation, resulting in a performance loss.

  To increase the Maximum Frame Size, you have to create a new line description because it is not possible to change the existing LIND's Maximum Frame Size.

Our results after adapting this parameter are illustrated in Table 3-15.

Table 3-15 Speed of the virtual LAN

Value	Speed
1496	2 MByte/s
8996	9 MByte/s

It should be noted that these values are for illustration only, but they show that there can be a marked improvement.

- To avoid data loss when power down, first vary off the network servers. When running the PWRDWNSYS *IMMED command, they will not properly shut down. When running the PWRDWNSYS *CNTRLD command, you do not know for sure whether it shut down properly, and something could have gone wrong.
- If you run into problems while working with devices or any other kernel related problem, use the **dmesg** command to print out the startup log. This can be of great help when resolving problems, for example:

```
[root@bauer4 /]# dmesg
Mapping load area - physical addr = 0, absolute addr = 1c000000
Load area size 32768K
HPT absolute addr = 24400000, size = 4096K
D-cache line size = 128  (log = 7)
I-cache line size = 128  (log = 7)
mf.c: iSeries Linux LPAR Machine Facilities initialized
Registering c00b92b0 in reg c01e9590
Registering c00b9a9c in reg c01e9598
Total memory: 264241152 bytes
Progress: [0300] - MMU:hash init
Progress: [0105] - hash:enter
iSeries_hashinit: added 24576 hptes to existing mapping
Progress: [0205] - hash:done
Progress: [0301] - MMU:mapin
Linux version 2.4.2 (root@bauer4.mydomain.com) (gcc version 2.96 20000731 (Red Hat Linux
7.1 2.96-75)) #2 Mon Mar 26 12:57:14 CST 2001
Progress: [3eab] - setup_arch: enter
Progress: [3eab] - setup_arch: bootmem
Max  logical processors = 32
Max physical processors = 16
Processor frequency = 500.04
Time base frequency = 500.04
Processor version = 3538945
Progress: [3eab] - arch: exit
On node 0 totalpages: 64512
zone(0): 64512 pages.
zone(1): 0 pages.
zone(2): 0 pages.
Kernel command line:
viocons registering console
viocons: in viocons_setup
Calibrating delay loop... 499.71 BogoMIPS
Memory: 251220k available (1232k kernel code, 736k data, 48k init, 0k highmem)
Virtual Bus TCE table built successfully.
...
```

- To solve software problems, use the log files in /var/log. Some daemons have their own log files in that directory, but most of them just use /var/log/messages. See 3.7.3, “System logs” on page 111.

4

Advanced administration and development

This chapter introduces how to develop applications on a Linux system. It also explains more complex administrative tasks that may be encountered when operating a Linux system. Plus it covers the basic development process on a Linux system, from compiling to debugging.

It explains how to:

- Create scripts
- Install programs
- Compile open source applications
- Recompile your own kernel

The purpose of this chapter is to give an overview of the possibilities that you have with a Linux system. It pays particular attention to fine tuning and installation for better performance or better security.

4.1 Development process

In Linux, almost anything is compiled in ANSI C. Integrated Development Environments (IDEs) are available, similar to Microsoft Visual Studio, but the Linux community does not use an IDE because of historical and cultural reasons. An experienced Linux programmer normally uses **emacs** or **vi** as an editor. If you are not familiar with one of these editors, try using **pico** in text environments or use one of the many notepad-like graphical editors in X Windows.

To automate the build process, makefiles are used. They automatically check dependencies to determine whether something has to be rebuilt.

To debug, **gdb** is most often used. Several graphical debuggers exist that provide you with a frontend to the gdb debugger, which is still used behind the scenes.

We advise that you search for additional information either on the Internet or in a bookstore. A quick link for more information is the *The Linux Programmer's Guide*, which is available on the Web at http://www.linuxdoc.org/LDP/lpg/index.html

For other formats (such as PS or PDF), go to:
http://www.ibiblio.org/pub/Linux/docs/linux-doc-project/programmers-guide/

4.1.1 Compiling using the GNU C compiler (gcc)

The GNU C compiler (**gcc**) is the standard compiler on any version of Linux. It also exists in cross-compile versions so that it is possible to compile sources on a PC to create executables for PowerPC processors.

Its invocation is quite straightforward. You can compile a source file named `helloworld.c` such as:

```
#include <stdio.h>

int main()
{
  printf("Hello World.\n");
  return 0;
}
```

Then you can create a runnable program named `helloworld` from this sourcefile by using:

```
[klaas@bauer4 klaas]$ gcc -o helloworld helloworld.c
```

When you try to run this program, make sure you explicitly specify the current path. Otherwise, it might seem that the shell cannot find the program you just built, and you can receive an error message when trying to run it:

```
[klaas@bauer4 klaas]$ helloworld
bash: helloworld: command not found
[klaas@bauer4 klaas]$ ./helloworld
Hello World.
```

As seen in the example, when running an executable, it might be possible that you need to explicitly run the program out of the current directory. The current directory is identified with a dot (.), such as in DOS. So, to run the program `helloworld` from the current directory, specify "`./`". If you want to avoid this problem, add "`:.`" to your environment variable *PATH*. This trick makes sure your current directory is searched when looking for a program.

Important: Never create a program named *test*, because a command named **test** already exists, which is called when you intend to run your program. If you would explicitly write the directory when calling the program (using **./test** instead of just **test**), this problem would not occur.

To compile a source file to an object file only (like compiling to a MODULE in ILE C or ILE RPG), use the **-c** option:

```
gcc -c helloworld.c
```

gcc then creates a file named *helloworld.o* that is used to link with other modules using **gcc** as well:

```
gcc -o runnable module1.o module2.o
```

Here only one of the *moduleX.o* files would contain a function called main(). The other would have a function that is used by the module containing main() or something similar.

For a more detailed explanation about the GNU C compiler, see:
http://www.linuxdoc.org/HOWTO/GCC-HOWTO/

4.1.2 C++

Other compilers exist for a number of other languages, but they are not covered in this section. To compile C++ programs, use **g++** as a command. **g++**, like many C++ compilers, shares much in common with its corresponding C compiler (**gcc**).

4.1.3 Scripts

In Linux, you can use scripts for programming as well. They actually call commands. In a way, they can be compared with CL programs in OS/400, except that Linux scripts are not compiled and therefore, run slower. However, it is a very powerful mechanism to automate tasks.

It is very easy to build scripts. Just create a text file with your favorite editor and set the first line to something like (see 3.1.2, “Editing files using text editors” on page 88, for information on editors):

```
#!/bin/sh
```

The program after the **#!** (in this case **/bin/sh**) is the shell used to execute that script. **sh** is the most used shell to do scripts, but you could also use another shell interpreter such as **bash**, **ksh** or **csh**, and languages like Perl are used in places that call for scripts.

Important: Before a script can be executed, it must have an executable tag set first. Use the **chmod** command for that file, for example:

```
chmod +x scriptname
```

Here, the **+x** simply stands for “set the executable tag to true”. For more information about **chmod**, refer to 3.8.3, “Permission bits” on page 112, and 3.8.4, “Setting permissions” on page 113.

For more detailed information, choose one of the numerous scripting howtos from the site at:
http://www.linuxdoc.org/HOWTO/HOWTO-INDEX/howtos.html

You could use scripts to build an executable out of two modules, named module1.c and module2.c:

```
#!/bin/sh

# This script builds the program named runme composed out of the files module1.c and
# module2.c

# First build the objectfile module1.o
gcc -c module1.c

# Then build the objectfile named module2.o
gcc -c module2.c

# Finally link the two modules together to a program called runme
gcc -o runme module1.o module2.o

./runme        # To test it at once, execute runme
```

Note that on a line the part after "#" is not executed, but is treated as a comment.

Of course, this is not the best way to compile large programs. All modules would be built each time you compile them, even if they weren't changed. As the program would become larger, the script would become unmaintainable.

4.1.4 Building with make

Use **make** and makefiles to build larger programs. Makefiles are actually advanced scripts, used by **make**, with rules that describe the dependencies that exists and how the program should be built. They contain an incredible amount of power to help developers in their job building programs.

You can find very extensive information in the *GNU make manual*, which is available on the Web at: http://www.gnu.org/manual/make/index.html

A makefile named `Makefile` (notice the capital "M") is picked up automatically by **make** when you issue the **make** command.

Rules

A simple makefile consists of rules of the following form:

```
target ... : prerequisites ...
<tab>     command
<tab>     ...
<tab>     ...
```

- **target:** A target is usually the name of a file that is generated by a compiler; examples of targets are executable or object files. A target can also be the name of an action to carry out, such as `clean`.
- **prerequisite:** A prerequisite is a file that is used as input to create the target. A target often depends on several files.
- **rule:** A rule, then, explains how and when to remake certain files that are the targets of the particular rule. **make** carries out the commands on the prerequisites to create or update the target.
- **command:** A command is an action that **make** carries out. A rule may have more than one command, each on its own line.

Note: You need to put a *tab* character at the beginning of every command line. This is an obscurity that catches the unwary.

Usually a command is in a rule with prerequisites and serves to create a target file if any of the prerequisites change. However, the rule that specifies commands for the target does not need to have prerequisites. For example, the rule containing the **delete** command associated with the target “clean” does not have prerequisites.

Here is an example:

```
.SUFFIXES:.c .C .o
.c.o:
   $(CC) $(CFLAGS) -c $<
.C.o:
   $(CCPP) $(CFLAGS) -c $<
all: runme
CC = gcc
CCPP = g++
CFLAGS = -O3
# include any required libraries.  -lm for math is common.
LDFLAGS = -lm
OBJS = module.o module2.o
runme: $(OBJS)
   $(CC) $(CFLAGS) -o stripHTML $(OBJS) $(LDFLAGS)
test:runme
   ./runme
clean:
                                    rm -f *.o runme
```

If you want to make the **runme** program, simply execute the **make runme** command. It then automatically compares the last changed date of all files and concludes whether object files need to be rebuilt. When a file has been changed, the defined rule for that file is executed.

To delete all object files and the created program, run **make clean**.

To test the program, a rule named “test” is defined. If one of the source files has been changed, or the program does not exist yet, it recompiles the program because the rule “test” is depends on **runme**. If no errors occur while compiling and linking, the command **./runme** is executed. The same logic is used while compiling. If a compilation fails, the consecutive commands (compiling other files in this case) are not executed.

make has a lot of options, like creating variables, creating template rules, and including other makefile rules. Dependencies can be checked automatically, and complete directory structures can be compiled by recursive makefiles.

Actually, most makefiles are generated from *makefile.in* files using **automake**. **automake**, **autoconf**, and **libtool** are three packages that most open source software use and that greatly ease making software portable. For more information on this topic, see:
http://www.gnu.org

Tip: When using a multiprocessor system (or in the case of Linux in an iSeries partition assigned more than one processor to the partition), it's best to run **make** using the **-j x** option. A command like **make -j 2 runme** starts two processes, which can be easily split up over two processors. It's even a good idea to start up two processes per processor, so that when one compile process is waiting for disk I/O, the processor can stay working on the other task instead of having to wait for the slow disk I/O (relatively slow compared to the processor speed).

For this to work, however, you must be sure that the makefiles are perfectly clean of dependencies. If not, **make -j** could lead to race conditions.

4.1.5 Debugging programs using gdb

This section provides a short checklist of how to debug programs using **gdb**. For extended information, consult the *Debugging with GDB* manual, which is available on the Web at: http://www.gnu.org/manual/gdb/html_mono/gdb.html

Before debugging, make sure you compile the binary with debugging support. For **gcc**, this is the **-g** option.

Example 4-1 shows you the source code of a program used to explain **gdb**.

Example 4-1 debugtest.c

```
#include <stdio.h>
int main(void)
{
  int varname;
  printf("Beginning of program.\n");
  varname=4;
  if (varname==4) {
    printf("Variable was the original 4.\n");
  }
  else {
    printf("Variable was %i.\n", varname);
  }

  printf("End of program.\n");
}
```

First compile this program with debug information:

```
[root@bauer4 debugtest]# gcc -g -o debugtest debugtest.c
```

If you run it, you get:

```
[root@bauer4 debugtest]# ./debugtest
Beginning of program.
Variable was the original 4.
End of program.
```

These results are quite predictable. However, while running **gdb**, you can change the value of the variable.

Here is an example debug session that shows some of the basic tasks:

```
[root@bauer4 debugtest]# gdb ./debugtest
                                    Start the debugger with the debugtest program.
GNU gdb 5.0
Copyright 2000 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show warranty" for details.
This GDB was configured as "ppc-redhat-linux"...
(gdb) break main                    Add a breakpoint at the first line of the
                                    function main().
Breakpoint 1 at 0x100004fc: file debugtest.c, line 5.
(gdb) run                           Start running the program.
Starting program: /tmp/debugtest/./debugtest

Breakpoint 1, main () at debugtest.c:5
                                    It runs until it encounters the breakpoint.
5         printf("Beginning of program.\n");
                                    The next line that will be executed is shown.

(gdb) next                          Step one line.
Beginning of program.
6         varname=4;
(gdb) print varname                 Print the value of the variable with name varname.
                                    It was not initialized yet, so it has a strange
                                    value.
$1 = 268340084
(gdb) next                          Step one line.
7         if (varname==4) {
(gdb) print varname                 Now it is initialized, and its value is 4.
$2 = 4
(gdb) set variable varname=42       Manually change that value to 42.

(gdb) next                          Execute the next step, which is the if statement
                                    that evaluates to false, and thus the program jumps
                                    to the else statement.

11          printf("Variable was %i.\n", varname);
(gdb) next
Variable was 42.
14        printf("End of program.\n");
(gdb) continue                      Continue to the next breakpoint, but as there is
                                    none, the program runs until the end.
Continuing.
End of program.

Program exited with code 020.
(gdb) quit                          And quit the debugger.
```

There are shortcuts available for most commands like **next** and **continue** (**n**, **c**, **v**, **q**, ...). For more information, type **help** on the command line of the debugger; then online documentation is displayed.

There are X Windows-based debuggers, such as **XXgdb** or **ddd**, which are open source debuggers.

4.2 Open source applications for Linux on iSeries

Of the Linux systems out there, most have Intel or Intel-compatible processors (also called x86 processors). More people can use an Intel machine to develop applications. For this reason, most applications on Linux are primarily compiled for the Intel platform. Because of the open source nature of the Linux community, all source code can be recompiled for another platform. If the code is written properly, there is no problem when recompiling, and no modification of the source code is needed.

However, a lot of the applications use platform-dependent code, for example statements that depend on byte order. (The differences between PowerPC and Intel are covered later.) These applications have to be modified to run on other platforms. We say they are *ported to other platforms*.

To compile code for another platform, the easiest way is simply to put the source on the target platform and compile it on that platform. You could also compile it on another type of platform using a cross-compiler. The power of the Linux platform is that the base functionality is already available on most platforms. This means the kernel and almost all libraries can be used to build on, when creating a platform independent application, and you use the **gcc** compiler (which also works on every platform).

4.2.1 Differences between PowerPC and Intel processors

The biggest difference between PowerPC and Intel processors is, of course, the instruction set and consequently the assembler instruction set. The approach to registers, memory, etc. is totally different. However, hardly any Linux application uses assembly code directly incorporated in the C source programs, so this is not a problem for platform independency.

Another difference is the approach to the system devices, I/O, disk, etc. This is not only a matter of processor architecture, but this depends on the systems architecture as well. Normal programmers, however, do not need to worry about these different approaches to the systems devices. For example the iSeries uses the PowerPC processor, but an iMac also uses it (although there are some differences), and it is clear the approach to devices is quite different (although the iSeries uses a PCI bus as well). The kernel takes care of that, and once that is ported, you can use all Linux facilities you use on any other platform.

It's important to note that Linux is completely ASCII based and no EBCDIC is involved at all. However, EBCDIC is OS/400 specific and is not linked at all with the PowerPC processor, therefore we don't need to worry about that.

There is only one difference most programmers should be concerned with and that's *byte ordering*.

Byte ordering ('endianness')

The iSeries server works with PowerPC processors. The main difference between PowerPC and x86 is the byte ordering. A 32-bit binary integer is always represented as four bytes in the memory of a computer. But, what part of that integer does the first byte get? The representation of the memory of a PowerPC processor is the way you would see it in a C source program. On PowerPC, the line of code `q=0x0A0B0C0D`; is stored with the `0A` byte at the first memory byte of q, `0B` at the second, and so on. This is called *big endian*, because the high order byte ("big") is at the lowest memory offset.

However, on an x86 processor, the representation is the other way around. It is called *little endian*, because the low order ("little") part is stored at the first offset of q for the same statement `q=0x0A0B0C0D`. Therefore, in little endian, the `0D` is stored at the first byte position, `0C` at the second, and so on. There is a long history behind these differences. What is important is that programmers understand the situation and code, in a Linux environment, in a way to avoid depending on the differences so code can easily port to both CPUs. Please refer to Table 4-1.

Table 4-1 Byte ordering (endian)

Decimal value, C source code value	Big endian representation	Little endian representation
125 0x7D	7D	7D
500 0x1F4	01 F4	F4 01
200,000,000 0xBEBC200	0B EB C2 00	00 C2 EB 0B
8,589,934,594 0x0000001000000002	00 00 00 01 00 00 00 02	02 00 00 00 01 00 00 00

The most straightforward thing to do to avoid byte ordering or endian problems is to minimize endian dependencies as listed here in case of C and C++ programs:

- Avoid casting smaller sized storage over larger sized ones (for example, no `(short *)` over `(int *)`).
- Avoid casting larger sized storage over smaller sized ones (for example, no `(int *)` over `(short *)`).
- Avoid casting one structure over another.
- Avoid unions with different-sized items (this is almost the same as saying "avoid unions").
- Avoid the use of bit variables.

If these coding styles cannot be avoided, the use of a suitable `#ifdef` should be employed and big and endian versions of the code provided. This lets the compiler select the right byte-dependent code sequence for whichever byte order the current complier is targeting.

4.2.2 RedHat Package Manager (RPM)

The RedHat Package Manager (RPM) is used to keep track of all installed programs on a system. Therefore it has a database that tells which files are owned by which packages, so that upgrades or uninstallations can be done flawlessly. It also keeps track of dependencies between packages and dependencies of specific versions of packages. It is now widely used among other distributors as well. For more information, refer to:
`http://www.redhat.com/support/manuals/RHL-7-Manual/ref-guide/index.html`

You can also refer to the documentation that came with your Linux distribution.

Installing packages

Normally, when you installing a program, try to install the RPM package. You can easily do this by running (the **`-i`** is equivalent to specifying **`--install`**; most people use the short options):

```
rpm -ihv wu-ftpd-2.6.1-6.ppc.rpm
```

Apart from **-i**, we use two other switches here for our convenience:

-h Prints hash marks to show the progress
-v Makes RPM a bit more verbose. (Without **-v**, RPM gives us no visible feedback, except an error occurs.)

Architectures

Note the `ppc` in the filename; it means this package is compiled for the PowerPC platform. Trying to install a package of the wrong architecture (for example, an i386 package on a PPC machine) usually results in some misleading error message about unsatisfied dependencies with a library. Packages that do not contain binary data are not associated with an architecture, and their name contains *noarch*. The **--ignorearch** switch overrides the architecture to force an installation (this might happen with ppc/ppc64 packages).

Tip: You may wonder how the Linux folks manage to type commands with extremely long filenames like **rpm -Uhv /usr/src/packages/RPMS/i386/screen-3.9.9-9.i386.rpm** all day. Very easy – with their tricks. Here are some to help you get started:

- **The Tab (tabulator) key:** It tries a completion. Type the first letters of a command or a filename, and as soon as the word is unambiguous, it can be completed by pressing the Tab key. As long as what you typed is still ambiguous, pressing the Tab key once more will at least show you all matches.

  This works for filenames and for commands alike. Enter **ps** and press the Tab key twice, and you will see all commands that start with "ps".

  Especially when typing a long pathname like `/usr/share/doc/packages/screen`, it is useful to use the Tab key completion in between. Then you notice a typo once it occurs, and not only when you finally press Enter. And, it is very fast!

- **yank-last-argument:** This is a shell (bash) feature that repeats the last word from the last command you typed. It is bound to the key combination "Alt-.". This means you press the "." key while holding down the Alt key. Note the following from the bash man page:

  ```
  Insert the last argument to the previous command (the last word of the previous
  history entry). [...] Successive calls to yank-last-arg move back through the
  history list, inserting the last argument of each line in turn.
  ```

  Consider this example. You typed **l lengthyfilename**. Next, you need the name again because you want to install the package, and enter **rpm -ihv**. Then, instead of entering the *lengthyfilename* again, just press "Alt-."

  On some deficient terminals, combined keys with Alt do not work. In that case, you can get by with the sequence Esc <key> instead of Alt-<key>, for example "Esc .".

- **Copy and paste:** Use your mouse and simply click and then right-click. And that's all!

- **Command-line history:** Use the cursor up/down keys to access previous commands from the history.

Uninstalling packages

To uninstall a package, you do the inverse:

```
rpm -e wu-ftpd
```

-e is the corresponding short option to **--uninstall** (to memorize it, think of "erase"). Here *wu-ftpd* is the name of the package, which normally corresponds to the first characters of the *filenameXXXXX.ppc.rpm* file, without the version number.

Updating packages

You cannot install a package twice. To install a newer version, you could uninstall the old version first. However, most of the time, it is better to use the dedicated update mechanism of RPM, because the handling of certain files (for example, configuration files) can be different. The usage is very similar, only you use **-U** instead of **-i**, for example:

```
rpm -Uhv wu-ftpd-2.6.1-6.ppc.rpm
```

When installing or updating packages, there are a few interesting things to know:

- **-i** and **-U** also take multiple filenames on one command line or even **rpm -Uhv *.rpm**
- To avoid the extra step of downloading a package before installing it, you can hand over the URL to RPM, like **rpm -Uhv ftp://ftp.suse.com/path/filename.rpm** . You can also use **http://**.
- **-U** does not care if a package is already installed. If it is already installed, it will do an update; if it is not installed, it will install it. This has two consequences:
  - Most people just use **-U** as a habit, and never **-i**.
  - Imagine the situation where you get an entire directory of new packages, and some of them are installed in your system (in older versions), while many of them are not. If you use **rpm -Uhv *.rpm**, all packages would be installed, but you only want to update the ones that are already installed. In this situation, you would use **-F** (freshen) instead, like this **rpm -Fhv *.rpm**.

Querying RPM

To see if a package is installed, use **-q**, which corresponds to **--query**:

```
[root@bauer5 /]# rpm -q wu-ftpd
wu-ftpd-2.6.1-6
```

This outputs a line that indicates whether the package is installed. If it is not installed, it shows something like:

```
[root@bauer5 /]# rpm -q something
package something is not installed
```

Attention: If you mistype the name, no corresponding package will be found. For instance, querying for `wuftpd` would not yield anything because the dash is missing (provided it is called `wu-ftpd` as in our example). But often we do not know the exact name! Therefore, a common approach is to query for all packages (**-qa**) and filter the output through **grep** in order to search for a part of the name that we do know. This helps to find the name, for example:

```
[root@bauer5 /]# rpm -qa | grep ftp
ftpdir-2001.5.29-1
lukemftp-1.5-46
wu-ftpd-2.6.1-6
vsftpd-0.9.3-0
```

Show details

Each package comes with a description that can be shown with **-qi**. The list of contained files is displayed with the **-ql** switch, and **-v** can be added (**-qlv**) to obtains more detail. To see only the documentation files, you use **-qd**.

To query a package that is *not* installed but is available as a file in the file system, you would add the **-p** switch (for example, **rpm -qlp packagename**).

Tip: When querying for packages, RPM displays the names in the form name–version–releasenumber, such as `vsftpd-2.6.1-6`. A frequent catch is to use this full name in an RPM command, like **`rpm -qi vsftpd-2.6.1-6`**. In this case, RPM will tell you it can't find the package, because you are supposed to give the package name only, not the version number. The actual package name in this example would be `vsftpd`.

Where does file X come from?

Another useful query option is **`-qf`**. It is used to find out to which package a given file belongs. The filename has to be specified with its full pathname, for example:

```
ROOT@bro ~/ # rpm -qf /bin/df
fileutils-4.0.35-42
```

What did they do to the package?

To see the change log of a package (that is the history of changes applied to the package by the package maintainer), use **`-q --changelog`**.

Package integrity

The **`-V`** switch checks the integrity of all files of a given installed package. As a matter of fact, RPM keeps detailed information about each file in its database, like file size, access times, MD5 checksum, permissions, and ownership. RPM -V reports on each file that has a changed status.

Miscellaneous features RPM

There are some other frequently used features of RPM. In various situations, you might want to override RPM's behavior (sometimes you just know it better than RPM.). For example, you can force installation with **`--force`**, although files of another package would be overwritten, or when the same package, or a newer version, is already installed. The dependency check can be disabled via **`--nodeps`**. To check if something *would* work, use **`--test`**.

Other sources

The RPM man page lists and explains all the options. A short summary can be displayed with **`rpm --help`**.

RPM is one of those tools that you need everyday in a Linux environment, just like text editors we discussed in 3.1.2, "Editing files using text editors" on page 88. Therefore, we urge you to read the RPM chapter in your distributor's documentation and look for detailed information at: http://www.rpm.org/

For daily RPM usage, you won't need the next one. But if you are going to build packages, then the maximum RPM book is a most valuable resource. It is available online at: http://www.redhat.com/docs/books/max-rpm/index.html

4.2.3 Building source RPM files (src.rpm)

You need to install the created package afterward.

When a PowerPC version of an RPM is not found, the next step is to look for a source RPM file. You can recognize one by the *.src.rpm* suffix or a similar convention with "src" or "source" in the filename. To build it, you simply run the command **`rpm --rebuild`**, as shown in Example 4-2. You don't need to do this as root, although we recommend you do because sometimes to build packages, you need special tools and installing packages needs write access to system directories. We left out some lines there because the general process is more important than the details here.

Example 4-2 Installing a source RPM file

```
[root@bauer4 klaas]# rpm --rebuild wu-ftpd-2.6.1-6.src.rpm
Installing wu-ftpd-2.6.1-6.src.rpm
Executing(%prep): /bin/sh -e /var/tmp/rpm-tmp.90079
+ umask 022
+ cd /usr/src/redhat/BUILD
+ cd /usr/src/redhat/BUILD
+ rm -rf wu-ftpd-2.6.1
+ /bin/gzip -dc /usr/src/redhat/SOURCES/wu-ftpd-2.6.1.tar.gz
+ tar -xf -
...

...
+ ./configure ppc-redhat-linux --prefix=/usr --exec-prefix=/usr --bindir=/usr/bin
--sbindir=/usr/sbin --sysconfdir=/etc --datadir=/usr/share --includedir=/usr/include
--libdir=/usr/lib --libexecdir=/usr/libexec --localstatedir=/var --sharedstatedir=/usr/com
--mandir=/usr/share/man --infodir=/usr/share/info --enable-quota --enable-pam
--disable-rfc931 --enable-ratios --enable-passwd
creating cache ./config.cache
checking host system type... powerpc-redhat-linux-gnu
checking target system type... powerpc-redhat-linux-gnu
checking build system type... powerpc-redhat-linux-gnu
checking for gcc... gcc
...

...
+ make
(cd support ; make all)
make[1]: Entering directory `/usr/src/redhat/BUILD/wu-ftpd-2.6.1/support'
gcc -O2 -fsigned-char -c -o authuser.o authuser.c
rm -f libsupport.a
ar cq libsupport.a authuser.o
ranlib libsupport.a
make[1]: Leaving directory `/usr/src/redhat/BUILD/wu-ftpd-2.6.1/support'
(cd util/privatepw ; make all)
make[1]: Entering directory `/usr/src/redhat/BUILD/wu-ftpd-2.6.1/util/privatepw'
...

...
Executing(%install): /bin/sh -e /var/tmp/rpm-tmp.93041
+ umask 022
+ cd /usr/src/redhat/BUILD
+ cd wu-ftpd-2.6.1
+ rm -rf /var/tmp/wu-ftpd-root
+ mkdir -p /var/tmp/wu-ftpd-root/etc /var/tmp/wu-ftpd-root/usr/sbin
+ make install DESTDIR=/var/tmp/wu-ftpd-root
(cd support ; make all)
...

...
+ rm -rf wu-ftpd-2.6.1
+ exit 0
```

The **rpm** command untars the contents of the src.rpm file and finds out the kind of system on which it is running. Then it configures the makefiles to deal with this environment, makes the binaries, does the installation, and afterwards, deletes all the working files. Some other things happen, such as setting up configuration files, setting permissions, etc.

The advantage of this approach is that it basically gives the same result as a regular RPM package, which means you can upgrade it, verify the installation, safely remove it from the system, etc. This is very useful if a binary PowerPC package for what you need could not be found!

Installing the built package

After rebuilding the source RPM, a compiled .ppc.rpm file was created in `/usr/src/packages/RPMS/ppc/wu-ftpd-2.6.1.ppc.rpm`. This package must be installed afterward by running:

```
rpm -iuhv /usr/src/packages/RPMS/ppc/wu-ftpd-2.6.1.ppc.rpm
```

4.2.4 Compiling an open source program yourself

When you do not find an src.rpm file or rpm file, you need to compile the program yourself and manually do all steps the **rpm** command would have done.

Important: Although the main procedure of compiling programs follows the same basic steps, you should always read the specialized compilation instructions for each program you try to compile. Usually these files are called `INSTALL` or `README` (both are case sensitive).

The steps in the following sections explain how to compile a version of wu-ftpd and how to modify one line in the source code (which you might do for yourself as well, for various reasons such as security issues).

As an example, we modify the source code of the program and then compile it. You can skip the first part of this section in normal situations where you just build the program.

Goals

When you connect to a standard wu-ftp server, it sends you the following lines (or something similar):

```
C:\>ftp bauer4
Connected to bauer4.rchland.ibm.com.
220 bauer4.rchland.ibm.com FTP server (Version wu-2.6.1(1) Thu Jan 25 15:17:37 E
ST 2001) ready.
User (bauer4.rchland.ibm.com:(none)):
```

This might be a security issue because computer criminals can see which exact version of wu-ftpd is running and perhaps know an exploit for that version of the FTP server. To fix this, the FTP daemon source code can be modified and recompiled as described here:

1. Download the sources of the program, which are almost always packed in the .tar.gz format. In case of wu-ftpd, grab them from `http://www.wu-ftpd.org/` or one of its mirror sites. Untar the file:

   ```
   [root@bauer4 /usr/local/src]# tar xzf wu-ftpd-2.6.1.tar.gz
   [root@bauer4 /usr/local/src]# cd wu-ftpd-2.6.1
   [root@bauer4 wu-ftpd-2.6.1]# ls
   CHANGES       INSTALL      README.AUTOCONF  config.h.noac  doc          support
   CONTRIBUTORS  LICENSE      build            config.sub     install-sh   util
   COPYRIGHT     Makefile.in  config.guess     configure      makefiles
   ERRATA        README       config.h.in      configure.in   src
   ```

2. Look for the file that contains the message, by looking in all files in the `src/` directory and see if they contain the string "`FTP server`":

   ```
   [root@bauer4 wu-ftpd-2.6.1]# grep -r "FTP server" .
   src/ftpd.c:/* FTP server. */
   src/ftpd.c:     reply(500, "%s FTP server shut down -- please try again later.",
   src/ftpd.c:         reply(220, "%s FTP server (%s) ready.", hostname, version);
   ```

```
src/ftpd.c:         reply(220, "%s FTP server ready.", hostname);
src/ftpd.c:         reply(220, "FTP server ready.");
src/ftpd.c:    lreply(211, "%s FTP server status:", hostname);
src/ftpd.c:     syslog(LOG_WARNING, "FTP server started without ftpaccess file");
src/ftpd.c:    syslog(LOG_INFO, "FTP server (%s) ready.", version);
```

3. It is clear the file you need to modify is `src/ftpd.c`. Modify the file using:

   ```
   reply(220, "%s FTP server (%s) ready.", hostname, version);
   ```

   Change this line to the following line so that it only give the message “FTP server ready”:

   ```
   reply(220, "FTP server ready.");
   ```

   You can see in the source code that there probably is another way to configure this *greeting* message by a configuration parameter, but this is not covered in this book. (For more information, consult the wu-ftpd documentation, and look for the greeting parameter.)

4. Compile the program. Look in the `README` or `INSTALL` file for details on building this. Most open source programs use the same approach. Start by running the **`configure`** script in the main directory. You could give parameters to configure, for example, using **`configure --prefix=directoryname`** so it is installed in another directory.

   ```
   [root@bauer4 wu-ftpd-2.6.1]# configure
   creating cache ./config.cache
   checking host system type... powerpc-unknown-linux-gnu
   checking target system type... powerpc-unknown-linux-gnu
   checking build system type... powerpc-unknown-linux-gnu
   checking for gcc... gcc
   checking whether the C compiler (gcc  ) works... yes
   checking whether the C compiler (gcc  ) is a cross-compiler... no
   checking whether we are using GNU C... yes
   checking whether gcc accepts -g... yes
   checking for POSIXized ISC... no
   checking for ranlib... ranlib
   checking for bison... bison -y
   checking for a BSD compatible install... /usr/bin/install -c
   checking for ar... ar
   checking how to run the C preprocessor... gcc -E
   checking for ANSI C header files... yes
   checking whether time.h and sys/time.h may both be included... yes
   checking if we can do hard links... yes - using ln
   checking for dirent.h... yes
   ...

   ...
   updating cache ./config.cache
   creating ./config.status
   creating Makefile
   creating src/Makefile
   creating support/Makefile
   creating util/privatepw/Makefile
   creating src/pathnames.h
   creating config.h
   creating src/config.h
   ```

   As mentioned, the configure script checks out the environment for minor differences related to the version of system libraries, the version of the compiler, etc. (actually that’s autoconf, that we mentioned in 4.1.4, “Building with make” on page 140). In the end, the makefiles are created.

5. Proceed by running **`make`**. This compiles all C sources and creates the binaries. In this case, they are placed in the bin/ directory.

```
[root@bauer4 wu-ftpd-2.6.1]# make
(cd support ; make all)
make[1]: Entering directory `/tmp/wu-ftpd-2.6.1/support'
gcc -g -O2 -c -o authuser.o authuser.c
rm -f libsupport.a
ar cq libsupport.a authuser.o
ranlib libsupport.a
make[1]: Leaving directory `/tmp/wu-ftpd-2.6.1/support'
(cd util/privatepw ; make all)
make[1]: Entering directory `/tmp/wu-ftpd-2.6.1/util/privatepw'
(sh ../../src/newvers.sh)
gcc  -g -O2 -L../../support -lcrypt -lnsl -lresolv -lsupport -o privatepw privatepw.c
../../src/COPYRIGHT.c vers.c
/tmp/cc6ihqRR.o: In function `main':
/tmp/wu-ftpd-2.6.1/util/privatepw/privatepw.c:297: the use of `tmpnam' is dangerous,
better use `mkstemp'
make[1]: Leaving directory `/tmp/wu-ftpd-2.6.1/util/privatepw'
(cd src ; make all)
make[1]: Entering directory `/tmp/wu-ftpd-2.6.1/src'
gcc  -g -O2 -I.. -I../support -c -o COPYRIGHT.o COPYRIGHT.c
bison -y ftpcmd.y
mv -f y.tab.c ftpcmd.c
sh newvers.sh
gcc  -g -O2 -I.. -I../support -c -o vers.o vers.c
gcc  -g -O2 -I.. -I../support -c -o ftpd.o ftpd.c
gcc  -g -O2 -I.. -I../support -c -o ftpcmd.o ftpcmd.c
...
gcc  -g -O2 -o ftpd COPYRIGHT.o vers.o ftpd.o ftpcmd.o glob.o logwtmp.o popen.o access.o
extensions.o realpath.o acl.o private.o authenticate.o conversions.o rdservers.o paths.o
hostacc.o sigfix.o auth.o routevector.o restrict.o domain.o wu_fnmatch.o timeout.o
getpwnam.o -L../support -lcrypt -lnsl -lresolv -lsupport
gcc  -g -O2 -I.. -I../support -c -o ftpcount.o ftpcount.c
gcc  -g -O2 -o ftpcount COPYRIGHT.o vers.o ftpcount.o -L../support -lcrypt -lnsl
-lresolv -lsupport
...
make[1]: Leaving directory `/tmp/wu-ftpd-2.6.1/src'
if [ ! -d bin ]; then mkdir bin; fi
rm -f bin/ftpd bin/ftpshut bin/ftpcount bin/ftpwho bin/ckconfig bin/ftprestart
ln src/ftpd bin
ln src/ftpshut bin
ln src/ftpcount bin
ln src/ftpcount bin/ftpwho
ln src/ckconfig bin
ln src/ftprestart bin
size bin/ftpd bin/ftpshut bin/ftpcount bin/ftpwho bin/ckconfig bin/ftprestart
   text    data     bss     dec     hex filename
 158828   10236   96644  265708   40dec bin/ftpd
   9542     340   10284   20166    4ec6 bin/ftpshut
   7755     352      36    8143    1fcf bin/ftpcount
   7755     352      36    8143    1fcf bin/ftpwho
   6447     320    8228   14995    3a93 bin/ckconfig
   6997     324   12320   19641    4cb9 bin/ftprestart
```

6. To install the binaries, simply run **make install**. This command copies all files to their appropriate directories. Use care here, because the location of the programs can depend on the distributor, and standard make install scripts might not copy the programs into the right directory. (In case the packages of those programs were already installed, first uninstall them with **rpm -e** to avoid conflicts. In general, it's much better to install source RPM packages to maintain consistency in the RPM database on your system!)

```
[root@bauer4 wu-ftpd-2.6.1]# make install
```

```
(cd support ; make all)
...
/usr/bin/install -c -o bin -g bin -m 755 bin/ftpcount //usr/bin/ftpcount
/usr/bin/install -c -o bin -g bin -m 755 bin/ftpwho //usr/bin/ftpwho
/usr/bin/install -c -o bin -g bin -m 755 bin/ckconfig //usr/sbin/ckconfig
/usr/bin/install -c -o bin -g bin -m 755 bin/ftprestart //usr/sbin/ftprestart
/usr/bin/install -c -o bin -g bin -m 755 util/privatepw/privatepw //usr/sbin/privatepw
installing manpages.
/usr/bin/install -c -m 644 -o bin -g bin -m 644 doc/ftpd.8 //usr/man/man8/ftpd.8
/usr/bin/install -c -m 644 -o bin -g bin -m 644 doc/ftpcount.1 //usr/man/man1/ftpcount.1
...
```

7. In case **inetd** or **xinetd** is used, you don't need to restart it because ftpd is reloaded each time again. In case ftpd is running standalone, restart it like this:

```
[root@bauer4 wu-ftpd-2.6.1]# killall wu-ftpd
[root@bauer4 wu-ftpd-2.6.1]# wu-ftpd
```

8. When you connect to the FTP server now, it shows you the following message:

```
C:\>ftp bauer4
Connected to bauer4.rchland.ibm.com.
220 FTP server ready.
User (bauer4.rchland.ibm.com:(none)):
```

   This doesn't say anything about the server version anymore.

4.2.5 Differences among C library revisions

Recently, the GNU Project released **glibc2.2**, which includes a very important enhancement of internationalization to the C library.

Checking the installed C library version

There are two ways to check version of the C library currently installed.

The simplest way is to see the output of the **ls /lib/libc*** command. A pattern **libc*** matches any file name starting with **libc**:

```
$ ls /lib/libc*
/lib/libc-2.2.1.so  /lib/libcom_err.so.2    /lib/libcrypt-2.2.so
/lib/libc-2.2.so    /lib/libcom_err.so.2.0  /lib/libcrypt.so.1
/lib/libc.so.6      /lib/libcrypt-2.2.1.so
```

The other way is to query the information of the RPM package. Use the **rpm -qf** command to query the package name that owns the C library (**/lib/libc.so.6**). Then see the version part of the package name you get:

```
$ rpm -qf /lib/libc.so.6
glibc-2.2-12
glibc-2.2.1-3.as400
glibc-2.2.1-3.as400
```

Or to see more detailed information, just start the library (which is also an executable):

```
iserieslinux:~ # /lib/libc.so.6
GNU C Library stable release version 2.2, by Roland McGrath et al.
Copyright (C) 1992,93,94,95,96,97,98,99,2000 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.
Compiled by GNU CC version 2.95.2 19991024 (release).
Compiled on a Linux 2.4.2 system on 2001-05-18.
Available extensions:
```

```
GNU libio by Per Bothner
crypt add-on version 2.1 by Michael Glad and others
Berkeley DB glibc 2.1 compat library by Thorsten Kukuk
linuxthreads-0.9 by Xavier Leroy
BIND-8.2.3-T5B
NIS(YP)/NIS+ NSS modules 0.19 by Thorsten Kukuk
NSS V1 modules 2.1 by Thorsten Kukuk
libthread_db work sponsored by Alpha Processor Inc
software FPU emulation by Richard Henderson, Jakub Jelinek and others
```

Difference between version 2.2.x and 2.1.x

There are some important differences between 2.2.x series of C libraries and 2.1.x ones. It is crucial when you develop or build applications that you use internationalized functions.

In the versions of 2.1.x series, wide-character based input and output functions, such as **wprintf()**, **wscanf()**, and **fgetws()**, are not provided. This means applications using these functions cannot be compiled. In addition, some locale-sensitive functions, such as **localeconv()**, **nl_langinfo()**, **strftime()**, and **strfmon()**, have not fully implemented their functions defined in ISO C standards.

As for the versions of 2.2 or later, these problems are solved completely. You should remember these facts to build application software from their source codes.

4.3 Building the kernel

To most non-Linux users, a statement such as, “Oh, when this doesn’t work, just rebuild your kernel,” sounds really frightening. However, if you make sure you never overwrite your own working copy of the kernel, nothing really bad can happen. The worst thing to happen would be that the kernel is not working, and thus the Linux system is not booting. In this case, just return to your older kernel (by changing the IPL source as you see later in this chapter) and try to configure and compile your kernel again.

This section discusses the basic procedure for configuring and compiling your kernel. It tries to focus as much as possible on the iSeries specific part of this procedure. To find out more exact information about the generic process of kernel compilation, check out the kernel howto at: http://www.linuxdoc.org/HOWTO/Kernel-HOWTO.html

4.3.1 Acquiring the kernel

You may download the kernel from http://www.kernel.org or one of its mirror sites. To “install” this kernel source, untar it to /usr/src, which is the standard place for the kernel sources:

```
[root@bauer4 /]# cd /tmp/download
[root@bauer4 download]# ls
linux-2.4.2.tar.gz  patch-lpar-linux-checkpoint-03-01-2001
[root@bauer4 download]# cd /usr/src
[root@bauer4 src]# ls
linux  lpar-linux  redhat
[root@bauer4 src]# mkdir 2.4.2
[root@bauer4 src]# cd 2.4.2/
[root@bauer4 2.4.2]# tar xzf /tmp/download/linux-2.4.2.tar.gz
[root@bauer4 2.4.2]# ls
linux
```

In most distributions, there even is an RPM package that corresponds to the installed version of that distribution. It's usually called *kernel-source* and can be easily installed using **rpm --install kernel-source-2.4.2.rpm** or something similar.

4.3.2 Acquiring the iSeries kernel patch

At the time this redbook was written, the changes that IBM made are not part of the standard kernel. They are packaged as a patch to the standard kernel. In the future, it is possible that you can skip this section, simply because it is integrated into the kernel. For the time being, download the patch from:
http://oss.software.ibm.com/developerworks/opensource/linux/projects/ppc/iSeries_notes.php

Unzip the downloaded patch, and uncompress it:

```
[root@bauer4 /]# cd /tmp/download
[root@bauer4 download]# ls
linux-2.4.3.tar.gz patch-lpar-linux-2.4.3-09-26-2001.gz
[root@bauer4 download]# gunzip patch-lpar-linux-2.4.3-09-26-2001.gz
```

Then apply the patch using the **patch** command:

```
[root@bauer4 linux]# cd /usr/src/2.4.2/
[root@bauer4 2.4.3]# patch -p0 < /tmp/download/patch-lpar-linux-2.4.3-09-26-2001.gz
patching file linux/arch/ppc/Makefile
patching file linux/arch/ppc/config.in
patching file linux/arch/ppc/configs/iSeries_defconfig
patching file linux/arch/ppc/iSeries/HvCall.c
patching file linux/arch/ppc/iSeries/HvLpConfig.c
...
patching file linux/include/linux/vethdevice.h
patching file linux/kernel/timer.c
patching file linux/mm/vmalloc.c
patching file linux/net/netsyms.c
```

It then outputs a complete list of all files that have been patched. If something went wrong, it would have printed that as well. However, a simple way to check whether the patch was successful is by searching for names ending with ".rej", using the **find** command:

```
[root@bauer4 2.4.2]# find linux -name "*.rej"
[root@bauer4 2.4.2]#
```

If it outputs nothing, no files with that name were found, and you have successfully extended the Linux kernel with iSeries-specific device drivers (such as the PowerPC processor, virtual device support, etc.). If something went wrong, make sure you used the latest path with the correct kernel version and try again.

This symptom should not occur when correctly following the guidelines, and you don't have to try compiling when **patch** finds errors, because that would end in a failure anyway.

4.3.3 Configuring the kernel

The general procedure to compile the kernel is started by selecting the functions you want in your kernel. You can actually choose how to compile in each function: as a module (dynamically linked after booting) or statically linked.

Compiling functions as modules provides the most functionality because an already compiled version of the kernel can be fine-tuned by inserting the appropriate modules (check the man pages of **depmod**, **modprobe**, and **insmod** for instructions on how to load modules at run-time). This is the approach most distributors use because it clearly provides the most functionality.

When you want to optimize performance, you should opt for the second approach where you link all the functions statically. Every function that is needed for the kernel to boot should be statically linked in the kernel as well, because, as noted above, modules can only be inserted after booting. If your system is located on a virtual DASD, for example, you should compile the virtual DASD support statically linked into the kernel. This last example is really easy to understand; modules are located on the virtual disk. It is quite clear that before you can access this disk to load modules, the virtual DASD function must be loaded already.

There are three ways to configure your kernel. They all change the .config file located in the main directory (this directory is called “linux”). Actually, there is a fourth way, by manually modifying the `.config` file and then running **make oldconfig**.

make config

If you type **make config** in the main directory, you receive a *long* list of questions, for example:

```
[root@bauer4 linux]# make config
rm -f include/asm
( cd include ; ln -sf asm-ppc asm)
/bin/sh scripts/Configure arch/ppc/config.in
#
# Using defaults found in arch/ppc/defconfig
#
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL) [Y/n/?]
*
* Loadable module support
*
Enable loadable module support (CONFIG_MODULES) [Y/n/?]
  Set version information on all module symbols (CONFIG_MODVERSIONS) [Y/n/?]
  Kernel module loader (CONFIG_KMOD) [Y/n/?]
*
* Platform support
*
Processor Type (6xx/7xx/740O/826O, 4xx, POWER3, POWER4, 8xx, ISERIES_LPAR)
[6xx/7xx/740O/826O]
  defined CONFIG_6xx
MPC826O CPM Support (CONFIG_826O) [N/y/?]
Machine Type (PowerMac/PReP/MTX/CHRP, APUS) [PowerMac/PReP/MTX/CHRP]
  defined CONFIG_ALL_PPC
Workarounds for PPC6O1 bugs (CONFIG_PPC6O1_SYNC_FIX) [Y/n/?]
...
```

This is, of course, the most difficult way to configure the kernel, but also the most fail proof, because it is command-line based. If something goes wrong with the connection or terminal emulation in your session, this might be the only option you have.

make menuconfig

The second approach is to run **make menuconfig**. As you may have guessed, this is a menu-based configuration tool, text oriented. When you type **make menuconfig** in the main directory, you see a screen similar to the example in Figure 4-1.

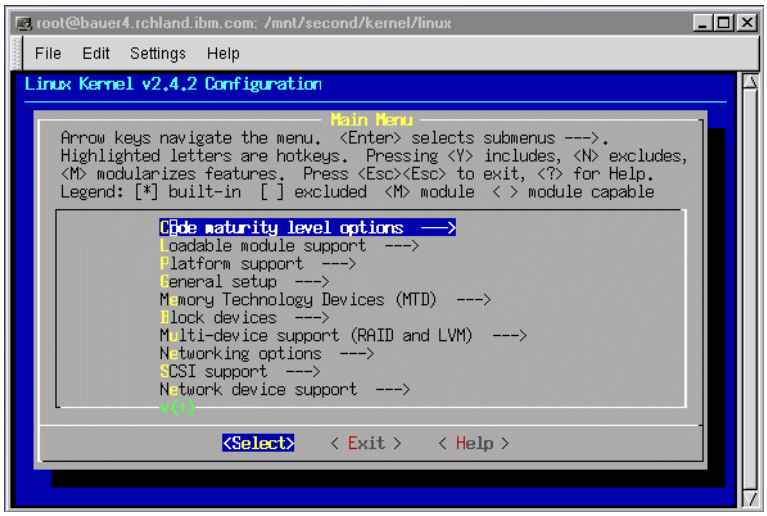


Figure 4-1 Configuring the kernel using make menuconfig

For example, if you configure the Platform support section, you see the windows shown in Figure 4-2. Functions are compiled as modules, in case you choose **M** to select them instead of * or a simple **Y**.

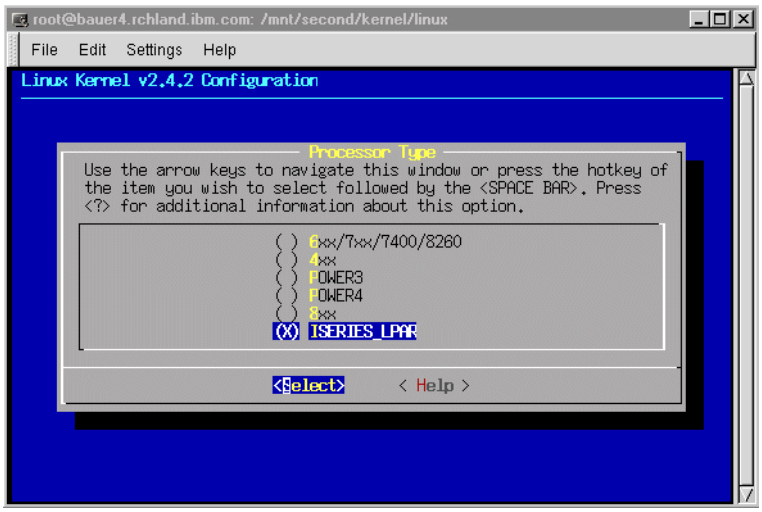


Figure 4-2 Configuring Platform support using make menuconfig

make xconfig

The third way is also the most user-friendly way of configuring the kernel. That is using a real graphical user interface, although it's not the most stable configuration tool right now. If you type **`make xconfig`**, you see the display in Figure 4-3. Make sure you set up the correct environment to use X Windows. It is best to launch the **`make xconfig`** command out of an xterm equivalent terminal. See Chapter 5, "X Windows and OpenOffice" on page 165, for more information about how to work with X Windows.

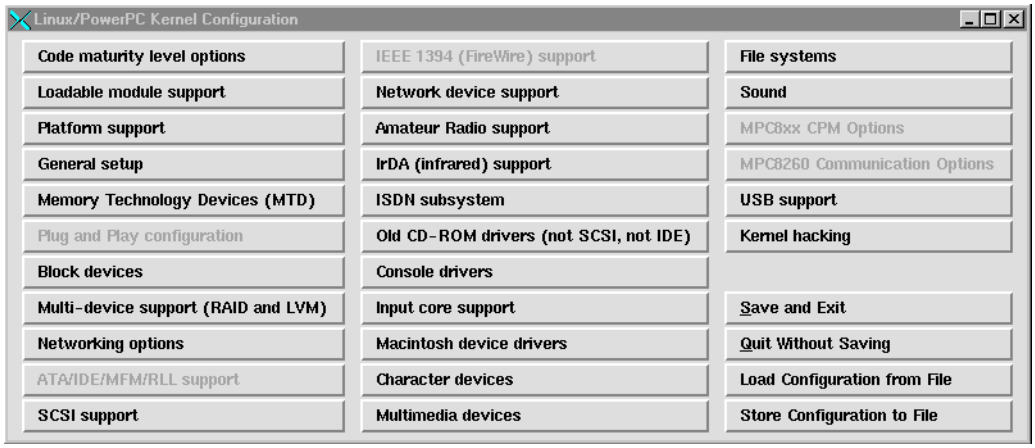


Figure 4-3 Configuring the kernel using make xconfig

When you do the same as above, and go to Platform support, you see the display in Figure 4-4.

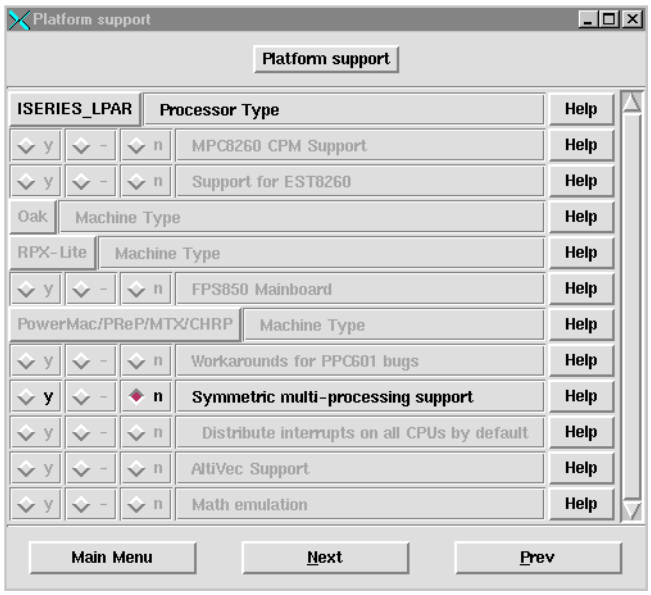


Figure 4-4 Configuring Platform support using make xconfig

4.3.4 Must-have modules for Linux on iSeries

Apart from many optional modules, there are some settings you *really need* for the iSeries Linux kernel.

Note: We use the term “modules” for general packages or specific functions, regardless of whether they are compiled as a module. It is probably best to compile all functions you definitely need as statically linked functions.

First, you must set the platform to ISERIES_LPAR. This is located in **Platform support -> Processor Type -> ISERIES_LPAR**. This setting enables other options further down the configuration that are iSeries specific.

Modules specific to iSeries

We recommend that you minimally enable the following modules for virtual I/O support:

- **Block devices -> iSeries Virtual I/O DASD disk support** to enable virtual disks. See “Virtual disk” on page 93.
- **Character Devices -> iSeries Virtual Console support** to enable the virtual console. See “Virtual console” on page 92.
- **Character Devices -> iSeries Virtual Tape support** to enable the virtual tape drive. See “Virtual tape” on page 94.
- **Network Device -> iSeries Virtual Ethernet driver support** to enable the virtual Ethernet adapter. See “Virtual Ethernet” on page 93.
- If you want devfs support, **File systems -> /dev file system support** and **File systems -> /dev file system support -> Automatically mount at boot time** to enable the devfs kernel module. See 3.2, “Devices” on page 90. When changing from devfs to non-devfs or vice versa, several changes are needed, for example in /etc/fstab. We recommend that you stick to the approach that your distributor installed.

4.3.5 Direct attached LAN adapters

To enable direct attached network adapters, IBM made patches to existing kernel modules. Four devices are supported, and they each have a corresponding Linux module, as illustrated in Table 4-2. For more information on directly attached LAN, refer to 2.5.1, “Virtual LAN configuration” on page 52.

Table 4-2 Directly attached LAN adapters and their kernel modules

Feature code	Description	Kernel configuration menu location	Module name
2744	Token Ring 4/16/100 Mb	Network device -> Token Ring -> IBM Olympic chipset PCI adapter support	olympic.o
2838	Ethernet 10/100 Mb UTP	Network device -> Ethernet (10 or 100 Mbit) -> EISA, VLB, PCI and on board controllers -> AMD PCnet32 PCI support	pcnet32.o
2743	Fiber optical 1 Gb	Network device -> Ethernet (1000 Mbit) -> Alteon AceNIC/3Com 3C986/Netgear GA620 Gigabit support	acenic.o
2760	Ethernet 10/100/1000 Mb UTP		

4.3.6 Direct attached I/O adapters (ibmsis)

Note: Skip this section when no direct attached I/O is needed, because it would only make things more difficult.

In case you need direct attached I/O, there is one IBM proprietary module that enables support for the IOAs in the iSeries. It supports the IBM iSeries Storage Adapters with the following feature codes:

- 2763
- 2748
- 2778

All directly attached disk, tape, and CD devices are detected by this driver as if they were normal SCSI devices. Once this driver is installed, all directly attached devices are detected as standard SCSI peripherals and all generic tools can be used on them. The name of this module is ibmsis.o. See Table 4-3 for a list of needed modules for direct attached SCSI. For a more general high level explanation on directly attached I/O, see 1.4.6, “Native I/O: IOAs directly attached to the Linux partition” on page 12.

Table 4-3 Directly attached SCSI adapters and their kernel modules

Description	Module name
Base SCSI proprietary driver	ibmsis.o
SCSI Disk (DASD) in base Linux kernel	sd.o
SCSI Tape in base Linux kernel	st.o
SCSI CD-ROM in base Linux kernel	sr.o
SCSI driver in base Linux kernel	scsi_mod.o

When your kernel is installed by one of the distributors, the ibmsis.o module normally matches the installed kernel. When the module is not loaded already, it can be loaded using the command **`insmod imbsis.o`** or **`modprobe ibmsis`** in case the module is installed in the correct directory. When the kernel indeed is ready for that module no *unresolved symbol errors* should occur when trying to insert the module.

Compiling it yourself

When, for one reason or another, the ibmsis.o module is not compiled for the kernel binary you are running now, here is a list of instructions to follow to build ibmsis.o yourself. The following guidelines should help you with this device driver. A readme file is included in the package you download from `http://www.iseries.ibm.com/linux` where you need to go to **Developer Resources** and then **ibmsis iSeries Linux device driver**.

Download both *ibmsis* and the *sisutils*. ibmsis is the driver itself, while sisutils are tools such as sisconfig that is similar to the options you have in DST to “Work with disk units”. For example, you need this to remove parity protection from the disks you want to use in Linux.

The device driver and the associated user space tools require you to use either:

- **`devfs`** (see 3.2.1, “Linux devices” on page 90 for more information on devfs)
- **`scsidev`** that can be found at `http://www.garloff.de/kurt/linux/scsidev/#scsidev`

When changing from devfs to non-devfs or vice versa, several changes are needed, for example in /etc/fstab. We recommend you stick to the approach that your distributor installed.

Follow these steps to compile ibmsis.o:

1. After downloading the driver, untar the packages:

   ```
   tar xzvf ibmsis-$ver.tgz
   cd ibmsis/src
   ```

2. Run **`./config.pl --help`** for a list of available options. Then run with the correct options for your system. The only required options are **`--base`**, which points to your kernel source tree, and **`--devfs`** if you are compiling the device driver for a devfs kernel. For example, the following statement will generate the makefile for using devfs and points to the kernel sources in /usr/src/linux (which is normally correct):

   ```
   ./config.pl --base=/usr/src/linux --devfs
   ```

If this configuration file does not run, you might not have Perl installed. Install it using your distributors provided tools.

3. Compile the module, and let it link to the currently running kernel version. Or better yet, it compiles using the kernel headers located in `/usr/src/linux`, so make sure the running kernel matches with the installed kernel sources.

   ```
   make
   ```

4. Install the module to the appropriate directory (/lib/modules/2.4.x/), using the command:

   ```
   make install
   ```

5. After the kernel is compiled, load the module. Make sure you have the kernel option **`modversions`** disabled. Otherwise dynamically loading will not work with modules that were not available at kernel compile time.

   a. Because this module is installed in the appropriate directory, we can load it using the command:

      ```
      modprobe ibmsis
      ```

   b. Otherwise, when no **`make install`** was done, use **`insmod`** to load the module, as shown here:

      ```
      insmod ibmsis.o
      ```

6. The device driver should now be loaded, and you should be able to see the devices in the /dev file system when using devfs. When this does not occur, make sure you followed the detailed steps as described in the included readme file because the most up-to-date instructions will be contained there.

If you want your system to boot from the native DASD, see 4.4, “Installing the kernel” on page 162.

4.3.7 Compiling the kernel

When configuration is finished, you can go and compile the kernel. Enter the following commands in the order shown:

```
make clean
make dep
make Aside / make Bside / make vmlinux
make modules
make modules_install
```

Normally everything finishes without errors. If you receive errors, you probably selected a module that is not supported by the iSeries Linux kernel. Try to find out which module the error corresponds to and disable that module. (Use the files in the linux/Documentation/ directory or the file linux/scripts/kconfig.tk to determine the invalid module.)

Step 1 (**`make clean`**) does not need to be done each time you compile. As explained in 4.1.4, “Building with make” on page 140, the power of **`make`** is that you don’t need to compile every module each time.

Step 2 (**`make dep`**) calculates all the dependencies that exist between the modules. It determines which modules to compile first, and which it doesn’t need to compile at all. Instead of having its dependencies static, such as in the example in 4.1.4, “Building with make” on page 140, the kernel configuration procedure calculates it semi-dynamically.

Step 3 (**make Aside**) compiles the kernel and immediately places it the IPL at slot A. The command **make Bside** puts the kernel in the IPL at slot B. When doing **make vmlinux**, it creates a file named *vmlinux*. This file can then be placed in the IFS of the hosting OS/400 partition to boot from. The last possibility is to boot from a PReP boot partition in the virtual disk. Use the **dd** command to place it there. All these IPL sources can be chosen in the NWSD. For a more extensive explanation about how to install the kernel, see 4.4, "Installing the kernel" on page 162.

Step 4 (**make modules**) compiles all the modules that are chosen.

Step 5 (**make modules_install**) copies all the modules built in step 4 to their proper directory in the Linux file system. This is normally */lib/modules/2.4.2* in case of a 2.4.2 kernel.

In case you want to erase all configurations and all binaries, in order to start all over again, use:

```
make mrproper
```

Be aware that this also deletes the `.config` file.

4.4 Installing the kernel

There are three locations where the kernel can be installed. Refer to 2.4.3, "Creating an initial network server description (CRTNWSD)" on page 38, for more information about these IPL sources. Table 4-4 explains how to install to these different locations.

Table 4-4 Locations of the kernel and how to get it there

IPL source	Install procedure

A or B slot	1. Do a **make Aside** or **make Bside**
	2. Change the NWSD to boot from the A or B slot: **IPLSRC(A)** or **IPLSRC(B)**
	or
	1. Run the command: **make vmlinux**
	2. Use the **dd** command to copy the file to the appropriate spot in the /proc file system:
	`dd if=vmlinux of=/proc/iSeries/mf/A/vmlinux`
	or
	`dd if=vmlinux of=/proc/iSeries/mf/B/vmlinux`
	3. Change the NWSD to boot from this A or B slot: **IPLSRC(A)** or **IPLSRC(B)**

IFS file on hosting partition	1. Run the command: **make vmlinux** 2. Place the directory in the IFS of the hosting partition (using FTP or some other tool) 3. Change the NWSD to boot from this IFS file: **IPLSRC(*STMF)** and **IPLSTMF(/dirname/vmlinux)**

IPL source	Install procedure

PReP disk partition on first NWSSTG of Linux partition	1. Run the command: **make vmlinux**
	2. Find out which partition on the first disk is the PReP boot partition using **fdisk /dev/viod/disc0/disc**
	Then type **p** to print the partition table. Find out which partition is the PReP boot partition, and quit fdisk by typing **q**. Assume fdisk tells /dev/viod/disc0/disc3 is the PReP boot partition.
	3. Place the kernel in the right disk partition using:
	`dd if=vmlinux of= /dev/viod/disc0/disc3`
	Or from an OS/400 system, perform these steps:
	1. Run the command: **make vmlinux**
	2. Use the OS/400 putkern program to install it, type from the OS/400 hosting partitions command line:
	`call putkern ('/tmp/vmlinux' 'diskname')`
	Here, *diskname* is the name of the NWSSTG (note the space after putkern).

Non-hosted partitions

There is one problem involved with loading the module using **insmod**. This can only be done whenever the system has already booted and a file system has been mounted. Now, when having a stand alone non-hosted partition, the `ibmsis.o` module is needed to mount the file system. Consequently we can't use the normal **insmod** command to install the module.

To make sure the kernel can access the `ibmsis.o` module, it somehow has to be compiled into the kernel, or made sure the kernel can access the module without the need to access a disk. To solve this problem, a so-called initial *ramdisk* is used. This is a kind of disk, but attached to the kernel, where the kernel can load `ibmsis.o` to access its real native disk.

For up-to-date information, refer to your distributors information or the readme file that is included with the ibmsis.o driver when you download it from the Internet.

Important: Before doing any of the following steps, we strongly recommend that you maintain an alternate method of booting your Linux partition in the case where the following steps fail. This is simple to do in iSeries Linux and is the reason for the multiple IPL sources.

1. Create an initial ramdisk containing the ibmsis.o module. Use **make initrd** to build the kernel. This creates `vmlinux.initrd`. (This must be done as *root*.)
2. Store the kernel into the kernel slot you wish to use, A or B (see 4.4, "Installing the kernel" on page 162), for example:

   ```
   dd if=vmlinux.initrd of=/proc/iSeries/mf/A/vmlinux bs=4096
   ```

3. Modify the kernel command line parameters. These are located in /proc/iSeries/mf/A/cmdline. Consider this example (this is on one line!):

   ```
   echo "root=/dev/scsi/host0/bus0/target3/lun0/part1" > /proc/iSeries/mf/A/cmdline
   ```

   Make sure the "`root=`" parameter points to your real root file system.
4. Switch your IPL source so Linux IPLs off of either the A or the B side (depending on what you configured) and reboot Linux.

4.5 Problem determination

Several things can go wrong when compiling or installing programs:

- When installing an RPM file, always make sure you work with the PowerPC version of the package. (filename ends with .ppc.rpm)
- When during the installation of an RPM file, **rpm** starts to complain about failing dependencies, download the RPM files that it depends on with a minimal version that is needed. RPM can upgrade programs as well. To check which version is installed, use

  ```
  rpm -q packagename
  ```

  To download packages, first go to your distributor. If it cannot be found there, go to: http://rpmfind.net

  There is a good chance that the version you need can be found there. Make sure you download the PowerPC version.
- Make sure the programs you try to compile have been adapted to compile on PowerPC. When running the configure script, it will automatically determine which platform it runs on. If all tests are passed, you can be sure there is no problem at all. When installing a source RPM file (.src.rpm), and the PowerPC platform is not detected, this results in the same problem.

  If problems arise, and you're not a developer who can dive into the code and fix problems, there probably isn't much you can change about it. Try to find the latest version of the source program, or go to the Web site of that program to find information about ongoing ports. Most open source developers can be easily contacted by e-mail and are happy to help someone. If you cannot find the Web site, try: http://freshmeat.net

  The only thing you could try is to force the configure script to take another platform. Make sure you choose another little endian platform such as SPARC or PA-RISC. Check out the readme file contained in the source directory to find out how to do this.
- As described in 4.3.7, "Compiling the kernel" on page 161, make sure you always keep a working instance of your kernel so that in the case the newly compiled kernel fails to load, you can always go back into your system to adapt that kernel.

  In the case that no working copy of the kernel can be found, compile a new kernel from another Linux partition, or just take the kernel of that other Linux partition and place it into an IFS streamfile. Then let your broken partition boot from that stream file. You can even build a kernel on another platform such as your PC, using a cross compiler.
- When the kernel itself does not compile, make sure you disabled all Intel or other platform dependent modules. When you start from the original kernel source delivered by your distributor, there should be no problem. Work step-by-step, take the original source, and try to compile it. This should work, and if it does not, something else is wrong (check your compiler version etc). In normal cases, it does compile. Then, add modules step-by-step to determine which module causes the problem.
- Check the Web to make sure you always work with the latest versions of all sources and compilation tools you need. Search mailing lists for information about the problem you are having. After checking whether such an error previously been discussed in the mailing list, post a message to that mailing list yourself.
- If problems persist, make your own contribution to the open source community by sending proper bug reports, that includes, in as many details as possible, an explanation on how to reproduce it.

5

X Windows and OpenOffice

This chapter describes how to make the X Window graphical user interface environment work in an iSeries Linux partition. Several approaches are explained:

- A basic approach. Easy to set up for someone how technical; not very handy for daily use
- How to use a graphical login; very easy to use, especially in a multi-user environment where non-technical people need to work with the machine
- How to use VNC to enable a more flexible way of using X Windows; very handy in some situations
- How to set up OpenOffice, an example GUI program; this a good and usable example of a program that needs X Windows to run

5.1 X Windows concept

X Windows provides a graphical user interface (GUI) to Linux. The programs run on the server side, while the display occurs at the client side. With X Windows, programs have to be installed only once on a Linux system, and then they are accessible to every user on the Linux system.

If you can use the X Window programs, a whole new world is opened for the iSeries server. Instead of only being able to run programs using 5250 display emulation, you can have graphical user interfaces being run natively on the iSeries. These programs have to run in the Linux partition on the iSeries.

For more detailed information about all topics discussed here, you can look at one of the two X Windows Howtos at:

http://www.linuxdoc.org/HOWTO/XWindow-User-HOWTO.html
http://www.linuxdoc.org/HOWTO/XWindow-Overview-HOWTO/index.html

5.1.1 What X Windows is

X Windows can be compared to the 5250 concept on OS/400, as illustrated in Figure 5-1. Just like the 5250 data stream contains a description on how a screen should look like, the X Window stream contains descriptions of how the windows, buttons, and all other GUI components should look. This information is sent to the X Window Server, which renders all components and displays the result on the screen, just like a 5250 terminal (or emulation program such as Client Access) displays the output on the screen.

While the user is working with the program, a 5250 terminal sends the filled in values to the server. The X Window Server also sends information back regarding the mouse position, mouse clicks, and keyboard events. (In the 5250, the information is only sent to the server whenever you press Enter, Page Up, etc.; an X Window Server sends information all the time, so more is interaction possible.)

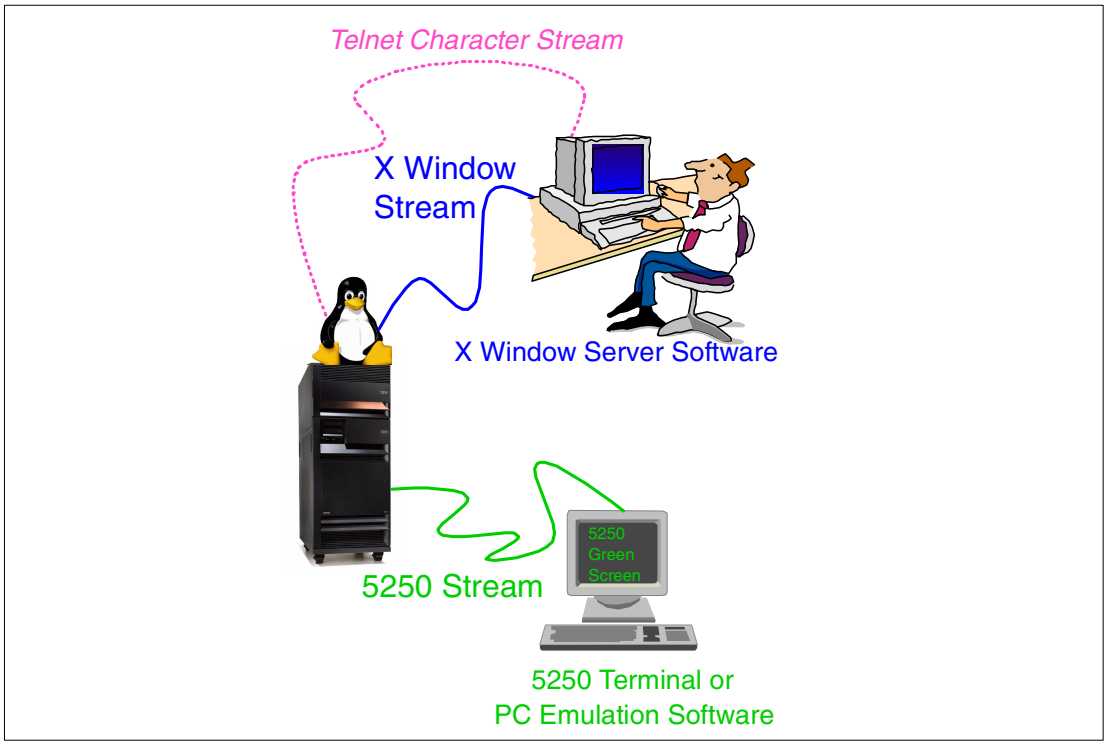


Figure 5-1 The X Windows concept

It might sound strange that the X Window Server is normally running on your PC, and therefore, serves as a piece of “client software”. However it is called “X Window Server” because the GUI program running on your Linux server makes a TCP/IP connection to the X Window Server and so it technically plays the role of a client.

5.1.2 X Window Servers for Microsoft Windows

For PCs running Microsoft Windows, one piece of third-party software needs to be installed on your PC to fulfill the role of X Window Server.

At the Web site `http://www.tucows.com`, you can find a few trial versions of X Window Servers, in the software directory named “X-Window Servers”. You can try them out; for example try *X-Win32*. See `http://www.xwin32.com/`

Another X Server program is Exceed, downloadable from `http://www.hummingbird.com` as a trial version.

5.1.3 X Window Servers for Linux

For PCs running Linux, you don’t need a third-party tool. Each distribution today comes with an X Window Server built in. (Normally XFree86 is installed.) To start your X Window Server on your Linux machine, just start your favorite desktop environment, and you are using it.

5.1.4 IBM Netvista Thin Client

Many IBM Netvista Network Stations are equipped with a flashdisk containing a Linux installation. This means that the graphical user interface is based on X Windows, and you don’t need to explicitly start an X Window Server. You might not see this, but it is started automatically for you.

The Network Stations can easily be used for working with X Window programs on any Linux system, including iSeries Linux partitions. In addition, the IBM Netvista Network Stations have a 5250 emulator, so an integration can be done between 5250 programs running in OS/400 partitions and X Window programs running in Linux partitions.

The Netvista Network Stations can also boot a kernel and a Linux system from a central server. Then they run as what is called a *thin client* (often called a *diskless workstation*, because they don't need a hard disk). Depending on the amount of memory and CPU speed they have, applications can either be run on the client or on the server. This concept can greatly facilitate the administration of a network, because it can be done centrally. There are several open source projects that describe how to set up a server for thin clients. A very popular one is the Linux Terminal Server Project, which you can find on the Web at: `http://www.ltsp.org/`

5.2 Basic X operation

The most straightforward way to run an X Window program is by specifying the remote display for programs to connect to. You can do this on any command line interface, whether it is by Telnet or SSH, or from the virtual console.

Requirement: One basic requirement to run X Windows is a working network connection for both your Linux partition and your PC. See 2.5.1, "Virtual LAN configuration" on page 52, to set up networking.

You must have this because X Window programs need TCP/IP to connect to an X Window Server. This requirement is not there when running Linux on a machine with a display adapter like a PC, for example, where you only need a loopback interface (127.0.0.1). The iSeries obviously doesn't have a display adapter.)

5.2.1 Basic steps

To begin, follow these steps:

1. Find out your PC's IP address. In Microsoft Windows, you can do this by running **`ipconfig`**. On Linux, you would use **`ifconfig`**.
2. Start your preferred X Window Server on your Windows PC or start X Windows on your Linux machine.
3. Log on to the Linux system with the user profile you want to launch an X Window program using ssh, Telnet, or just in the virtual console. Then set the environment variable named DISPLAY in on the command line, as in the following example (where `192.168.140.10` is your local IP address and `:0` is the display number):

   ```
   export DISPLAY=192.168.140.10:0
   ```

   Important: Like many other things in Linux, variable names are case sensitive. DISPLAY is written in uppercase!

   Depending on the X Server you are using and the settings of that X Server, the display number (`:0` in the example) might need to be changed to something else like `:1` or `:1.0`. You can also give a hostname instead of the IP address, but you might not have that for your system.
4. After you've set the DISPLAY variable, start the program you want to run. If you want a simple X terminal with a command line, you could run:

```
xterm -ls
```

Here, the `-ls` option makes sure the logon script is run in this new X terminal to initialize your path and other things. The command is run and the display is shown. Other basic programs that can be run in Linux to test the connection could also be **`xclock`** or **`xload`**.

You should now have a GUI program running from your iSeries Linux partition!

5.2.2 Security

If you don't see the GUI program, a possible cause could be that the X Window program tries to connect to the X Window Server, but the server rejects its connection. This is done for security reasons.

With the concept of X Windows, any program from any server could connect to the X Window Server and mess up your display or read the keyboard (for example passwords). If you see a message such as "Client connection refused by server" (as shown in the following example), this probably means that the X Server is not accepting connections from the IP address of your server.

```
iserieslinux:/etc # xclock
Xlib: connection to "10.5.62.36:0.0" refused by server
Xlib: Client connection refused by Server
Error: Can't open display: 10.5.62.36:0
```

For X Window Servers on your PC, find the configuration of your X Window Server and modify that setting to allow connection from any host, or only the IP address of the Linux server from which you are trying to start programs. For example, for Exceed, you select **Tools -> Configuration -> Security** and add the IP address or hostname of your iSeries Linux partition to the xhost.txt file. For X-Win32, choose **X Config -> Security** and add the hosts to the X-Host list.

For X Window Servers in Linux, you need to use the **`xhost`** command. Type **`xhost +`** on the command line while running X Windows to allow connections of any host. For maximum security, you can only allow connections from the servers you want by running **`xhost -`** first to reject all connections. Then, for each server, run **`xhost +server`**, where *server* is the name or IP address of the Linux server.

There is an even tighter security concept using **`xauth`**. This is to be used when several people are running X Window programs from one and the same server, each displaying the X programs on different displays. If you allow all connections from the server, other people with access to that server could also access your X Window Server. **`xauth`** can be used in this context and concepts as cookies are used. For more information on this, see the X Window User Howto at: http://www.linuxdoc.org/HOWTO/XWindow-User-HOWTO.html

5.2.3 Window managers

For X Windows, you need a window manager. For example, if you don't have a window manager, xterm will not have any handles and you will not be able to move the windows position of the X program.

Such environments as Exceed bring their own window manager that is already started so you can move or close windows. But if you have a naked X Window Server (or work in single window mode), you can start the window manager remotely by simply entering the command to start it, for example **`fvwm2`**, **`gnome-wm`** or **`kwin`**. KDE and Gnome actually provide more functions than just a window manager, and so they are called *desktop environments*.

Of course X Windows is more than just running a program with a graphical user interface. You can launch a complete desktop environment that gives you a complete environment to start programs, configure the system, etc. There are two major desktop environments available in Linux. One of them is *Gnome*, another one is *KDE*. Depending on the distribution and the installation options you choose, one or both of them are installed on your system.

Tip: For PCs running Microsoft Windows, you might want to set your X Window Server to *Single window* mode if you want to work with a desktop environment. All X program windows will then obviously be shown in one single X Server window instead of having the task bar, the desktop, and each application appear in a different X Server window. This also normally enables the windows manager of the iSeries Linux partition.

Gnome

If Gnome is installed, you can launch **`panel`** or **`gnome-session`** after you've set the DISPLAY environment variable. You should then see the Gnome task bar, from which you can access your iSeries Linux partition and all programs with a graphical interface. An example is shown in Figure 5-2.

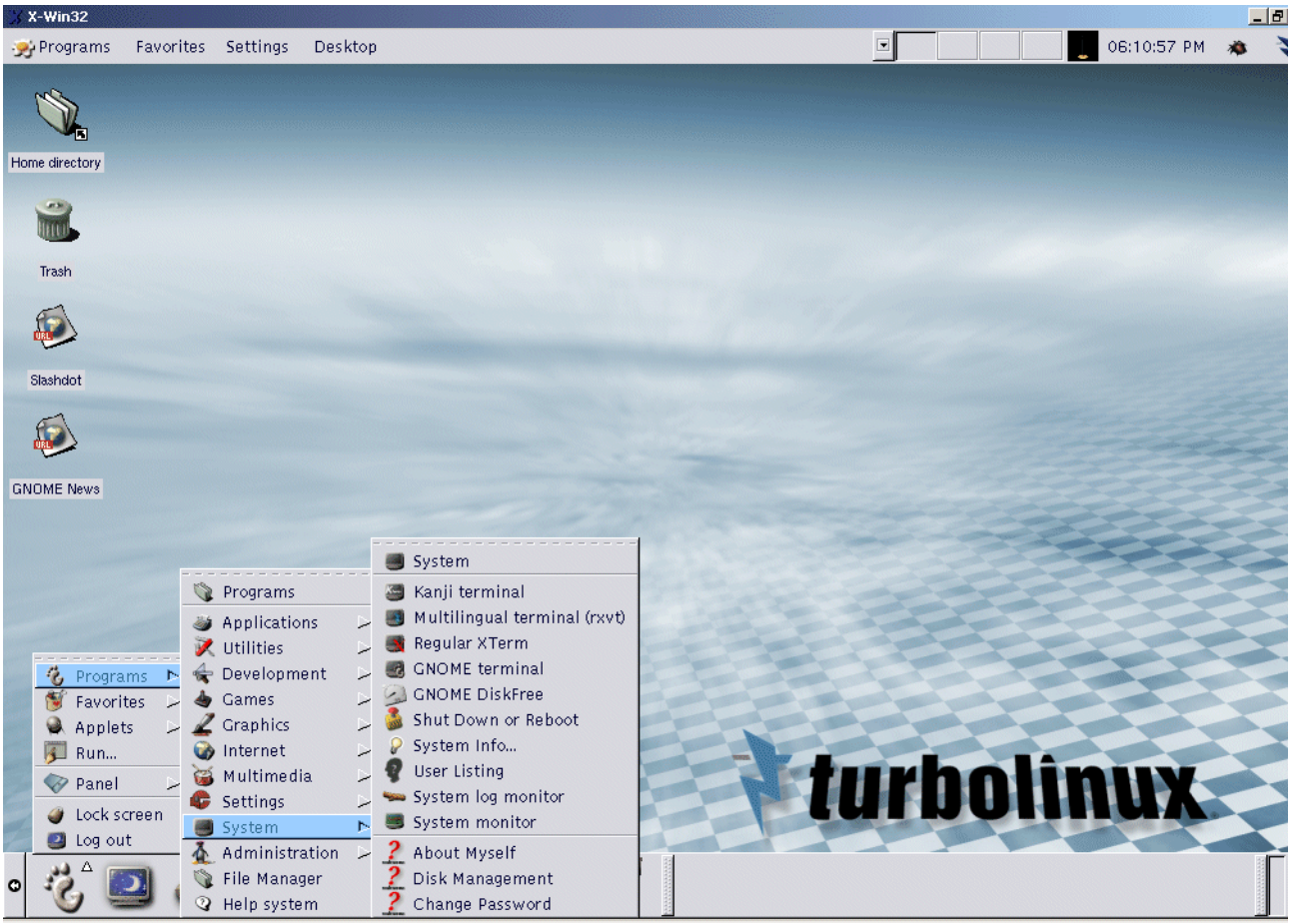


Figure 5-2 An example Gnome session (in Turbolinux) using X-Win32

KDE

If you have KDE installed, simply run **`startkde`** instead to start the KDE desktop environment. An example is shown in Figure 5-3.

If you're having problems stopping KDE, as we had on our machines, you may want to force it to stop by pressing Ctrl-C on the command line where you started it and then explicitly kill the `kdeinit` process by running:

```
killall kdeinit
```

Instead of launching the complete environment, you can start the task bar by simply running:

```
kpanel -no-KDE-compliant-window-manager
```

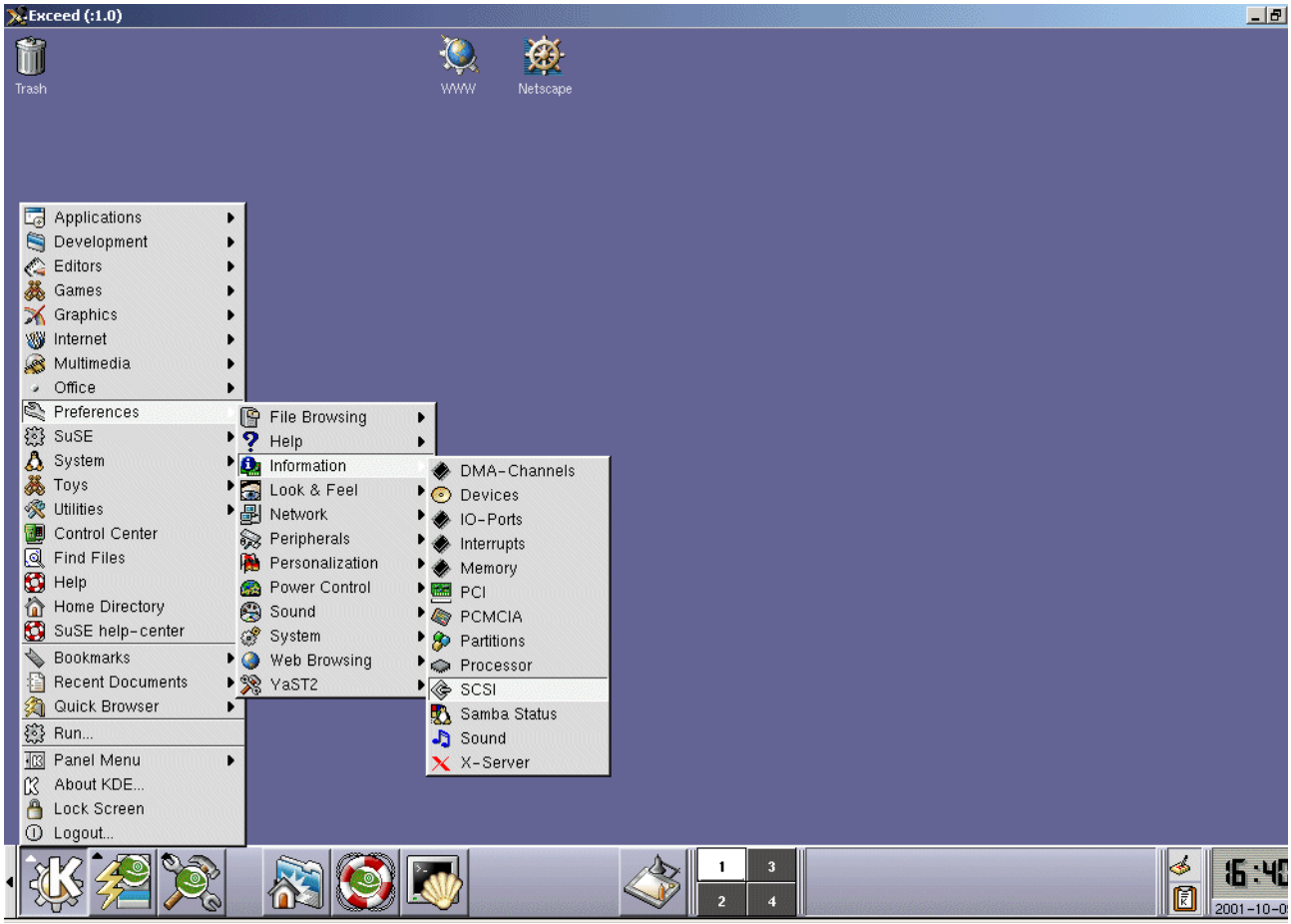


Figure 5-3 An example KDE session (on SuSE), using Exceed as an X Window Server

5.3 A graphical login screen (XDM)

The process just described (where you need to set the DISPLAY variable) is fine for a occasional use of the X Window programs. However, for more frequent use, a graphical login screen is preferable where you enter your name and password and the desktop environment is started. A graphical login screen might look like the one shown in Figure 5-4, where you simply enter the username and password and you automatically see the session.

This approach is very suitable for non-technical people who might be uncomfortable with command lines and IP addresses.

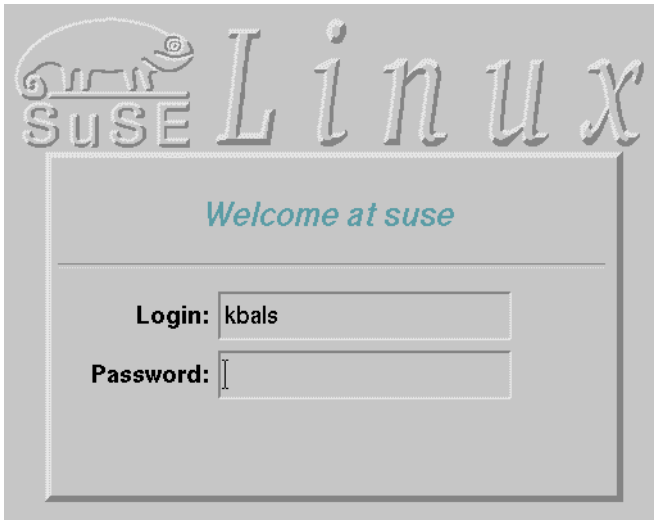


Figure 5-4 A graphical login screen by XDM (on a SuSE Linux partition)

This section explains how to obtain a graphical login screen using the X Display Manager (XDM). Each distribution normally includes XDM. It is included in RPM package xf86 or XFree86 or something similar. For information on installing programs, refer to the distributors explanation, or see 4.2.2, “RedHat Package Manager (RPM)” on page 145.

It might be that some distributions will (in the future) correctly set up XDM by default on iSeries servers. That would make sense because the iSeries does not have any graphical display adapter (like an SVGA card). At the time this redbook was written, however, this was not done by any distributor.

5.3.1 Step 1: Checking the initial XDM configuration

To see how XDM is installed, try the following command when you are logged on as *root*:

```
iserieslinux:~ # xdm -nodaemon -debug 1
DisplayManager.errorLogFile/DisplayManager.ErrorLogFile value /var/log/xdm.errors
DisplayManager.daemonMode/DisplayManager.DaemonMode value false
DisplayManager.pidFile/DisplayManager.PidFile value /var/run/xdm.pid
DisplayManager.lockPidFile/DisplayManager.LockPidFile value true
DisplayManager.authDir/DisplayManager.authDir value /var/lib/xdm
DisplayManager.autoRescan/DisplayManager.AutoRescan value true
DisplayManager.removeDomainname/DisplayManager.RemoveDomainname value true
DisplayManager.keyFile/DisplayManager.KeyFile value /etc/X11/xdm/xdm-keys
DisplayManager.accessFile/DisplayManager.AccessFile value /etc/X11/xdm/Xaccess
DisplayManager.exportList/DisplayManager.ExportList value
DisplayManager.randomFile/DisplayManager.RandomFile value /dev/mem
DisplayManager.greeterLib/DisplayManager.GreeterLib value /etc/X11/xdm/libXdmGreet.so
DisplayManager.choiceTimeout/DisplayManager.ChoiceTimeout value 15
DisplayManager.sourceAddress/DisplayManager.SourceAddress value false
DisplayManager.willing/DisplayManager.Willing value su nobody -c /etc/X11/xdm/Xwilling
Found new display:  :0  local /usr/X11R6/bin/X :0 vt07
...
Before XOpenDisplay(:0)
After XOpenDisplay(:0)
OpenDisplay failed 111 (Connection refused) on ":0"
waiting for server to start 0
Manager wait returns pid: 479 sig 0 core 0 code 1
Server for display :0 terminated unexpectedly, status 1
xdm error (pid 475): Server for display :0 terminated unexpectedly: 1
```

```
Terminating session pid 480
time 1002039096 1002039081
Server crash frequency too high: removing display :0
xdm error (pid 475): Server crash rate too high: removing display :0
Nothing left to do, exiting
iserieslinux:~ #
```

On a basic installation, this normally does not work because it tries to create a local X Server on the local display adapter (where local means “on the iSeries Linux partition”). Of course, it will not be able to start the local X Server because there is no display adapter. As you can see, after a while, it exits with the message like “Nothing left to do, exiting”.

5.3.2 Step 2: Keeping the local displays from starting

Next, you need to remove the line that starts up the local interface. Edit the script */etc/X11/xdm/Xservers* with your favorite text editor. For some help on editing files, see 3.1.2, “Editing files using text editors” on page 88. For information about scripts, see 4.1.3, “Scripts” on page 139.

Make sure you comment out a line similar to the following example (comment out by adding a # character at the beginning):

```
:0 local /usr/X11R6/bin/X :0 vt07
```

If you have a problem editing the file because it is read-only, make it writable by entering the command:

```
chmod +w /etc/X11/xdm/Xservers
```

After editing the file, try to run XDM again:

```
root@iserieslinux:~ # xdm -nodaemon -debug 1
DisplayManager.errorLogFile/DisplayManager.ErrorLogFile value  /var/log/xdm.errors
DisplayManager.daemonMode/DisplayManager.DaemonMode value  false
DisplayManager.pidFile/DisplayManager.PidFile value  /var/run/xdm.pid
DisplayManager.lockPidFile/DisplayManager.LockPidFile value true
DisplayManager.authDir/DisplayManager.authDir value  /var/lib/xdm
DisplayManager.autoRescan/DisplayManager.AutoRescan value true
DisplayManager.removeDomainname/DisplayManager.RemoveDomainname value true
DisplayManager.keyFile/DisplayManager.KeyFile value  /etc/X11/xdm/xdm-keys
DisplayManager.accessFile/DisplayManager.AccessFile value  /etc/X11/xdm/Xaccess
DisplayManager.exportList/DisplayManager.ExportList value
DisplayManager.randomFile/DisplayManager.RandomFile value /dev/mem
DisplayManager.greeterLib/DisplayManager.GreeterLib value /etc/X11/xdm/libXdmGreet.so
DisplayManager.choiceTimeout/DisplayManager.ChoiceTimeout value 15
DisplayManager.sourceAddress/DisplayManager.SourceAddress value false
DisplayManager.willing/DisplayManager.Willing value  su nobody -c /etc/X11/xdm/Xwilling
Nothing left to do, exiting
```

OK, it is still exiting without doing anything, but it is not giving any error messages anymore either.

5.3.3 Step 3: Giving access to connect

Apart from the lists of hosts allowed to connect to your X Server (refer to 5.2.2, “Security” on page 169), there also exists a list of hosts that can connect to your XDM server. This list is located on the iSeries Linux partition in the /etc/X11/xdm/Xaccess file. To allow hosts to connect to your XDM server, you need to *remove the comment* (remove the first “#” character) from the following line:

```
# *                                     #any host can get a login window
                    <<< Remove the comment from previous line to enable hosts to connect
                     <<< using XDMCP
```

You can also define a list of hosts that can connect, but it is easier to accept all connections from any host. Evidently, this is less secure. In some situations, where your PC the IP address is assigned dynamically with DHCP, you can't just specify one host because you don't know which IP address to accept, but you should specify a range.

5.3.4 Step 4: Starting the listener

The next step is to make the listeners start so we can connect to the iSeries Linux partition using our XDM-aware X Server. To do this, edit the /etc/X11/xdm/xdm-config file and *comment out* the following line time by adding an exclamation mark (!) at the beginning of the line:

```
! SECURITY: do not listen for XDMCP or Chooser requests
! Comment out this line if you want to manage X terminals with xdm
DisplayManager.requestPort:   0  <<< Comment out this line by adding a '!' at the beginning
```

Try to start XDM again:

```
root@iserieslinux:~ # xdm -nodaemon -debug 1
DisplayManager.errorLogFile/DisplayManager.ErrorLogFile value  /var/log/xdm.errors
DisplayManager.daemonMode/DisplayManager.DaemonMode value  false
DisplayManager.pidFile/DisplayManager.PidFile value  /var/run/xdm.pid
DisplayManager.lockPidFile/DisplayManager.LockPidFile value true
DisplayManager.authDir/DisplayManager.authDir value  /var/lib/xdm
DisplayManager.autoRescan/DisplayManager.AutoRescan value true
DisplayManager.removeDomainname/DisplayManager.RemoveDomainname value true
DisplayManager.keyFile/DisplayManager.KeyFile value  /etc/X11/xdm/xdm-keys
DisplayManager.accessFile/DisplayManager.AccessFile value  /etc/X11/xdm/Xaccess
DisplayManager.exportList/DisplayManager.ExportList value
DisplayManager.randomFile/DisplayManager.RandomFile value /dev/mem
DisplayManager.greeterLib/DisplayManager.GreeterLib value /etc/X11/xdm/libXdmGreet.so
DisplayManager.choiceTimeout/DisplayManager.ChoiceTimeout value 15
DisplayManager.sourceAddress/DisplayManager.SourceAddress value false
DisplayManager.willing/DisplayManager.Willing value  su nobody -c /etc/X11/xdm/Xwilling
creating socket 177                 <<< The listener is started!!!
Created chooser socket 5            <<< The listener is started!!!
WaitForSomething
```

This time the listener is started and you should be able to connect with your XDM-aware X Window Server. Do not interrupt XDM and go to the next step.

5.3.5 Step 5: Configuring and starting the X Window Server

Before you configure the X Window Server, first try to start an X Program the simple way if you haven't done so already. This is explained in 5.2, "Basic X operation" on page 168.

For PCs running MS Windows, you might want to set your X Window Server to Single window mode if you want to work with a desktop environment. All X program windows will then obviously be shown in one single X Server window instead of having the task bar, the desktop, and each application appear in a different X Server window. This also normally enables the windows manager of the iSeries Linux partition.

What you basically need to do is open or create an X session to the iSeries Linux partition using *XDMCP-Query* as the startup mode. For XDMCP-Query, you also need to fill in the hostname or IP address of the iSeries Linux system. In Figure 5-5 and Figure 5-6, you see the Exceed and X-Win32 windows to set up an X session.

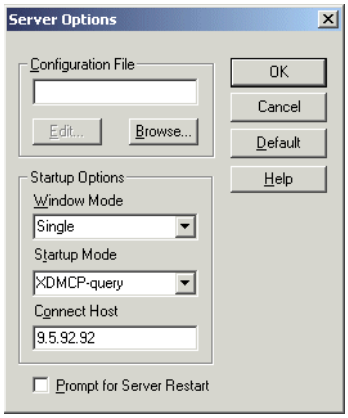


Figure 5-5 Setting up an X Session with Exceed

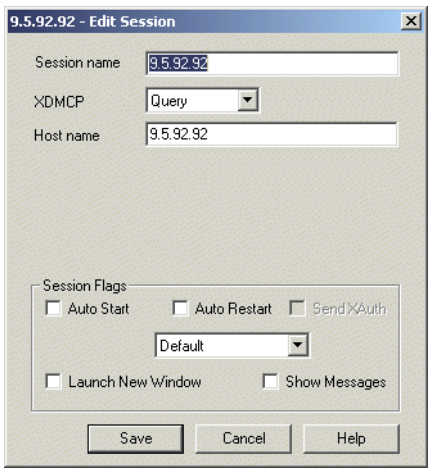


Figure 5-6 Setting up an X session with X-Win32

For the X Server to successfully start the X Session, you must choose to explicitly run it when starting the X Window Server. For example in Exceed, you must choose the **Run!** menu option of the X session configuration screen to run it. For X-Win32, make your selection in the **Session** menu after you've saved the previously configured session.

From a PC running Linux, you can also start an X Window Server with an XDMCP-Query X Session. Start an X Server on your second display (use Ctrl-Alt-F8, for example, to access it) by running:

```
Xwrapper :2 -query hostname
```

To start such a session inside of another X Window Session, use:

```
Xnest :2 -query hostname
```

For Xnest, you need to install package xextraXXX.rpm.

If you start a session to a SuSE machine for example, you should see the display shown in Figure 5-4 on page 172. A session to Turbolinux would produce the display shown in Figure 5-7.

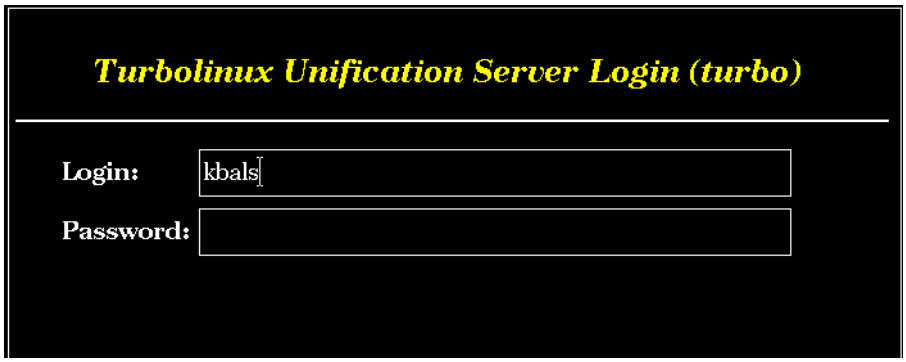


Figure 5-7 A graphical login screen by XDM (on a Turbolinux partition)

5.3.6 Step 6: Automatically starting XDM

The most important thing to do for running XDM automatically when the system is started is to set the runlevel. Normally distributions automatically start XDM when they are running in runlevel 5, and by default, they install in runlevel 3. To make sure your systems boots in runlevel 5, modify the file /etc/inittab in the following line:

```
# default runlevel
id:3:initdefault:
```

Change the *3* (the default value on a system) to a *5*. Sometimes configuration programs, such as YaST, will overwrite this value when they are run. Make sure you take that into account. For YaST specifically, take a look at http://sdb.suse.de/sdb/en/html/ke_inittab-rc.html for a solution.

Checking whether XDM is started

To see if it works, reboot the Linux system and run the following command as *root* to see if **xdm** is running:

```
iserieslinux:~ # pidof xdm
3725
```

The number you see (3725 in the example) is the process ID of the **xdm** process. (See 3.7, “Monitoring the system” on page 108, for information on Linux processes and how to control them.) If no number is shown, **xdm** is not running. In that case, read the following section to find out how to start it.

Why XDM starts or doesn’t start

If you want, you can double check whether XDM is started when runlevel 5 is booted. There are two approaches. One of them is accomplished by the following lines in /etc/inittab. This line tells to the Linux system that XDM should be started in runlevel 5.

```
# Run xdm in runlevel 5
# xdm is now a separate service
x:5:respawn:/etc/X11/xdm -nodaemon
```

The other possibility is that in /etc/init.d/rc.d/rc5.d, there exists a link to ../xdm. If you don’t know how to accomplish this, you should use the distributor’s specific tool to configure **xdm** to run when runlevel 5 is started.

5.3.7 Possible problems with XDM

You may encounter problems during the XDM configuration. Some of these problems are discussed here.

Firewall causing problems

If you don't succeed in getting a connection, it might be that a firewall is running. For information on how to disable or adapt this firewall to fulfill your needs, refer to the distributors installation manual or user's guide.

Cannot login as root

If you can't login to graphical login of the iSeries Linux machine using root, this might be caused by some special built in checks. For example on SuSE, if this happens, you should change the configuration variable ROOT_LOGIN_REMOTE to "yes" using YaST.

Actually, for security reasons, you should *not* log on with root. If you want to perform a task as root, run **`ssh -X root@localhost`** on a command line in X Windows, enter the root password, and then do the task.

5.4 Virtual Network Computing (VNC)

An alternative for the default X Window Servers is VNC. It uses a different approach and needs special installation at the server side as well, as opposed to the regular operation. VNC is actually an X Window Server that is run at the server side instead of on your PC. Then the rendered screens are sent to the client where they are displayed. (With a normal X Server running on a PC for example, rendering and display is both done on that PC.)

5.4.1 VNC concept

Because the X Server (called **`vncserver`**) runs at the server side, the X Applications communicate to that server using an X Window Stream. Using **`vncclient`**, on your PC, you can communicate to the vncserver using a framebuffer protocol, called the VNC protocol. You can use any device at the client side actually, not only PCs.

The VNC Web site (`http://www.uk.research.att.com/vnc/`) describes this tool very well.

5.4.2 VNC installation and operation

To install VNC on the Linux partition, install an RPM package that you find with your distribution. For information on how to install RPM packages, refer to your distributors documentation, or see 4.2.2, "RedHat Package Manager (RPM)" on page 145.

Make sure you have the latest version installed! Eventually check for updated versions from your distributor. At the time this redbook was written, we had some problems using an older version, but we had success with `vnc-3.3.3r1-91.ppc.rpm`. The problem we had with the older versions of the vncserver was a scrambled screen.

Starting the server

After you install VNC on the Linux system, the first time it runs it asks for a password that a vncclient needs to supply before the virtual screen is displayed. Just enter the password, and it will be scrambled and stored in the file *passwd* in the *.vnc* directory in your home directory.

Run **vncserver** and it will automatically start a default environment with its own window manager:

```
root@iserieslinux:~/.vnc # vncserver :2     <<< Run the vncserver and explicitly specify a
                                            <<< display number, :2 in the example
                                            <<< The display number does not need to be
                                            <<< specified and it will pick a free one.
You will require a password to access your desktops.

Password:                                   <<< First time it asks for a password
Verify:

New 'X' desktop is iserieslinux:2           <<< The display number on which it started is
                                            <<< is shown.
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/iserieslinux:2.log
```

Starting the client

To see the virtual screen that is rendered at the server side, you need to start a **vncclient** (for example, on your PC running Windows) that can be downloaded from the VNC Web site in binary form. Start it, and it will ask for a server and a display number as shown in Figure 5-8. Make sure you enter the IP address or hostname and the display number of the vncserver you just started.

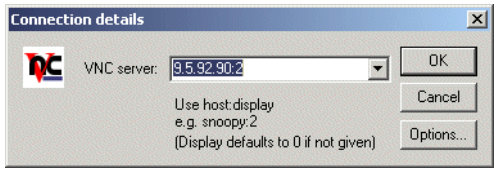


Figure 5-8 A Windows VNC client

After you enter the password, you see a windows like the example in Figure 5-9 that shows you a typical X terminal that is started by default.

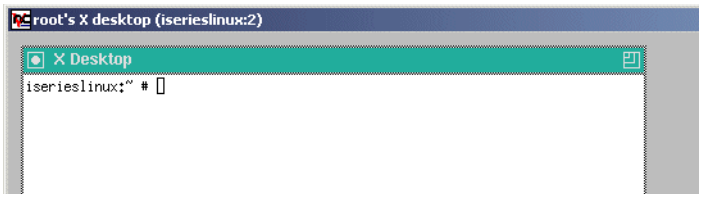


Figure 5-9 A default VNC view

Stopping the server

To stop the vncserver, issue the following command, including the display number of the vncserver that should be stopped:

```
root@iserieslinux:~/.vnc # vncserver -kill :2
Killing Xvnc process ID 1189
```

Changing the server to start a desktop environment

You can actually start any X Window program automatically when the vncserver starts. When vncserver is called, it calls the shell script ~/.vnc/xstartup (= the file named *xstartup* in the directory *.vnc* in your home directory).

For example, if you want to start a desktop environment, like KDE or Gnome, you can start them here. Set the **`startkde`** or **`gnome-session`** commands in the shell script to start your favorite desktop environment.

For more information about shell scripts, see 4.1.3, “Scripts” on page 139. See 3.1.2, “Editing files using text editors” on page 88, for information on how to edit a document.

Tip: You can set **`vncclient`** in full-screen mode to have a complete view of your desktop instead of scroll bars at the bottom and right-hand side. Use the icon menu in Microsoft Windows to do this, or press F8 in the Linux vncclient.

5.5 OpenOffice

OpenOffice is a multi-platform office suite. It is an open source version of the StarOffice suite from Sun. Visit http://www.openoffice.com for more information. It can be downloaded in various compiled forms as well as the source. It contains a spreadsheet, word processor, presentation program, etc. An example of OpenOffice on Linux running inside an iSeries partition is shown in Figure 5-10.

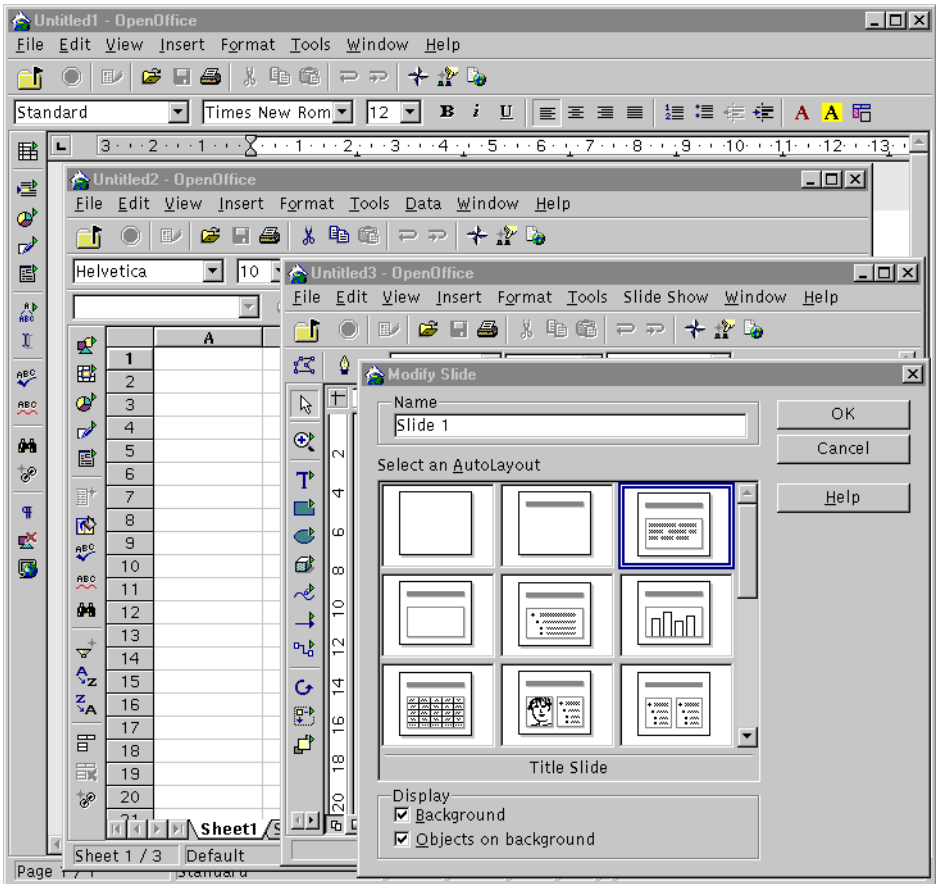


Figure 5-10 OpenOffice running on a Linux system in an iSeries partition

5.5.1 Installing OpenOffice

There are two ways to install OpenOffice once you have obtained it:

- Single-user installation
- Multi-user installation

We start with the single-user installation which is a little bit easier. Later we explain the multi-user installation, so we will work with the *root* user profile all the time (since we need write-access to system-wide directories). However, if you only want to install this for one user (for example, your own user profile), you can do all these steps using a normal user profile.

Obtaining OpenOffice for PowerPC Linux

You can obtain a PowerPC compiled version of OpenOffice on various Web sites. An important detail is that you need a PowerPC version. These versions are not as common and at the time this redbook was written, a PowerPC build is not available from `http://www.openoffice.org`. We found the latest version on several Web sites, including:

`ftp://ftp.suse.com/pub/projects/powerpc/openoffice/`
`http://penguinppc.org/files/pub/openoffice/`

Expanding the .tar.gz file

Normally the OpenOffice installation program itself is packed in a .tar.gz, or .tgz archive. This file can be instantly unpacked, for example:

```
root@iserieslinux:/tmp # tar xzvf oo638c_installer.tar.gz
```

Single user installation

In our example, the directory `oo638c_installer` is created. Change to the created directory and run **`./setup`**. After running setup, a window like the one shown in Figure 5-11 appears.

```
root@iserieslinux:/tmp # cd oo638c_installer
root@iserieslinux:/tmp/oo638c_installer # ./setup
```

Tip: Run setup in an X Windows-enabled session because the installer itself needs X Windows to run. Otherwise, an error message similar to "./setup: cannot connect to X server" will appear.

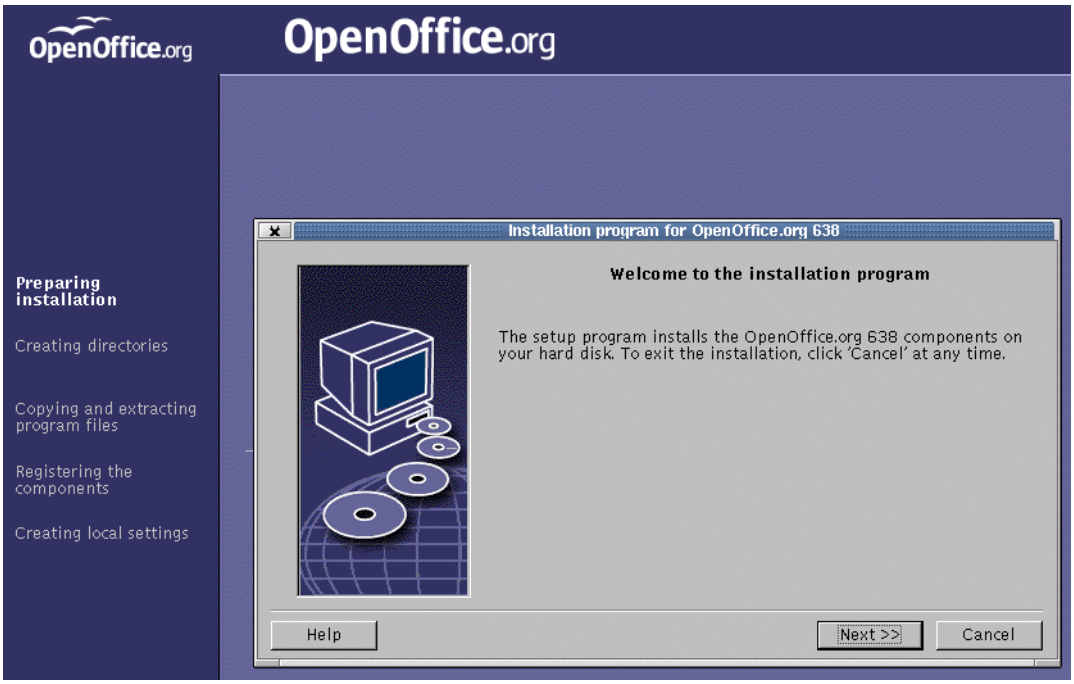


Figure 5-11 The OpenOffice setup program

Follow the instruction guidelines. Since we are doing a single-user installation, install this in your home directory. In our example, as we are logged on as *root*, we install it in /root. If you are doing this as single-user installation for a normal user, log on with that username and install it in /home/*username*.

After installing, change to the created directory and run **`./soffice`**.

OpenOffice now should be installed.

Multi-user installation

If you already installed a single user version, it's better to delete it and start over again. You can delete it with:

```
rm -rf OpenOfficeDirectoryName
```

Next, logged on as *root*, issue this command from within the "install" directory (note the /net option):

```
./setup /net
```

At first instance, you don't see anything special about this setup mode, and it seems exactly the same a single-user installation. The difference becomes clear later on. Install it into a publicly accessible directory, for example:

```
/opt/openoffice
```

Now each user who wants to use OpenOffice, should run **`/opt/openoffice/program/setup`** and follow the simple installation procedure. They should select **Workstation Installation** when prompted. The programs are not copied; the installation program just creates a local directory in the home directory of that user with user settings. This only takes about 1 MB of disk space.

After installation, change to the install directory of that user and start OpenOffice, for example:

```
kbals@iserieslinux:~ > cd OpenOffice.org638/
kbals@iserieslinux:~/OpenOffice.org638 > ./soffice
```

The multi-user version of OpenOffice should now be running.

Starting OpenOffice

To make it more convenient to start OpenOffice, you might want to add ~/OpenOffice.org638 to the PATH environment variable, or specify a system wide alias. This alias can be done by adding the following line at the end of the /etc/profile.local file:

```
alias soffice=~/OpenOffice.org638/soffice
```

You can achieve the same by running the following command:

```
echo 'alias soffice/~/OpenOffice.org638/soffice' >> /etc/profile.local
```

It might be a little early to use OpenOffice at this moment in a business environment, because not all functions have been implemented yet and it still has to prove its stability. However, it can serve you very well and does a very good job for editing documents, spreadsheets, presentations, etc.

6

Apache HTTP server on Linux

This chapter describes the Apache Web server when running it on a Linux partition on the iSeries server. It covers the following topics:

- IBM HTTP Server (powered by Apache) in OS/400
- The configuration file
- Default configuration settings
- Operation of the Apache Web server
- Hosting several Web sites on one host (virtual hosts)
- Dynamic content Web pages (CGI and PHP)

This chapter is *not* meant to be a complete reference on the Apache Web server. Please refer to the official Apache Web site at `http://www.apache.org` for more in depth information.

Apache is *the* HTTP server for Linux. Every iSeries Linux distribution installs it or has the ability to install it. Normally, an extensive explanation on how to set up and configure your Apache server using the tools is provided by the distributor. For specific installation instructions, refer to the distributor documentation.

At the time of this writing, Apache Version 1.3 is available. Version 2.0 has been released to the public in the beta version, but it may take a while before it's officially declared stable. Its most important new feature is that it provides support for multi-threaded server processes, which can be combined with the currently existing multi-process approach to result in better scalability (although the Apache Web server already is very scalable).

6.1 OS/400 HTTP Server (powered by Apache)

Do not confuse the Apache Server in iSeries Linux with the IBM HTTP Server in OS/400. IBM developed an HTTP Server for OS/400, which is based on Apache 2.0 (5722-DG1). This might sound odd, since Apache 2.0 is still in beta. However, IBM created its own version out of the Apache source code and is giving the guarantee to OS/400 users that it is working.

Since the IBM HTTP Server (powered by Apache) for OS/400 is a separate version, most Apache modules (see 6.2, “Apache modules” on page 184) will not work. People are in the process of porting modules, but right now there is no real Hypertext Preprocessor (PHP; a certain kind of scripting language to embed code in html pages, similar to Active Server Pages) support for example.

IBM has added functionality to enable you to configure the HTTP Server from within a browser. This functionality is only for the IBM HTTP Server and is not available in the Linux version. You can lean more about this on the Web at:
http://www.ibm.com/iseries/software/http/services/apache.htm

6.2 Apache modules

Apache has support for plug-ins, called *modules*. There are lots of Apache modules available, all with their own purpose. There are Apache modules for PHP, for streaming MP3 music files, XML support, XML RPC support, etc.

However, you should only install those you really need, because modules might create conflicts, for example, when using two modules that perform functions on files with the same extension (Which one is executed? Only one? Both? In which order?).

For a complete list of modules, see http://modules.apache.org

6.3 Configuration

The main configuration file that Apache uses is /etc/httpd/conf/httpd.conf. This file can be edited by any text editor to configure the server. For help using a text editor, see 3.1.2, “Editing files using text editors” on page 88.

The location of this configuration file might depend on the distribution. It should be in /etc/httpd/conf/httpd.conf or /etc/httpd/httpd.conf. If you have problems finding it, you can issue the following command:

```
find / -name httpd.conf
```

This httpd.conf configuration file contains very useful comments.

The httpd daemon only reads the configuration file at startup or when being restarted. This means that when you change the configuration file, you have to restart the HTTP server, normally by issuing the command:

```
/etc/rc.d/init.d/httpd restart
```

When Apache is started (httpd daemon), it is listening by default on port 80 and 443 (SSL), which are the default port numbers for HTTP and SHTTP.

6.3.1 Default Web page

The directory where the home page is located greatly depends on the distribution. A good indication of where this directory is located can obtained by running the following command:

```
rpm -ql apache | grep index.html
```

The directory location can also be determined by looking in the configuration file for the *DocumentRoot* configuration value. Use your text editor to find it, or you can this command:

```
grep DocumentRoot /etc/httpd/httpd.conf
```

The file index.html is the default presented page. This means that if you go to http://*hostname*/, where *hostname* is the name or IP address of the Linux partition, the index.html file will be sent to you. The name of the default page can be configured to be something different from index.html.

6.4 Operation

Depending on the installation, Apache might be automatically started. Refer to your distributor's documentation for information on how to accomplish this. You can verify whether it is running by using the command **`pidof httpd`**. When it prints out at least one number, the server is started.

You can start it manually by running either of the following commands:

```
/etc/rc.d/apache start
/etc/rc.d/httpd start
```

6.4.1 Access control

There are several ways of changing access control on your systems.

If you are having authority problems on the files you are trying to access on a default installation, you are dealing with the file-system authorities. Try to use the "`ll`" in the directory and check if the file has the right authority.

Authority can be changed with the by issuing commands similar to:

```
chmod 755 [filename/directoryname]
```

See 3.8.1, "Changing file ownership (chown and chgrp)" on page 112, for information on **`chmod`**.

However, Apache also has built-in authorization settings that allow you to ask for a username and password before giving access to certain locations.

6.4.2 Virtual named hosts

Apache running on iSeries Linux gives the opportunity to run multiple Web sites on one IP address. This makes it possible for you to provide ISP services, such as hosting, very easily. Implementing it is easy, but you might want to consider that not all browsers support it. Usually all newer versions of browsers support it.

In the old days (HTTP 1.0) when a Web page was to be retrieved, if a user asked the browser to retrieve "HTTP://www.hostname.com", the browser sent a request to the DNS to have the name resolved into an IP address so that the browser could send a "GET HTTP://x.x.x.x" to that IP address. The server did not see the originally requested hostname because it was already translated into an IP address.

Starting from HTTP 1.1, the hostname is sent in the HTTP request, and so the HTTP server can detect which one of its virtual hosts would be shown.

The actual configuration is quite simple. In the configuration file, a few lines have to be added to enable virtual hosts. The example shows you a server running on IP address 192.168.1.2, which serves three virtual hosts. One of them has its DocumentRoot in /var/www/dummy.com/, another one in /var/www/yourdomain.com, and the third one in /var/www/yourpal.com. Consequently, when the browser requests the URL http://www.yourdomain.com/filename.html, the browser sends the file /var/www/yourdomain.com/filename.html from its file system.

Here is an extract from the httpd.conf configuration file:

```
NameVirtualHost 192.168.1.2

   <VirtualHost 192.168.1.2>
      ServerName www.dummy.com
      DocumentRoot /var/www/dummy.com/
      </VirtualHost>

   <VirtualHost 192.168.1.2>
      ServerName www.yourdomain.com
      DocumentRoot /var/www/yourdomain.com
   </VirtualHost>

   <VirtualHost 192.168.1.2>
      ServerName www.yourpal.com
      DocumentRoot /var/www/yourpal.com
   </VirtualHost>
```

Of course, this involves more than just setting up the HTTP server. You need to register DNS entries that link the names www.yourdomain.com and www.yourpal.com to the correct IP address.

Old versions (or very old versions) of browsers might still send the request as "GET HTTP://x.x.x.x" (where *x.x.x.x* is the resolved IP address mentioned earlier). When an old browser requests "GET HTTP://x.x.x.x" to an Apache server configured for virtual named hosts, Apache passes the Web page of the first defined virtual host. Because of this, it is advised that the first defined named based host is just a dummy domain name (as in the example). In this directory, you could the place a start page (default named index.html), which informs the user that they should upgrade the browser and try the request again.

For more in depth information about virtual name based hosts, you might want to visit the Apache Web site at: http://httpd.apache.org/docs-2.0/vhosts/index.html

6.5 Dynamic content Web pages

Modern Web pages contain more than just static content. There are a number of ways to add dynamic content.

6.5.1 CGI programs

A Common Gateway Interface (CGI) program is a normal program that is called whenever the user requests its filename. Instead of just sending the binary of the program, the Web server executes it. This must be configured in the httpd.conf file, using the *ScriptAlias* configuration value.

The program is called with a specific set of arguments that represent the desired environment (like variables). Everything the program would normally print to the screen is redirected to the browser. This way the CGI program can send dynamic data. CGI programs can be written in any programming language, like C and C++, but very often Perl or Python are used because they have good text handling functions built in.

Generally, CGI bins are not used too much anymore, because developing them requires quite a lot of effort.

6.5.2 PHP

The PHP scripting language is very popular lately. It works in a way that is similar to JSP; you insert code in between the HTML code. This is done with special <? and ?> tags. For example, consider the following PHP source:

```
<html>
   <body>
      <?php echo “Hi there” ?>
   </body>
</html>
```

It generates as output (when it’s requested with a browser from a properly configured Apache server):

```
<html>
   <body>
      Hi there
   </body>
</html>
```

This makes it very easy to add dynamic content to a Web site. Mostly PHP is used to interact with databases, like the open source MySQL database (http://www.mysql.com). However, once the IBM DB2 Universal Database is running on iSeries Linux, it will be possible to use that database server by means of an ODBC driver. It will also be possible to connect to the OS/400 DB2 database by means of ODBC. For ODBC, see Appendix B, “ODBC” on page 285.

To use Hypertext Preprocessor (PHP), install the RPM packages phpXXX.ppc.rpm and all required packages. This installs the PHP module for the Apache server.

There are a lot of open source projects going on that make use of PHP to provide some function in a Web browser. For example e-mail clients, calendars, online discussion forums, interfaces to databases, and other useful examples, see http://www.freshmeat.net and select **Browse -> Browse By Programming Language -> PHP**. At time this redbook was written, more than 700 projects were registered there!

For more in depth information on PHP, go to: http://www.php.net

6.5.3 JavaServer Pages

You can also generate dynamic content by using JavaServer Pages (JSP) in combination with servlets which enable you to embed pieces of Java code inside of your HTML files. If you plan to use this, see Chapter 7, “Tomcat Web Application Server (using Java) on Linux” on page 189.

6.6 Problem determination

In case of problems with your Apache Server, we recommend that you look into the distributors documentation, or visit the Apache Frequently Asked Questions (FAQ) at:
http://httpd.apache.org/docs/misc/FAQ.html

7

Tomcat Web Application Server (using Java) on Linux

This chapter describes the Tomcat Web Application Server running on a Linux partition on the iSeries server. It explains:

- What a Web application server is
- How to install Tomcat
- How to use Tomcat
- The differences between Tomcat and WebSphere Application Server

7.1 Introduction

This section introduces you to Jakarta Tomcat. For more information, refer to the Web site at: http://jakarta.apache.org/tomcat/index.html

7.1.1 What Tomcat is

Tomcat is the official Reference Implementation for the Java Servlet and JavaServer Pages (JSP) technology. Tomcat 4.0 is an implementation of the Java Servlet 2.3 and JSP 1.2 specifications.

Java Servlets and JavaServer Pages enable you to make dynamic content Web sites using Java as a programming language. What you can do is embed certain pieces of Java code inside of Web pages. This way you can retrieve data from databases to fill in the Web page or fill in data in the database such as a product order or a user Web profile.

JSP pages are like html pages that contain pieces of Java code but as little programming logic as possible. The servlets will do all the work and pass the results to the JSP pages that make calls to the servlets whenever they need some information or need to make a decision. JSP pages, therefore, form a shell around the servlets.

Tomcat is programmed in Java and runs on every platform that has a Java Virtual Machine (JVM). This chapter only covers the installation of Tomcat on iSeries Linux, but you could also install it on an OS/400 partition.

7.1.2 License

Basically, Tomcat is free. It is an open source project and is released under the Apache Software License, which basically says that you are free to modify the source code and redistribute it in binary form without needing to publish your own changes to the code. This is a bit different from the GNU General Public License where you have to publish your own changes when redistributing the product.

Actually, the Apache license is relatively liberal. The Apache Web server for example, which also falls under the terms of the Apache license, has been modified and extended by IBM to run on OS/400. However, IBM was not obligated to release their source code with these modifications. And they effectively didn't release their sources.

Go to the http://www.apache.org/licenses/ site for the exact license.

7.2 Installation and operation

This section explains how to install and use Tomcat.

7.2.1 Installing a Java Virtual Machine

First of all, make sure a Java Virtual Machine is installed, since Tomcat is a Java program. We strongly encourage you to install the IBM Java Virtual Machine for PowerPC Linux. You can download it from: ftp://ftp.suse.com/pub/suse/ppc/update/7.1/pay2/

See 4.2.2, "RedHat Package Manager (RPM)" on page 145, for information on how to install RPM packages. The IBM Java Virtual Machine needs a recent version of **glibc**, which at time of this writing is only available in SuSE.

Another JVM can be found at http://www.blackdown.org. The problem with this is that this JVM does not support Java classes for runtime compilation, which results is errors when trying to run JSP pages. However, servlets work with the Blackdown JVM.

7.2.2 Acquiring and installing Tomcat

Download Tomcat from the following Web site:
http://jakarta.apache.org/builds/jakarta-tomcat-4.0/release/v4.0/

At time this redbook was written, Tomcat Version 4.0 is the most recent, so that's what is used in the examples. Download the *jakarta-tomcat-4.0.tar.gz* file to the iSeries Linux system, and untar it. Then use **less** or your favorite editor to read installation instructions in the files README.txt and RUNNING.txt.

Remember, Linux is case sensitive. (See 3.1.2, "Editing files using text editors" on page 88 if you need help on editing/viewing files.) Consider this example:

```
iserieslinux:/tmp/tomcat # tar -xzvf jakarta-tomcat-4.0.tar.gz
iserieslinux:/tmp/tomcat # cd jakarta-tomcat-4.0
iserieslinux:/tmp/tomcat/jakarta-tomcat-4.0 # less README.txt
iserieslinux:/tmp/tomcat/jakarta-tomcat-4.0 # less RUNNING.txt
```

7.2.3 Starting and stopping Tomcat

Before you start Tomcat, make sure you set the JAVA_HOME environment variable. For example, the IBM Java Virtual Machine was installed into /opt/IBMJava2-ppc-13, so we run:

```
iserieslinux:/tmp/tomcat/jaka...-4.0 # export JAVA_HOME=/opt/IBMJava2-ppc-13
```

And then we change to the bin/ directory and start the server by running the startup script:

```
iserieslinux:/tmp/tomcat/jakarta-tomcat-4.0 # cd bin
iserieslinux:/tmp/tomcat/jakarta-tomcat-4.0/bin # ./startup.sh
Guessing CATALINA_HOME from catalina.sh to ./..
Setting CATALINA_HOME to ./..
Using CLASSPATH:     ./../bin/bootstrap.jar:/opt/IBMJava2-ppc-13/lib/tools.jar
Using CATALINA_BASE: ./..
Using CATALINA_HOME: ./..
Using JAVA_HOME:     /opt/IBMJava2-ppc-13
```

The server is now started in the background so you can log out and still Tomcat stays running. This implies that you don't see any output. Section 7.2.4, "Log files and solving problems" on page 192, explains how to use the log files.

Try to connect to the Linux partition using a browser on port 8080. (This is the default port Tomcat uses.) For example, type:

```
http://10.5.92.90:8080/
```

If there is an error, refer to 7.2.4, "Log files and solving problems" on page 192.

To stop Tomcat, simply run the shutdown script:

```
iserieslinux:/tmp/tomcat/jakarta-tomcat-4.0/bin # ./shutdown.sh
Guessing CATALINA_HOME from catalina.sh to ./..
Setting CATALINA_HOME to ./..
Using CLASSPATH:     ./../bin/bootstrap.jar:/opt/IBMJava2-ppc-13/lib/tools.jar
Using CATALINA_BASE: ./..
Using CATALINA_HOME: ./..
Using JAVA_HOME:     /opt/IBMJava2-ppc-13
```

7.2.4 Log files and solving problems

In the logs/ directory, you find the log files of Tomcat. Use them to detect errors. See 3.1.2, “Editing files using text editors” on page 88, if you need help on editing/viewing files.

For example, Tomcat uses port 8080 by default. However, when another program is already using this port number, Tomcat is not able to use that port number, and as a consequence, it will fail to startup. In our example, we could not connect to the server on port 8080. Look into the logs/catalina.out log file:

```
iserieslinux:/tmp/tomcat/jakarta-tomcat-4.0/logs # less catalina.out
```

You might see the following error:

```
Catalina.start: LifecycleException:  Error creating server socket:  java.net.Bin
LifecycleException:  Error creating server socket:  java.net.BindException: Addr
        at org.apache.catalina.connector.warp.WarpConnector.initialize(Unknw...
        at org.apache.catalina.core.StandardService.initialize(Unknown Source)
        at org.apache.catalina.core.StandardServer.initialize(Unknown Source)
        at org.apache.catalina.startup.Catalina.start(Unknown Source)
        at org.apache.catalina.startup.Catalina.execute(Unknown Source)
        at org.apache.catalina.startup.Catalina.process(Unknown Source)
        at java.lang.reflect.Method.invoke(Native Method)
        at org.apache.catalina.startup.Bootstrap.main(Unknown Source)
----- Root Cause -----
java.net.BindException: Address already in use
        at java.net.PlainSocketImpl.socketBind(Native Method)
        at java.net.PlainSocketImpl.bind(PlainSocketImpl.java:414)
        at java.net.ServerSocket.<init>(ServerSocket.java:182)
        at java.net.ServerSocket.<init>(ServerSocket.java:132)
        at org.apache.catalina.net.DefaultServerSocketFactory.createSocket(Unkno...
        at org.apache.catalina.connector.warp.WarpConnector.initialize(Unkno...
        at org.apache.catalina.core.StandardService.initialize(Unknown Source)
        at org.apache.catalina.core.StandardServer.initialize(Unknown Source)
        at org.apache.catalina.startup.Catalina.start(Unknown Source)
        at org.apache.catalina.startup.Catalina.execute(Unknown Source)
        at org.apache.catalina.startup.Catalina.process(Unknown Source)
```

This shows that the port number is no longer a problem. If you don’t see any error similar to this one, the problem might be somewhere else.

For example, try to set the browser so that it doesn’t use a proxy server. It might be that the proxy does not redirect connections to that server (possibly because it has an internal IP address when in test phase). Another possible cause could be a firewall that is blocking requests to your Tomcat server. (See your distributor’s documentation on how to disable this, or for more information on firewalls, refer to Chapter 8, “Firewall on iSeries Linux” on page 195.) For more information on how to solve problems, see the Tomcat home page at: http://jakarta.apache.org/tomcat/index.html

7.2.5 Changing the configuration

For detailed information about the configuration files, go to the Jakarta Web site at:
http://jakarta.apache.org/tomcat/tomcat-4.0-doc/config/index.html

As an example and to solve the common error discovered in 7.2.4, “Log files and solving problems” on page 192, modify the configuration file of Tomcat to set another portnumber. For example, change it to 8081 instead of the default 8080. (See 3.1.2, “Editing files using text editors” on page 88, if you need help on editing files.)

The conf/ directory contains the configuration files. Edit the server.xml file and change the following element to use another port number:

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
    <Connector className="org.apache.catalina.connector.http.HttpConnector"
               port="8080" minProcessors="5" maxProcessors="75"
               enableLookups="true" redirectPort="8443"
               acceptCount="10" debug="0" connectionTimeout="60000"/>
```

Modify 8080 to 8081. To find this element, search for the string 8080 in the configuration file:

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8081 -->
    <Connector className="org.apache.catalina.connector.http.HttpConnector"
               port="8081" minProcessors="5" maxProcessors="75"
               enableLookups="true" redirectPort="8443"
               acceptCount="10" debug="0" connectionTimeout="60000"/>
```

Now restart Tomcat with **startup.sh** (in case it’s still running, first stop it by running **shutdown.sh**), and try to connect again with your browser. For example, connect to http://10.5.92.90:*8081*/ (note the new port number 8081).

7.2.6 Default home page

If the Tomcat server is up and running, and you connect to it using your browser, you see the default Tomcat home page, as shown in Figure 7-1.

Figure 7-1 The default Tomcat home page

Everything you need to know to get started is explained on this home page. There are many examples of JSPs and servlets. There are links to the information you need to configure the server or to create JSPs and servlets.

7.3 Difference with IBM WebSphere Application Server

The main difference from the IBM WebSphere Application Server is that you don't have an Integrated Development Environment (IDE), and there is no built in support for the Java 2 Enterprise Edition (J2EE) containing Enterprise JavaBeans. However, you can download the J2EE from: http://java.sun.com

IBM WebSphere Application Server also contains very advanced caching mechanisms to handle high loads. Therefore, it is suited as a heavy duty Web application server.

8

Firewall on iSeries Linux

This chapter presents an overview of a firewall on iSeries Linux. Instead of giving a complete explanation of how the firewall in Linux has to be configured, this chapter discusses several approaches on how to implement a firewall using a Linux system running in an iSeries logical partition.

The topics in this chapter include:

- What a firewall is
- What a perimeter network is
- How the Linux firewall is configured
- Netfilter filtering basics
- Several Linux on iSeries firewall strategies
- Hosted or non-hosted partitions

8.1 Understanding the concept of a firewall

Typically you want to protect the computers in one network from possible attacks from malicious users in another network. Usually this other network is the Internet. However, you still want access from the internal network to the external network, for example to allow internal computers to browse the World Wide Web.

A firewall is basically a *packet filter function* with a logging facility, making it possible for a system to decide which users, applications, and systems should have access to predefined systems and their services. These decisions are made at the transport level (at the level of IP and TCP/UDP/ICMP), which is why we speak of a *packet filter*. Another type of firewall would be an application-level filter via proxy servers. Application-level filtering can be understood as content-aware filtering, while a packet filter primarily knows about connections, but not necessarily about contents. Often a packet filter is combined with application-level proxy servers, like *squid* (the popular HTTP and FTP proxy daemon) or SMTP for mail transfer.

Since the address space of the Internet has become crowded and IP addresses are scarce, corporate networks usually use IP addresses from a private address range internally. If intranet hosts need access to the Internet, they will be connected through a router doing network address translation (NAT). Such a router is ideally suited as a firewall.

You may also want to access some computers in your internal network from the Internet, for example, if you host a Web site. Usually, access of this kind is done in a perimeter network (also known as *demilitarized zone (DMZ)*). A *perimeter host* or network is a computer host or small network inserted as a neutral zone between a private network and an outside public network. Your servers that need to be accessible from the Internet, for example to host a Web site, are placed in this perimeter network, and only the services that the server provides are accessible. For example to host a Web site, we allow only incoming connections to the Web server on specific ports the Web server needs (normally 80 for HTTP and 443 for HTTPS).

The reason that you need a perimeter network and you don't want this Web server to be in you internal network is because, if someone breaks into the Web server, the computer criminals could make connections to all computers in your internal network and therefore, bypass the firewall. For this reason, perimeter networks are used, where servers that accept incoming connections from the Internet are isolated in the perimeter network.

8.1.1 References

Administering a firewall is not an easy job. Firewall administration involves advanced knowledge of networking. For this reason, firewalls and network security are usually outsourced to experienced professionals such as Linux distributors, Internet service providers (ISP), or companies that specialize in Linux security.

For a more in-depth explanation of firewalls in general, see the *Building Internet Firewalls, 2nd Edition* by Elizabeth D. Zwicky, et al.

For iSeries-specific security issues, see *AS/400 Internet Security Scenarios: A Practical Approach,* SG24-5954.

If you are migrating from IBM Firewall for AS/400 (5769-FW1), we recommend that you study the IBM Redbook *All You Need to Know When Migrating from IBM Firewall for AS/400,* SG24-6152. This redbook describes what information to collect before doing a migration.

8.2 Bringing Linux to the iSeries server

Bringing Linux to the iSeries server gives customers another way to replace the discontinued IBM Firewall for AS/400 (5769-FW1). Until now, IBM advised customers to replace the IBM Firewall for AS/400 with a firewall on the Integrated xSeries Server for iSeries running Windows 2000. However, most firewall solutions using that strategy are very expensive.

Linux on iSeries is a honorable alternative. It provides you with a full strength firewall for a very reasonable price (the price of a Linux distribution).

8.3 Firewall on Linux

The firewall in Linux is built into the kernel since version 1.0. It is configured through the command line to add, replace, or delete filter rules on the firewall. The firewall architecture has changed throughout the years, but the mechanism of configuring the firewall has always been the same.

For more information on firewall and security on Linux, visit: http://www.linuxsecurity.com/

More specifically, you can refer to the following site for firewall information: http://www.linuxsecurity.com/resources/firewalls-1.html

You may also want to refer to the following resources:

- Firewall and Proxy Server HOWTO: http://www.linuxdoc.org/HOWTO/Firewall-HOWTO.html
- Linux IP Masquerade HOWTO: http://www.linuxdoc.org/HOWTO/IP-Masquerade-HOWTO.html
- Linux IPCHAINS HOWTO: http://www.linuxdoc.org/HOWTO/IPCHAINS-HOWTO.html
- The collection of Networking HOWTOs for an overview of all networking related HOWTOs: http://www.linuxdoc.org/HOWTO/HOWTO-INDEX/networking.html

8.3.1 iptables

The most recent firewall architecture is called *Netfilter*. It is included in the Linux kernel version 2.4, and the command to configure it is **`iptables`**. Since it's fairly new, not too many people are using it in productive use because it still needs to prove its stability.

The iptables framework offers:

- Packet filtering
- Extensive network address translation allowing you to change source IP address, destination IP address, source port, destination port, MAC address, Time To Live (TTL), etc.
- Logging facilities
- Stateful inspection/connection tracking

Other firewall functions like virtual private network (VPN), socks server, and mail relay are available on Linux through software, separate from iptables.

The Netfilter home page is on the Web at: http://netfilter.filewatcher.org/

8.3.2 ipchains

The **ipchains** architecture is the older, but still most popular, Linux firewall that was introduced with version 2.2 kernels. Most distributors install it by default and provide some tools to configure the firewall. It is possible to use the **ipchains** command with a version 2.4 kernel, but then you need to make sure the ipchains module is loaded. You cannot use the **ipchains** and **iptables** commands at the same time, because either the ipchains *or* the iptables modules can be loaded into the kernel (they're mutually exclusive).

8.3.3 ipfwadm

ipfwadm is the oldest firewall architecture. Just like with **ipchains**, you can still use this command in a version 2.4 kernel, but then you will not be able to use the new functionality that the Netfilter architecture provides.

8.3.4 The kernel

The architecture of the firewall is one part of the kernel. It determines the capabilities and available features. But the rest of the kernel is even more important, because the firewall is based on it. Only a proven kernel with a robust TCP/IP stack will give you security.

There are different versions of the Linux kernel. With each new version, new features are added, thereby introducing new bugs. Kernels with an even minor version number are considered to be "stable", while odd minor numbers indicate a development version.

The 2.2 kernel is considered to be very stable and development is finished (the current version being 2.2.20). The 2.4 kernel was first released early in the year 2001 and currently gets the finishing touches. Just as it took the initially released 2.2 kernel a couple of more bugfix releases to become really ripe and mature, there are still some issues to be solved with the 2.4 kernel. (The reason why open source software is released at the "pretty much stable" stage is that, at a certain point in time, broad testing in the field is most effective.) In addition to the mainstream development, those changes need to be synchronized with the porting effort to the iSeries platform.

For a firewall, the trust in the kernel you use will be based either on your own judgement, or you will trust the distributor that supplies the kernel to you. To judge yourself, you need a solid understanding of the kernel, a close look on ongoing development, maybe insider information about pending vulnerabilities, and a test environment. The 2.4 kernel is already in a good state, and it is expected to become mature soon. We recommend that you check with your distributor before you employ your firewall in productive use.

8.4 Netfilter filtering basics

When Linux is installed on the iSeries server, it will always be kernel version 2.4 or newer, because the 2.4 kernel was the first one to be ported to that platform. This implies that Netfilter support is normally included into the kernel. Depending on the distribution, there might already be a firewall active.

A firewall in Linux is typically configured in a shell script. This script contains on each line the **iptables** or **ipchains** commands in a particular order. These commands set up the filtering rules. This is done with a script because rules issued to the kernel using **iptables** or **ipchains** are not persistent across IPLs, and normally this script is executed at boot time.

Figure 8-1 shows you the basics of how an incoming packet traverses your firewall rules in the new Netfilter framework.

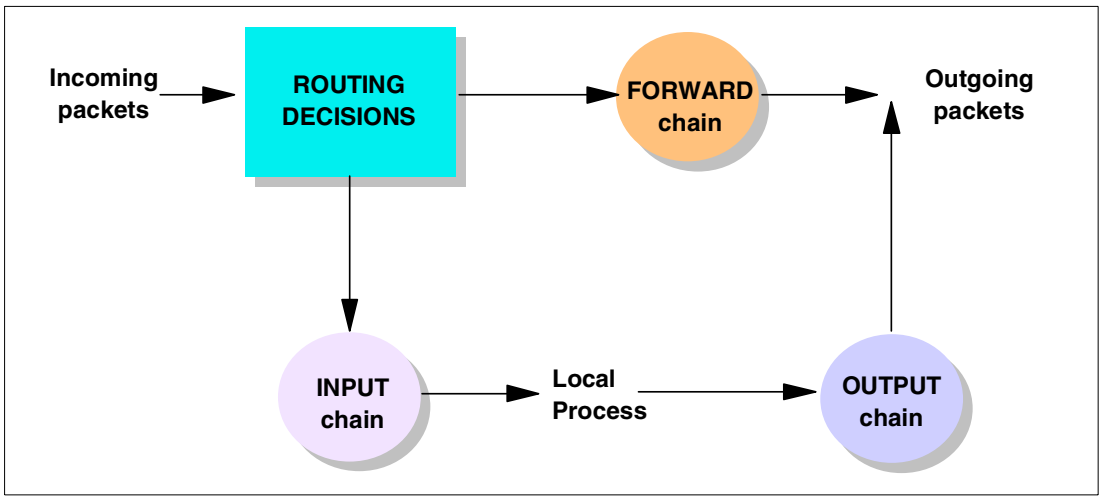


Figure 8-1 INPUT and FORWARD and OUTPUT chains

First a routing decision is made. If the packet is destined for your host, it is sent to the INPUT chain. If it is destined for a host in the local network, it is sent to the FORWARD chain. When your firewall replies to packets or just sends packets, they traverse the OUTPUT chain before being sent out.

You see the different channels are not connected. They are all independent. This used to be different with ipchains, where the chains were connected after each other. This means that when a packet needed to be forwarded, it first went into the input chain, then through the forward chain, and finally into the output chain. This used to be much harder to configure, and some security holes were discovered with this approach.

8.5 Performance

Usually, a firewall does not need a lot of processing power. The amount of processing power depends on the firewall strategy and the amount of traffic that needs to pass through the firewall. If you need to do a lot of logging or want to inspect the contents of the packets, then you certainly need more processing power.

An enormous advantage with Linux on iSeries is that the resources allocated to Linux can be easily adjusted. You can start with a moderate amount of processor and memory, and then increase or decrease the resources as needed. Also, disk space can be easily increased by adding virtual disks. For information on how to allocate resources, see 2.3, “LPAR considerations” on page 20.

8.6 Linux on iSeries firewall strategies

Several firewall strategies are discussed in this section that are typical for Linux on iSeries. There is one important difference compared to Linux on normal machines. To connect the Linux partition to a network, you can generally choose between native LAN adapters (direct attached LAN adapters) or virtual LAN adapters. The virtual LAN concept is used in all of the following examples.

8.6.1 Native LAN adapter requirement

There are 16 available virtual LAN channels to connect logical partitions, so there is a high level of flexibility. However, to connect the firewall to an external network directly, you must have at least one native LAN card.

Without a native LAN card, incoming traffic is routed by an OS/400 system to the Linux partition though a virtual LAN. This OS/400 partition is consequently not protected by the firewall, because it needs to forward all traffic to the Linux partition.

Ideally, the Linux partition should not have direct dependency on OS/400 TCP/IP. If virtual LAN is used, then the Linux partition may become isolated if it relies on the OS/400 partition to route packets. But for native adapters, Linux TCP/IP communication is completely separate from OS/400 TCP/IP. OS/400 TCP/IP can be stopped and started without affecting Linux.

8.6.2 Basic configuration

Figure 8-2 shows a basic configuration of a Linux firewall.

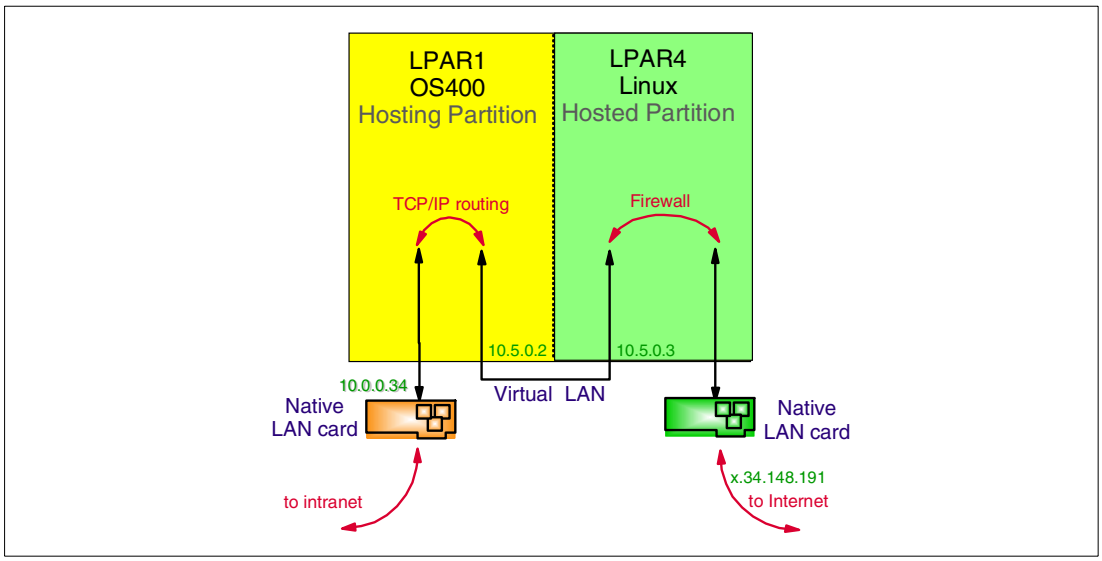


Figure 8-2 Basic configuration of Linux firewall

The Linux partition has one native LAN adapter connected to the Internet. Another native LAN adapter that belongs to the OS/400 partition is connected to the internal network (intranet). The internal network could access the Internet if needed, because packets could be routed through the OS/400 partitions LAN adapter and through the virtual LAN to the Linux partition. This Linux partition can then do simple routing, NAT, run a proxy to allow restricted access to the Internet, and of course run a packet filter.

In this example, all internal hosts have an IP address in a private address range (starting with 10.x.x.x). These address ranges cannot be addressed from the Internet. Consequently, no internal host could run a publicly available network service, such as a Web server, unless destination network address translation (DNAT) is used on the firewall. This is not really recommended because we do not have a perimeter network (as briefly explained in 8.1, “Understanding the concept of a firewall” on page 196).

8.6.3 Perimeter network for logical partitions

If we extend Figure 8-2 with a perimeter network, we see a situation like the one shown in Figure 8-3. The rest of the concept stays the same. You have one native LAN card for the Linux partition that runs the firewall connected to the Internet and one native LAN card for OS/400 that is connected to the intranet. The perimeter network is running inside of the iSeries box, and it's not possible to connect other servers to that perimeter network. Only partitions in between LPAR2 and LPAR4 (that is, the systems that use the virtual LAN connecting LPAR2, LPAR3, and LPAR4) are connected to the perimeter network.

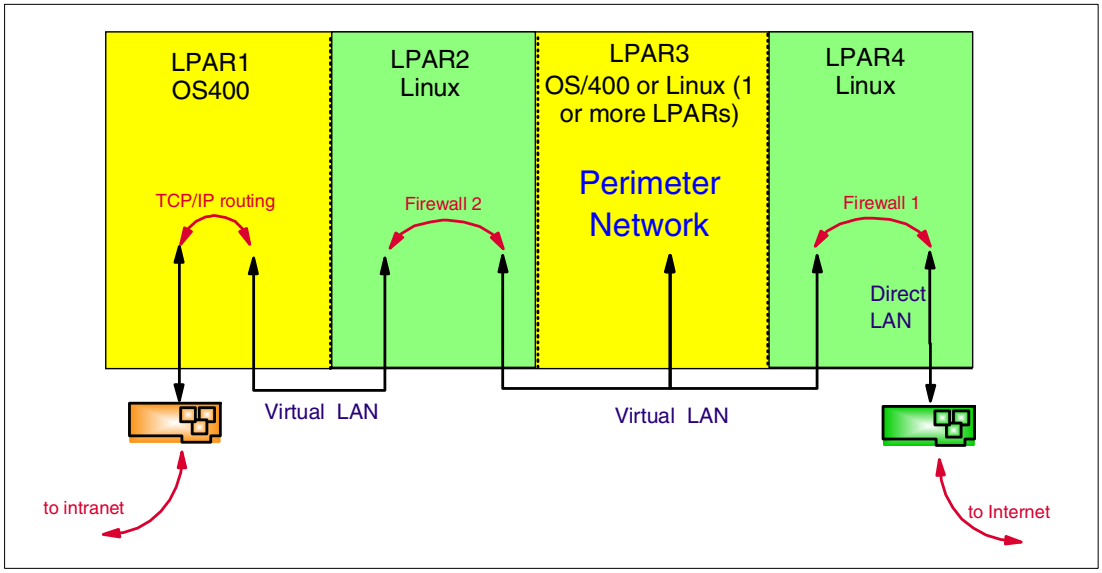


Figure 8-3 A basic configuration with a perimeter network

8.6.4 Perimeter network for other hosts

The firewall scenario explained in Figure 8-3 can be extended again to provide a perimeter network for hosts other than a logical partition in the iSeries box. One of the approaches we can take is illustrated in Figure 8-4.

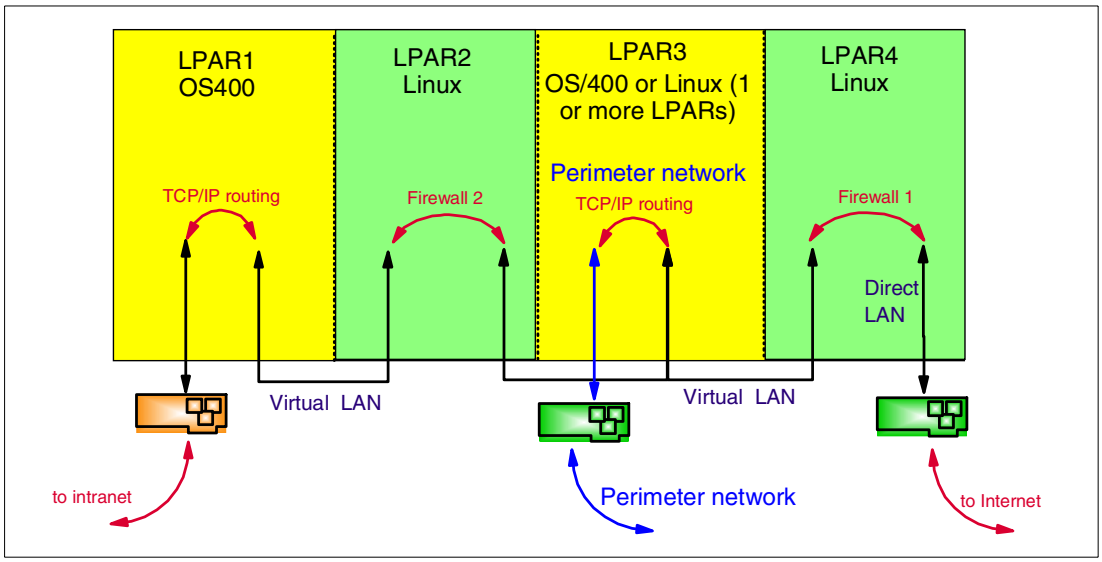


Figure 8-4 Firewall configuration with a perimeter network for other hosts (not recommended)

This is probably not a desirable approach because you need an extra partition to do the routing. If LPAR3 is needed to run a Web server, for example, it might not be a good idea to let that partition do the routing to other partitions too. Therefore, if you need a perimeter network for other hosts, it's probably better to add two native LAN cards to isolate routing and network access in the firewalls only and not to rely on LPAR3 to do extra routing. You can find an improved approach in Figure 8-5.

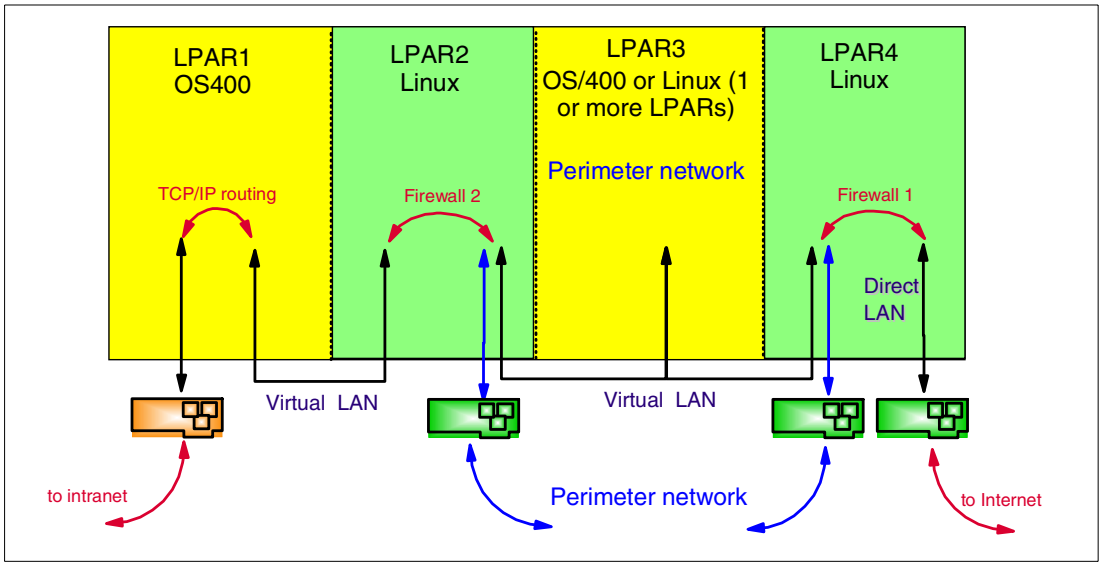


Figure 8-5 Firewall configuration with a perimeter network for other hosts (recommended)

8.6.5 OS/400 partition under control of firewall with a perimeter network

The previous examples all discussed situations where the internal network is connected to the Internet via the native LAN adapter in the OS/400 partition. This involves some setup in the OS/400 partition too, regarding TCP/IP routing etc. It also has some disadvantages. One such disadvantage is that the internal network cannot connect to the Internet if OS/400 TCP/IP is not activated. It is better to bundle the network routing and security in the firewalls.

An optimal situation is shown in Figure 8-6. To access the OS/400 partition LPAR1, a user first needs to pass though the firewall, so even traffic from and to hosts of the intranet can be screened. Therefore, it would be possible to restrict access to the OS/400 partition or log on connections to the OS/400 partition.

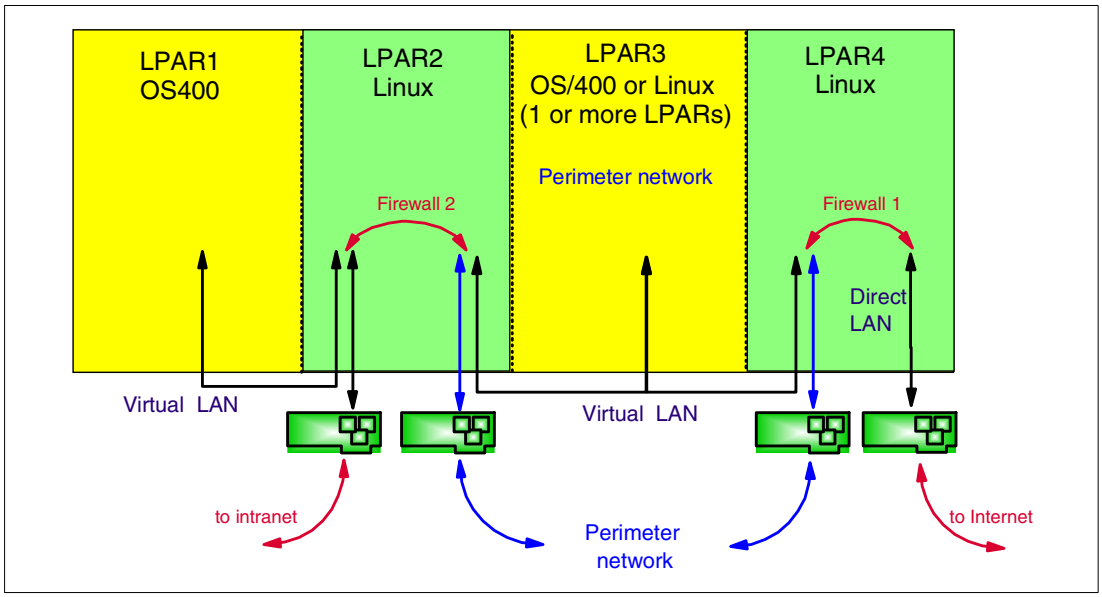


Figure 8-6 A firewall with a perimeter network and two Linux partitions

8.6.6 OS/400 partition under control of firewall without a perimeter network

For small systems or networks, Figure 8-6 can be simplified if a perimeter network is not needed. This topology is similar to Figure 8-2 on page 200. The only difference in Figure 8-7 compared to Figure 8-2 is that the OS/400 partition can now be controlled by the firewall, and routing is done in the Linux firewall, and no longer in Linux and OS/400.

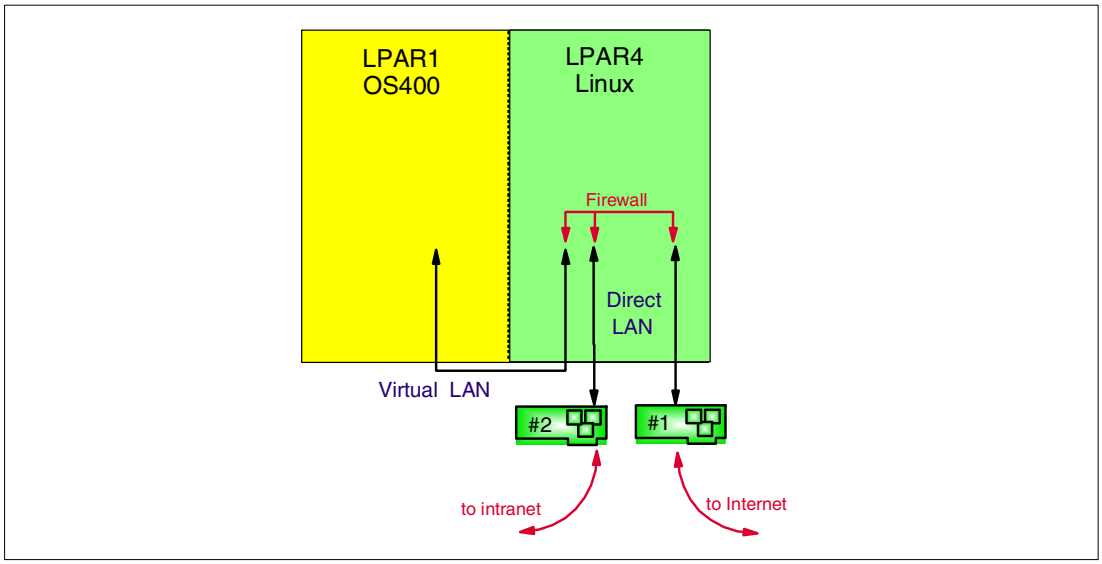


Figure 8-7 A firewall without perimeter network

8.7 Hosted or non-hosted partitions

It's not safe to run a firewall on the same system that you use to run other applications. These tasks should be separated into different systems, because if the firewall is broken into, the other applications can be changed or, more importantly, data from those partitions can be read. On the other side, if someone can break into the application (like a Web server for example), they could tamper with the firewall.

Another reason for separation is related to denial of service (DoS) attacks. If the firewall is running on the same system as a business application, and the firewall is extremely loaded, the business application would stop running.

When running a firewall and a business application in two separate logical partitions on one iSeries server, they are actually running on the same hardware box, but the LPAR implementation is done so that partitions cannot influence each other. For example, if the partition running the firewall would have 25% of a processor assigned to it, and the partition running the business application would have 75% of that same processor, there is no possible way that the Linux partition could take more than 25% of that processor, even if the Linux partition totally crashed.

However, there is one small way the Linux firewall could add some load to the OS/400 partition. If the Linux partition is a hosted partition, and it is using virtual DASD, the hosting partition will have to perform all the read and write instructions the Linux partition requests. Consequently, the Linux partition can use some resources of the hosting OS/400 partition.

To completely isolate the Linux firewall from OS/400 partitioning, the Linux firewall partition should be a non-hosted partition. However, a DoS attack will mostly effect the processor usage and the memory usage of the host being attacked. These resources are strictly limited for logical partitions, unless a huge amount of logfile entries is written to virtual disk.

This problem could perhaps be solved by putting the virtual disks on a separate auxiliary storage space (ASP), so this would not occupy the hard disks that the other applications in the OS/400 partition use.

Again, you could run the firewall in a non-hosted partition, but then you lose all of the flexibility (backup and recovery, changing amount of disk space), and the price of the hardware will be much higher.

8.8 The iptables example

As stated in 8.3.1, "iptables" on page 197, iptables has not yet proven its stability (at the time of this writing). However, because it's the new approach to firewalls on Linux, an example of the usage of iptables is presented here.

Please understand that this example is purely educational and has not undergone any security tests. It's only purpose is to serve as an example of what iptables can be used for and its usage and syntax. This script should work on all Linux distributions, as long as they have the appropriate modules ready to be loaded in the kernel.

Information on what is being done in the script is given with inline comments. The topology used in this example is similar to what is shown in Figure 8-2 on page 200. By executing this script, you configure the firewall and the firewall sets up filter tables. Whenever a packet comes in, it is matched against these filter rules, and the first rule that is matched is taken

(and all following rules are not evaluated anymore). When reviewing these rules, it is important to remember how packets traverse the chains in iptables. See 8.4, “Netfilter filtering basics” on page 198, for this overview. Also, for a general overview of the syntax, see the man page on iptables (**man iptables**).

```
#!/bin/sh
echo "Starting firewall..."
#*******************************************************************************************
# Define variables for things that may be variable among systems. These variables are
# defined here so it's easy to change them afterwards without having to change the whole
# script.
#
# veth0 is the virtual ethernet adapter, connected to the internal network
# eth0 is the native ethernet adapter, connected to the Internet
# The outside IP address is 217.81.212.163
# The address of the local network our is located in is 10.5.0.0 with network mask
#     255.255.255.0
#*******************************************************************************************
INSIDE="veth0"
OUTSIDE="eth0"
LOOPBACK="lo"
OUTSIDE_IP_ADDRESS="217.81.212.163"
LAN="10.5.0.0/255.255.255.0"

#*******************************************************************************************
# In most distributions, ipchains is currently used as a default. As we will work with
# iptables, we try to remove the ipchains module from the kernel. Otherwise it will cause
# problems with the iptables module.
#*******************************************************************************************
rmmod ipchains

#*******************************************************************************************
# Insert needed modules for iptables and connection-tracking
#
# Note:  This is not needed if if the modules are built into the kernel.
#
# Note:  The modules are located in /lib/modules/2.4.3/kernel/net/ipv4/netfilter
#  replace the '2.4.3' with whatever kernel version you have
#*******************************************************************************************
echo "Loading the needed modules..."

# for general iptables
insmod ip_tables

# for connection tracking
insmod ip_conntrack

# for connection tracking of FTP

insmod ip_conntrack_ftp

# for doing SNAT and DNAT (Source Network Address Translation and Destination Network
# Address Translation)

insmod iptable_nat

# for general iptables rules
insmod iptable_filter

# for logging
```

```
insmod ipt_LOG

#*****************************************************************************************
# for IP masquerading (As we will not do IP masquerading here, we don't insert the module.)
#
# IP masquerading is typically used if the outside IP address of the firewall is changing
# (for example if it is retrieved from a DHCP server, as many cable Internet providers do).
#*****************************************************************************************
# insmod ipt_MASQUERADE


#*****************************************************************************************
# Enable IP forwarding by putting a value of '1' in the appropriate pseudo file in
# the /proc filesystem.
#*****************************************************************************************
echo "Enabling IP forwarding..."
echo "1" > /proc/sys/net/ipv4/ip_forward


#*****************************************************************************************
# Set default policies ('-P' option) for the INPUT, FORWARD, and OUTPUT chains
# It might not be a good idea to have the OUTPUT chain policy set to 'ACCEPT'.  You should
# probably set this to 'DROP' and then add rules for all outgoing traffic to be more
# secure.
#
# Instead of DROP, we could also specify REJECT. REJECT has the same effect as DROP,
# except that the sender is sent an ICMP 'port unreachable' error message so he knows that
# the destination received the packet. With DROP, it seems for the sender that they just
# disappeared.
#*****************************************************************************************
echo "Loading all the rules..."
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

#*****************************************************************************************
# Perform Source Network Address Translation (SNAT) on all packets going out the 'outside'
# interface. The source packets will get SNAT'd to the IP address specified in the variable
# OUTSIDE_IP_ADDRESS.
#
# Note that the firewall will perform connection tracking and automatically 'reverse NAT'
# on all response packets.
#*****************************************************************************************
iptables -t nat -A POSTROUTING -o $OUTSIDE -j SNAT --to-source $OUTSIDE_IP_ADDRESS


#*****************************************************************************************
# In the FORWARD chain, accept packets that establish new connections if they came from
# the inside interface. This will allow clients on the inside to establish new connections
# to the outside.
#*****************************************************************************************
iptables -A FORWARD -m state --state NEW -i $INSIDE -j ACCEPT


#*****************************************************************************************
# In the FORWARD chain, accept all traffic that is part of an established connection.
# This allows responses to connections that our inside clients initiated.
#*****************************************************************************************
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
#*****************************************************************************************
# In the INPUT chain, accept all traffic that is part of an established connection.  This
# allows responses to connections that the linux firewall initiated.
# Note that our OUTPUT chain policy is ACCEPT, so outbound connections will already be
# allowed and this rule will allow responses.
#
# If you don't want the linux firewall itself to be able to communicate with the outside
# world just comment this line out.
#*****************************************************************************************
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT


#*****************************************************************************************
# In the INPUT chain, accept all from localhost
#*****************************************************************************************
iptables -A INPUT -i $LOOPBACK -j ACCEPT


#*****************************************************************************************
# In the INPUT chain, accept all TCP, UDP, and ICMP traffic from the inside LAN IP address
# to the Linux firewall if it came in on the inside interface.
#
# If you do not want your internal clients to communicate with the firewall, comment these
# lines out. Normally, you would only accept connections to the firewall from one inside IP
# address in order to avoid people from the internal network 'tampering' with the firewall.
#
# In this example, we trust the internal hosts.
#*****************************************************************************************
iptables -A INPUT -p TCP -s $LAN -i $INSIDE -j ACCEPT
iptables -A INPUT -p UDP -s $LAN -i $INSIDE -j ACCEPT
iptables -A INPUT -p ICMP -i $INSIDE -j ACCEPT


#*****************************************************************************************
# As we will accept all packets from the internal hosts, the following rules are only being
# applied on packets from external hosts. The packets from the internal host will not reach
# this far in the firewall rules because they already mached rules above.
#
#
# In the INPUT chain, accept TCP port 22(ssh) traffic from anyone.
# This rule would allow ssh to the linux firewall from the outside. If you do not wish to
# allow remote administration from the outside, comment this line out.
#*****************************************************************************************
iptables -A INPUT -p TCP -s 0/0 --dport 22 --sport 1024:65535 -j ACCEPT


#*****************************************************************************************
# Allow HTTP (port 80) and HTTPS(port 443) from external hosts. This would allow access to
# a webserver running on the Linux firewall. (We disabled this there.)
#*****************************************************************************************
# iptables -A INPUT -p TCP -s 0/0 --dport 80 --sport 1024:65535 -j ACCEPT
# iptables -A INPUT -p TCP -s 0/0 --dport 443 --sport 1024:65535 -j ACCEPT


#*****************************************************************************************
# Allow DNS queries (port 53) from external hosts. This would be needed if running a DNS
# server on the Linux firewall. (We disabled it here.)
#*****************************************************************************************
# iptables -A INPUT -p UDP -s 0/0 --dport 53 --sport 1024:65535 -j ACCEPT
```

```
#*****************************************************************************************
# Allow FTP (port 21) from external hosts.
#*****************************************************************************************
iptables -A INPUT -p TCP -s 0/0 --dport 21 --sport 1024:65535 -j ACCEPT


#*****************************************************************************************
# These ICMP packets are used to inform hosts of the status of the network. In the INPUT
# chain, accept ICMP types 0, 3, and 5.
# This allows ICMP echo reply(ping reply), ICMP redirect, and ICMP destination unreachable
# respectively.
#*****************************************************************************************
iptables -A INPUT -p ICMP --icmp-type 0 -j ACCEPT
iptables -A INPUT -p ICMP --icmp-type 3 -j ACCEPT
iptables -A INPUT -p ICMP --icmp-type 5 -j ACCEPT


#*****************************************************************************************
# In the INPUT chain, accept ICMP type 8. This allows ICMP echo (ping request) and allows
# the outside interface of the firewall to be 'pinged'. Comment this out if you don't want
# the outside to be able to ping the linux firewall.
#*****************************************************************************************
iptables -A INPUT -i $OUTSIDE -p ICMP --icmp-type 8 -j ACCEPT


#*****************************************************************************************
# We will do logging afterwards. However, before logging, we will drop the following UDP
# packets because they appear so much in a regular network that they would fill up the logs
# with unimportant information:
#
#     NetBIOS (port 137 and 138)
#     BOOTP and DHCP (port 67)
#     Routing Information Protocol (520)
#
#*****************************************************************************************
iptables -A INPUT -p UDP --dport 520 -j DROP
iptables -A INPUT -p UDP --dport 137 -j DROP
iptables -A INPUT -p UDP --dport 138 -j DROP
iptables -A INPUT -p UDP --dport 67 -j DROP

#*****************************************************************************************
# Everything that hasn't matched any rules above will be logged here.  Right after logging,
# we will drop them immediately.
#*****************************************************************************************
iptables -A INPUT -j LOG --log-prefix ' ## DROPPED (INPUT) ## '
iptables -A FORWARD -j LOG --log-prefix ' ## DROPPED (FOWARD) ## '

#*****************************************************************************************
# Now drop the packets we just logged. This does not need to be done explicitly as our
# default policy is already DROP. All packets that haven't matched a rule up to this point
# (except for the log rules_ will be dropped anyway.
#*****************************************************************************************
# iptables -A INPUT -j DROP
# iptables -A FORWARD -j DROP

echo "Done setting up firewall."
```

9

ssh and Telnet

This chapter provides an overview about ssh and Telnet. Both protocols are used to remotely login to other machines. ssh also allows the remote execution of commands. In addition, the **scp** and **sftp** commands, which are based on ssh, transfer files in a secure fashion.

The following topics are discussed:

- What is a remote login
- ssh (secure shell)
- scp (secure copy)
- sftp (secure FTP)
- Telnet

Each section begins by explaining how things work in Linux. Then, each section ends by mentioning the available options or alternatives for Microsoft Windows users.

9.1 Remote login

Remote login means the interactive (login) use of a computer over the network. The remote machine could be either next door or thousands of miles away. It can also be right under your desk, but if you don't use the keyboard or monitor of the machine itself you would still have to login over the network. In case of the iSeries, there is no monitor, keyboard or terminal directly attached to the virtual Linux machine; thus, you will always have to login remotely to work with these virtual machines.

For example, when you login in to work from home to check your mail, you are using ssh or Telnet to connect from one computer (your computer) to another computer (work's computer). Once you have established your connection, you then log in to that computer and execute commands remotely on that computer through your ssh or Telnet interface. Most often, you are connecting to a UNIX/Linux based system. Therefore, the commands you use, such as **`ls`**, **`cd`**, **`pine`**, **`talk`**, and **`rm`**, are UNIX/Linux commands, and not ssh or Telnet commands.

There are different means to login remotely: ssh and Telnet. Both are described in this chapter.

9.2 ssh (secure shell)

ssh is short for secure shell. In fact, ssh does not provide a shell at all. Instead, it allows you to access a shell. It also provides a means to run commands remotely in a secure way.

Traditionally, Telnet (see below) has been used to login into other machines over the network. In addition, the so-called r-commands (r for remote: rlogin, rsh, rcp) were widely used to login or run remote commands in a convenient way. These r-commands are completely replaced by login, ssh, and scp.

The SSH protocol encrypts all traffic (including passwords) to effectively eliminate eavesdropping, connection hijacking, and other network-level attacks. In addition, SSH provides a myriad of secure tunneling capabilities, as well as a variety of authentication methods.

Why there are two protocols: SSH1 and SSH2

There are several implementations. You can find a thorough discussion of them in the book *SSH, The Secure Shell, the Definitive Guide*, by Daniel Barrett and Richard Silverman.

OpenSSH

Later, the SSH2 license become loose; it now allows for free use under certain operating systems (Linux, NetBSD, FreeBSD, and OpenBSD). Therefore, free implementations of the protocol suite have started to appear, notably OpenSSH (`http://www.openssh.com/`).

The OpenSSH suite includes the ssh program, which replaces rlogin and telnet; scp, which replaces rcp; and sftp, which replaces ftp. Also included is sshd (the ssh daemon, which is the server side of the package) and other basic utilities like ssh-add, ssh-agent, ssh-keygen, and sftp-server. OpenSSH supports SSH protocol versions 1.3, 1.5, and 2.0, as server and as client.

OpenSSH is primarily developed by the OpenBSD Project and is freely usable and re-usable by everyone under the BSD license.

9.2.1 Features

You can learn about the features of SSH on the OpenSSH Web site. You can find this site at:
http://www.openssh.com/features.html

9.2.2 How it works

When you enter **ssh hostname** to connect to some remote host, the ssh client opens a TCP/IP connection to the ssh port (22) of the server *hostname*. Beforehand, it reads the system wide configuration file /etc/ssh/ssh_config and the user's configuration file ~/.ssh/config, and interprets options given on the command line.

Let's follow an example while we talk through the process, using the -v (verbose) switch to make visible what's happening:

```
[mac@M03H mac]$ id
uid=500(mac) gid=501(mac) groups=501(mac)
```

We are logged in as user *mac* at the host *m03h*.

```
[mac@M03H mac]$ ssh -v m01c
OpenSSH_2.9p1, SSH protocols 1.5/2.0, OpenSSL 0x0090600f
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: Seeding random number generator
debug1: Rhosts Authentication disabled, originating port will not be trusted.
debug1: restore_uid
debug1: ssh_connect: getuid 500 geteuid 0 anon 1
debug1: Connecting to m01c [10.10.5.90] port 22.
debug1: temporarily_use_uid: 500/501 (e=0)
debug1: restore_uid
debug1: temporarily_use_uid: 500/501 (e=0)
debug1: restore_uid
debug1: Connection established.
debug1: read PEM private key done: type DSA
debug1: read PEM private key done: type RSA
debug1: identity file /home/mac/.ssh/identity type -1
debug1: identity file /home/mac/.ssh/id_rsa type -1
debug1: identity file /home/mac/.ssh/id_dsa type -1
debug1: Remote protocol version 1.99, remote software version OpenSSH_2.5.1p1
debug1: match: OpenSSH_2.5.1p1 pat ^OpenSSH_2\.5\.[01]p1
Enabling compatibility mode for protocol 2.0
debug1: Local version string SSH-2.0-OpenSSH_2.9p1
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug1: kex: server->client 3des-cbc hmac-md5 none
debug1: kex: client->server 3des-cbc hmac-md5 none
debug1: SSH2_MSG_KEX_DH_GEX_REQUEST_OLD sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_GROUP
debug1: dh_gen_key: priv key bits set: 202/384
debug1: bits set: 1015/2049
debug1: SSH2_MSG_KEX_DH_GEX_INIT sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_REPLY
The authenticity of host 'm01c (10.10.5.90)' can't be established.
DSA key fingerprint is d9:a0:ad:ee:16:66:26:e2:e6:1a:12:29:d0:2a:af:c5.
Are you sure you want to continue connecting (yes/no)?
```

It is the first time that we try ssh login to the host m01c. Therefore, our ssh client cannot find its host key in the database (~/.ssh/known_hosts). If we decide to connect, the key will be put there. Next time the key that the server offers us can be compared with the stored version. This allows us to recognize whether the host we connect to is really the one that we mean.

At this point, we could also call the system administrator of the remote machine and ask them for the fingerprint of the DSA host key of the machine, to compare with. They would run the command:

```
ssh-keygen -l -f /etc/ssh/ssh_host_dsa_key
```

Then they would read out the fingerprint of the key for us. If the fingerprints match, we have very strong indication that we are talking to the right machine.

We reply with “yes”:

```
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'm01c,10.10.5.90' (DSA) to the list of known hosts.
```

The first thing the client and server do, after finding out which versions each other understand, is to establish an encrypted connection, to be used for all further communication. The client computes a new session key that will be used by client and server for the encryption. To transfer the key to the server securely (the encrypted connection is not yet established!), the client encrypts it twice, with the servers public host key and with the server key, thereby making sure that only the server can read it. (The server key is periodically generated from scratch by the ssh daemon.) After that, both sides can turn on encryption.

```
debug1: bits set: 1020/2049
debug1: len 55 datafellows 53248
debug1: ssh_dss_verify: signature correct
debug1: kex_derive_keys
debug1: newkeys: mode 1
debug1: SSH2_MSG_NEWKEYS sent
debug1: waiting for SSH2_MSG_NEWKEYS
debug1: newkeys: mode 0
debug1: SSH2_MSG_NEWKEYS received
debug1: done: ssh_kex2.
debug1: send SSH2_MSG_SERVICE_REQUEST
debug1: service_accept: ssh-userauth
debug1: got SSH2_MSG_SERVICE_ACCEPT
debug1: authentications that can continue: publickey,password,keyboard-interactive
debug1: next auth method to try is publickey
```

Our ssh client prefers to authenticate us using a public key mechanism. It looks for key files in our .ssh directory:

```
debug1: try privkey: /home/mac/.ssh/identity
debug1: try privkey: /home/mac/.ssh/id_rsa
debug1: try privkey: /home/mac/.ssh/id_dsa
```

But no key files can be found, because we did not create any key so far. (We do that in the next example.) The next available method is to ask the user for their system password on the remote machine:

```
debug1: next auth method to try is password
mac@m01c's password:                                          <<< enter password
debug1: ssh-userauth2 successful: method password
```

Since we entered the password correctly, a new session is started for us on the remote machine, and a login shell started, so we finally see the shell prompt:

```
debug1: channel 0: new [client-session]
debug1: channel_new: 0
debug1: send channel open 0
debug1: Entering interactive session.
debug1: client_init id 0 arg 0
debug1: channel request 0: shell
```

```
debug1: channel 0: open confirm rwindow 0 rmax 16384
SuSE Linux on iSeries -- the spicy solution!
Have a lot of fun...
mac@m01c:~ >
```

9.2.3 Creating a key

A *key* is a digital proof of your identity. Keys have some advantages over system passwords:

- The private key file is stored on the local machine, while a system password would be stored on the remote machine. In a setting where the local machine is much less secure than the remote machine, this does not turn into an disadvantage, because you can still use password authentication because you are not forced to store a key file anywhere.
- If you happen to enter your password while trying to login into a compromised host, even though the password cannot be sniffed over the network, it can be captured on the remote host. A key never leaves the client host, making it more difficult to steal.
- If the remote machine is compromised and your password becomes known to someone, you better have different passwords on all machines that you access. But using different passwords all the time and changing them periodically is difficult in a real world.
- It is easier to share key files than to share passwords, if an account is to be used by more than one person.
- With UNIX passwords, as they are usually used, only the first eight characters are significant. Key files are much more resistant to brute force attacks, because they are much longer.

A key is actually a pair of keys – a private part and a public part. The *public key* can be given to untrusted people/hosts, while the *private key* must be kept secret. To further protect it, it is again encrypted with a passphrase. The public key can be used to encrypt data, which can be decrypted *only* with the private key. This is called an *asymmetrical method*, and it is relies on the assumption that the private key cannot be derived from the public key. This is made possible by a so-called trapdoor function, with the properties that computation in one direction (encryption) is easy and in the other is virtually impossible (attack). See the cryptography FAQ for more information: http://www.faqs.org/faqs/cryptography-faq/part06/

You can create a key pair with the **ssh-keygen** command. By default, an SSH1 key is created (remember, there are different versions). For simplicity and compatibility, we stick to the default in this example:

```
[mac@MO3H mac]$ ssh-keygen
Generating public/private rsa1 key pair.
Enter file in which to save the key (/home/mac/.ssh/identity):      <<< press Enter
Enter passphrase (empty for no passphrase):                            <<< enter passphrase
Enter same passphrase again:                                           <<< again
Your identification has been saved in /home/mac/.ssh/identity.
Your public key has been saved in /home/mac/.ssh/identity.pub.
The key fingerprint is:
ed:53:c6:32:0f:08:60:39:95:9e:51:bc:07:64:bc:75 mac@MO3H.itso.ibm.com
[mac@MO3H mac]$
```

Now the key pair is in our .ssh directory. The private key (in the file identity) is not readable for anyone other than user *mac*. The public key (in the file identity.pub) is ready to be copied to remote servers:

```
[mac@MO3H mac]$ ls -l .ssh
total 12
-rw-------    1 mac   mac        539 Oct 15 16:47 identity
-rw-r--r--    1 mac   mac        343 Oct 15 16:47 identity.pub
```

```
-rw-r--r--    1 mac   mac        604 Oct 15 14:53 known_hosts2
[mac@M03H mac]$
```

ssh-keygen defaults to generating a RSA1 key for use by SSH protocol version 1. Specifying the -t option allows you to create a key for use by SSH protocol version 2. We also create such a key here:

```
[mac@M03H mac]$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/mac/.ssh/id_dsa):        <<< press Enter
Enter passphrase (empty for no passphrase):                           <<< enter passphrase
Enter same passphrase again:                                          <<< again
Your identification has been saved in /home/mac/.ssh/id_dsa.
Your public key has been saved in /home/mac/.ssh/id_dsa.pub.
The key fingerprint is:
7a:f3:36:a9:ed:c8:a4:ab:b6:f5:ae:fd:8b:4e:37:12 mac@M03H.itso.ibm.com
[mac@M03H mac]$
```

So the .ssh directory now looks like this:

```
[mac@M03H mac]$ ls -l .ssh
total 20
-rw-------    1 mac   mac        736 Oct 16 10:25 id_dsa
-rw-r--r--    1 mac   mac        614 Oct 16 10:25 id_dsa.pub
-rw-------    1 mac   mac        539 Oct 15 16:47 identity
-rw-r--r--    1 mac   mac        343 Oct 15 16:47 identity.pub
-rw-r--r--    1 mac   mac        604 Oct 15 14:53 known_hosts2
[mac@M03H mac]$
```

9.2.4 Placing a key somewhere

As an example, imagine that it is our task to administer the host m01c. Therefore, we need to login on m01c as *root* frequently. For this to work, we copy our public key onto that machine. Our private key stays on our workstation (m03h), which we assume is well secured and we trust it.

All public keys that are listed in the file ~/.ssh/authorized_keys can be used for public key authentication (protocol version 1). For protocol version 2, there is a similar file named ~/.ssh/authorized_keys2. Consequently, for protocol version 1, we place the contents of our ~/.ssh/identity.pub file on the workstation m03h into the ~/.ssh/authorized_keys file on the host m01c:

```
cat .ssh/identity.pub | ssh root@m01c 'mkdir -p .ssh; cat >> .ssh/authorized_keys'
```

For protocol version 2 access, we append a version 2 public key file to the .ssh/authorized_keys2 file on remote host m01c:

```
cat .ssh/id_dsa.pub | ssh root@m01c 'mkdir -p .ssh; cat >> .ssh/authorized_keys2'
```

Now let us explain in detail what these commands do. They show you several features of ssh that we have not talked about yet.

- **cat** shows the contents of a file, if it is given a filename. Normally, it prints to stdout (standard out) like this:

  ```
  [mac@M03H mac]$ cat .ssh/id_dsa.pub
  ```

```
ssh-dss AAAAB3NzaC1kc3MAAACBAN4+nFYco+/MrlsT1EczWNKCdzK5C/qTjS8u4Nn5tK3/3jA8BPDgvFQY/Pnw
ZE2EfOF5EwmfylVB8xmPWhI8bBFINju2D+iAUnI8FUhFXOns61OxQ7F3o4EgISo/wmQl255r3Ro5CagHgwqT3xOr
2FtjGUMvUDcmUzVjBpcKMPK9AAAAFQDnYAluisr33q5d2Yw73jSAloVb8wAAAIBwMK9b61DIhVjh9HNLORX/Le6h
mH9CmYH85gBnSfSl12jMCTWfRYrYU185hDCMJTweslNBoPe4R8rmAV/7c/2BNN4DqNI7QKrE9oavlEBPieIUmli4
Nepw/igq9mf6EWS3WxUp39nkYkSxOxcQwbpHPObpty2mBdaZx/gR8OHh5AAAAIEAxq87X1zu/wTI+hGQJwIN3GNa
sL6dRkUlFVHoA764Fgx7bNJQSYBTd2rMK71UIWApsIBSLuFcUiQROfIOSUvUoM3zQaxcOslu9zIu2NOdwytKZSd8
YfdFz7ePfybOwCWsb4RNYz3HOBqMVUPws9hECwqnotzp7bE8uXpzcFKUz2U= mac@MO3H.itso.ibm.com
[mac@MO3H mac]$
```

But if a *pipe* (|) and another command follow, the output is directed into stdin (standard input) of the next command. People say the output is “piped” into the next command.

- **ssh** can read from stdin (standard input). We will see later what it does with the input it is given.
- We mentioned that ssh can run remote commands for us. A simple example is to run **ssh remote_host ls**. Any command that is given on the command line, like **ls** here, is treated as a command to be executed on the remote machine. To protect special characters from being expanded by the local shell, you would put the entire expression in quotes as shown above.
- **mkdir -p .ssh** creates an .ssh directory, but only if it is not already present.
- If **cat** it *not* given a filename, it reads from standard input. Now we can put it all together. With our first **cat** command, we send the contents of our key file into the **ssh** command. Our ssh client puts this through and sends it into the commands to be run at the remote side – like a tunnel! On the remote side, **mkdir** does not need any input, so it ignores it. But the **cat** command then reads what it is given from **ssh**. Instead of showing the data, we redirect its standard output with **>>** to a file, which is our authorized_keys file. A single angle bracket (>) means *write* this to that file; a double angle bracket (>>) means *append* this to that file. We don’t want to overwrite the file, just add the key to the ones that might already be present.

Do not be afraid of these apparently complicated ways of putting commands together. All this can be done in single, easy steps. But the toolbox-like nature of UNIX systems facilitates and encourages this way. Another way to do it would be to first copy the key file to the remote machine with the **scp** command (see the following section).

9.2.5 Using a key to login

Sticking to our example that we are going to be the administrator of the host m01c, we now login there as *root*:

```
[mac@MO3H mac]$ ssh root@m01c
Enter passphrase for key '/home/mac/.ssh/id_dsa':      <<< enter passphrase for that key
Last login: Tue Oct 16 11:10:41 2001 from 10.10.5.62
SuSE Linux on iSeries -- the spicy solution!
Have a lot of fun...
root@m01c:~ #
```

The only difference here, from a user’s perspective, is that you do not enter the system password on the remote system m01c, but the passphrase of your DSA key. Note the small difference: the ssh people intentionally chose the term *passphrase* when talking about keys. In contrast, the term *password* would refer to the system password.

9.2.6 Using a key in conjunction with the ssh-agent

ssh-agent is an authentication agent. If your local workstation, where your private key files reside, is trusted, you can run an ssh-agent that holds your passphrases. We say, “You add an identity to the agent to represent for you.”

This allows you to use ssh without having to enter your passphrase or passphrases each time. It works like this:

```
[mac@MO3H mac]$ eval `ssh-agent`
Agent pid 7544
[mac@MO3H mac]$
```

Note the back quotes! In the shell, they have the meaning of command substitution. The ssh-agent command actually produces output like this:

```
[mac@MO3H mac]$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-HpZS7557/agent.7557; export SSH_AUTH_SOCK;
SSH_AGENT_PID=7558; export SSH_AGENT_PID;
echo Agent pid 7558;
[mac@MO3H mac]$
```

The two variables `SSH_AUTH_SOCK` and `SSH_AGENT_PID` are needed in the environment, so our ssh client knows which agent to communicate with. By evaluating the output of the ssh-agent, these two variables will be set in the current shell and marked to be exported.

Tip 1: If you start an X session from this shell, then all derived shells (and, therefore, all your shells in the X session) will inherit their environment from that shell.

Tip 2: In an X session, you can use the application x11-ssh-askpass – a graphical dialog box that will be opened whenever you need to enter a passphrase.

Now that we have started an agent, we can add an identity. This is done with the **ssh-add** command. If the identity needs a passphrase, **ssh-add** will ask us for that. Specifically, we want to add the passphrase of our DSA key:

```
[mac@MO3H mac]$ ssh-add .ssh/id_dsa
Need passphrase for .ssh/id_dsa
Enter passphrase for .ssh/id_dsa                          <<< enter passphrase here
Identity added: .ssh/id_dsa (.ssh/id_dsa)
[mac@MO3H mac]$
```

Now, we can login anywhere, without having to enter the passphrase, where our public key for this identity is present in one of the two authorized_keys files. Let's try this:

```
[mac@MO3H mac]$ ssh root@m01c
Last login: Tue Oct 16 11:11:06 2001 from 10.10.5.62
SuSE Linux on iSeries -- the spicy solution!
Have a lot of fun...
root@m01c:~ #
```

In fact, we are not asked for a password or passphrase this time!

To give you an idea why this is so useful, ssh-based logins and other commands that use ssh are easy now. You can copy files with **scp** or **run cvs** (concurrent version system) over ssh, or execute remote commands conveniently, without having to enter any password each time.

If you leave your workstation, you can remove all identities from the agent, so they can't be used by an attacker. This is done with **ssh-add** in conjunction with the **-D** command-line switch:

```
[mac@MO3H mac]$ ssh-add -D
All identities removed.
[mac@MO3H mac]$
```

The agent can be killed using the command **`kill $SSH_AGENT_PID`**.

9.2.7 X-forwarding

X-forwarding is usually disabled by default. To enable it, simply add -X to the command line when using ssh. This tunnels your X connection in a relatively secure manner through the ssh tunnel. Also, your DISPLAY variable on the remote system is automatically set.

> **Tip:** Running graphical programs as root is often required for system administration tasks. However, we do not recommend this, for security reasons, to run the entire X session under root permissions. A way to overcome this is to start the X session as a normal user, and then use **`ssh -X root@localhost`** to become the *root* and to use the display via X-forwarding.

9.2.8 ssh clients for Windows

To access a Linux machine from a MS Windows PC, you need an ssh client for Windows. These are your options:

- PuTTY (`http://www.chiark.greenend.org.uk/~sgtatham/putty/`) is used a lot, because it is entirely free. We recommend this client, because it lets you work with Linux in a way as similar as possible to working on a “real” Linux workstation.
- Tera Term (`http://hp.vector.co.jp/authors/VA002416/teraterm.html`) with the ssh extension (TTSSH, `http://www.zip.com.au/~roca/ttssh.html`). This one is also free, but does not have so many capabilities.
- SecureCRT (`http://www.vandyke.com/products/securecrt/`) is a very good commercial program. A 30-day trial version can be downloaded.
- OpenSSH implementations for Windows are starting to appear.

The handling of PuTTY is quite easy. One thing that you might want to change is the default font size, because it is very small. To change the default font size, follow these steps:

1. Double-click the PuTTY icon. A dialog box appears.
2. Select **Appearance** and click **Change...** as shown in Figure 9-1.

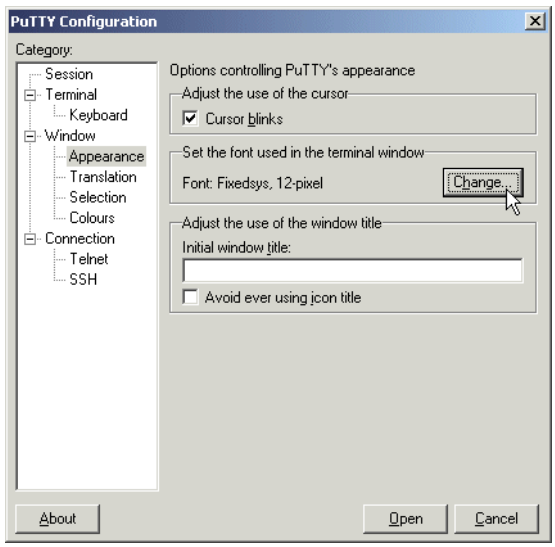


Figure 9-1 The PuTTY configuration dialog

3. Choose a font, as shown in Figure 9-2. Our suggestion is to choose **Fixedsys**, size **12**.
4. Select **Session** (upper left corner). You now see the dialog box as it looked initially.
5. Click **Default Session**, and then click the **Save** button right next to it.

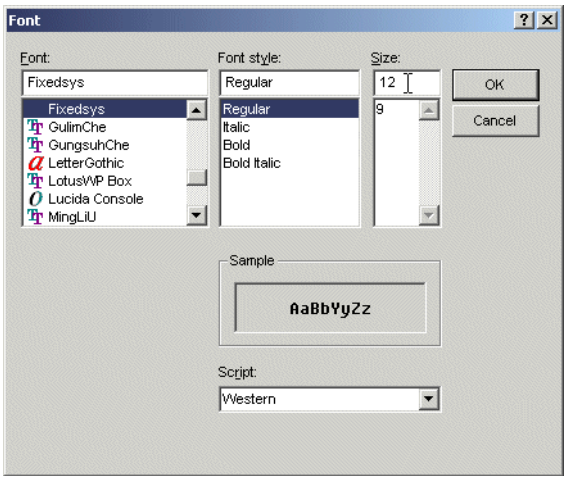


Figure 9-2 Choosing a font in the PuTTY configuration dialog

To enlarge the default window size to, for example 40x120 characters, select **Window** and enter the values into the Rows resp. Columns fields, and proceed as described above to make this the new default.

To use PuTTY to login into the host m01c, you need to fill in the Host name field in the PuTTY dialog box, and select **SSH**, as shown in Figure 9-3.

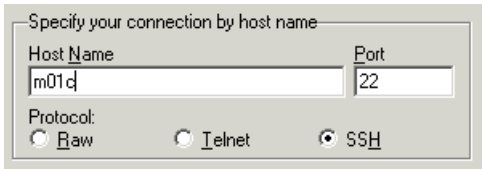


Figure 9-3 Entering the hostname and choose the protocol

The hostname needs to be a valid DNS name in your network or an IP address. Click the **Open** button. PuTTY compares the host key of the remote host to the keys in its database. If it is the first time that you connect to this host, you see the warning shown in Figure 9-4.

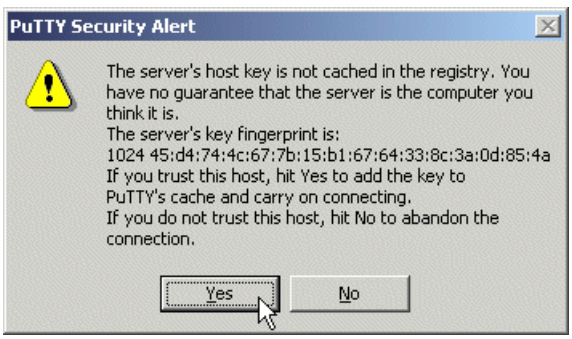


Figure 9-4 PuTTY showing a warning if the host key of a remote host has never seen before

We suppose here that you accept the key and reply with “yes”. In the screen that appears (Figure 9-5), you first have to specify as which user you are trying to login. In the example, we enter *root* there. Then you are asked for your password, and that’s it!

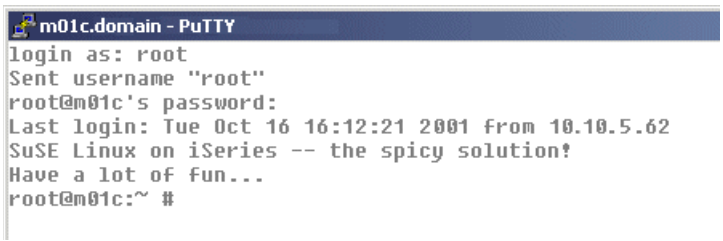


Figure 9-5 Logged in over ssh

To work with ssh keys under Microsoft Windows, look at the PuTTYgen utility to create keys (see the PuTTY home page at http://www.chiark.greenend.org.uk/~sgtatham/putty/). There is also an authentication agent, called *Pageant*.

9.2.9 Using the server

The server can theoretically be run from inetd, but in most cases it is run as a standalone server. The reason is that it needs to generate the server key before it can respond to the client, and this may take tens of seconds. However, it might be compiled with support for TCP wrapper, thereby using directives in /etc/hosts.allow and /etc/hosts.deny to grant/deny access to certain hosts based on DNS names or network addresses.

To start the server, you will probably simply use the init scripts of your distributor. Since sshd is normally started by default, you should not have to take any action here.

The configuration file is /etc/ssh/sshd_config. See **`man 8 sshd`** for more information.

9.3 scp (secure copy)

`scp` copies files between hosts on a network. It uses ssh for data transfer and uses the same authentication and provides the same security as ssh. Especially if you work with keys and ssh-agent, it is really a very convenient way to transfer files.

It is easy to use. The basic syntax to copy file `name` to host `host` is:

```
scp name host:
```

If you want to give the file a new name at the remote location, use:

```
scp file host:newname
```

To specify an alternative path where the file should be copied to, enter it after the colon:

```
scp file host:stuff/newname
```

stuff is a relative path to your home directory, because it is not preceded with a slash (/). Absolute path names start with a slash in front.

If your account on the remote machine is not the same as the one you logged in locally, you can specify a user name as *bob@*:

```
scp file bob@host:stuff/newname
```

Now, to reverse the procedure and retrieve a file, exchange the two command-line arguments. For example, to copy the file /etc/motd from the remote host m01c (which we can access as *root*) to the local current directory, use this command:

```
scp root@m01c:/etc/motd .
```

The single dot (**.**) means “current directory” (it is the single dot that you see when you run **ls -la**). The file is created with the same name as remote, motd. To name it m01c_motd, we could also have used:

```
scp root@m01c:/etc/motd m01c_motd
```

In addition, we want to mention two other command-line switches that **scp** understands:

-p Preserves modification times, access times, and modes from the original file.
-r Recursively copies entire directories.

On a Microsoft Windows PC, your options are:

- Use **pscp**. It belongs to the PuTTY suite and is used from the command line, just like its UNIX counterpart **scp**. It can be downloaded form the PuTTY site (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html) and works similar to **scp**.
- There is a graphical frontend for pscp, called iXplorer, which is available at: http://www.i-tree.org/ixplorer.htm
- WinSCP is another implementation; it is a freeware graphical scp client. See http://winscp.vse.cz/eng/

9.4 sftp (secure FTP)

scp, the secure copy command, is nice, but sometimes life is easier if you can actually look at files interactively. There is a way to do this – **sftp**.

sftp is an interactive file transfer facility based on ssh that provides an interface very similar to an FTP client. If you have not used a command-line FTP client yet, you will find the usage awkward. But there are some nice graphical frontends for Linux as well as for Microsoft Windows, so you don’t have to dispense with security (see the following example).

Here is an example of starting a session (note that we don’t give a password in this example because we have an ssh-agent running):

```
[mac@MO3H mac]$ sftp root@m01c
Connecting to m01c...
sftp>
```

We see the **sftp** prompt, waiting for us giving commands. The typical commands are:

ls List directory content
cd ... Change directory
pwd Print working directory (where am i?)
get Transfer a file from remote to local
put Transfer a file from local to remote
mkdir Create a directory
quit Close the connection and quit the client

You can use a prefix such as ls, cd, pwd, and mkdir with the letter “l” as in “local”. Then the commands will be executed for the local directory instead of the remote directory.

Subtle differences to the common FTP command set are:

- **sftp** does not understand **bye**; instead use **quit**.
- **get** and **put** understand a **-P** flag to preserve file access times and permissions.
- There is no **mget** or **mput** command.

On a Microsoft Windows PC, your options are:

- SecureFX is a graphical, commercial FTP and sftp client that works only with SSH2 servers. A 30-day trial version is available from: http://www.vandyke.com/download/securefx/index.html
- psftp is another tool from the PuTTY suite: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
- There is an sftp program in the OpenSSH implementation for Windows: http://www.networksimplicity.com/openssh/
- Shaolin Secure FTP (GPL): http://www.shaolinsecureftp.com/

9.5 Telnet

Telnet is another application used to communicate to other computers. It's use is very similar to that of ssh. The Telnet protocol usually uses port 23. It is described in RFC854, which was first published in 1983.

For information about all the RFCs for the Internet and the OSI networking model, go to: http://www.rfc.net/

9.5.1 Security issues with Telnet

Telnet is inherently insecure by design. This is because all characters are transmitted in clear text, including passwords.

For the purpose of a connection to a server from a client on a trusted LAN, it may be sufficient. For use over the public Internet, it is at best dangerous. There are several implementations of extensions that make Telnet secure, but none of them is particularly common in use or easy to set up.

Since the security issues can be eliminated by ssh, we strongly recommend that you use Telnet only in special situations. Since Telnet does not have any advantages over ssh, these situations should be rare (for example, if no ssh implementation is available for the platform you work on).

Linux distributors usually disable Telnet access by default.

A notable application of Telnet is the virtual console on the iSeries. The virtual console can only be accessed via the Telnet protocol, using a Telnet client to connect to port 2301 of the hosting partition. You should use precaution to access the virtual console over trusted networks only!

9.5.2 Telnet client software

Because of its generic design and limited support, the Windows Telnet client that is included with Microsoft Windows products should be used only in "emergencies". Instead, you can use one of several widely available Telnet clients. In fact, the above mentioned PuTTY program and the Windows ssh client can also be used as a Telnet client.

9.6 Problem determination

If you cannot connect to your Linux server, you should consider each of the items in the order provided:

1. Can you Telnet to another known host?
2. If you can, check the IP address of the Linux interface and make sure you entered the correct address.
3. Can you ping the address? If not, is the interface up? If the interface is up, did you set up the default routing on the iSeries and the Linux server correctly?
4. If you cannot, is your Telnet client set up correctly? If it is not, make sure you have configured it correctly and try the operation again. If it is set up correctly, make sure you are trying to Telnet to the proper IP address.
5. If you are using the correct IP address, make sure you set up the LAN adapters in Linux correctly.
6. If the adapters are up, make sure you set up the default gateway and the routing correctly.
7. Make sure that the Telnet service is enabled in the inetd or xinetd configuration.

10

FTP servers on Linux

This chapter gives you an overview on File Transfer Protocol (FTP) servers when running on a Linux partition on the iSeries server. This is *not* meant to be a complete reference about FTP. Please refer to the documents and Web pages referenced throughout the text for more in-depth information.

This chapter also provides assistance in learning about the initial configuration that is set by the installation program.

The following topics are discussed:

- FTP: The protocol
- Security considerations
- Available servers
- Configuration files
- Default setup in different Linux distributions
- Testing your server
- The OS/400 FTP daemon: Transfer tips

10.1 FTP: The protocol

FTP is based on TCP/IP and is used by FTP client software to access FTP servers. It provides the capability to upload or download files in a simple, robust, and efficient manner. It is often used to exchange files between different platforms because FTP software has been implemented under virtually every operating system. Servers are either configured to offer their service to registered users of a system or to unknown users (anonymous FTP).

FTP has been in wide use for many years before HyperText Transfer Protocol (HTTP), the "language" of the WWW, was even invented. The protocol is defined in RFC959 (see http://www.rfc-editor.org/rfc/rfc959.txt). Extensions were later described in other RFCs, of which you can find an extensive list on the WU-FTPD Web site at: http://www.wu-ftpd.org/rfc/

Web browsers like Netscape and Microsoft Explorer usually have a built-in FTP client. It is possible that you have been accessing an FTP server without being aware of it, because your Web browser automatically switches from HTTP to FTP when it accesses a URL that starts with "ftp://". Apart from the "transparent" implementations in Web browsers, there is dedicated FTP client software, like WS_FTP (http://www.ipswitch.com/), which is the most popular one on Windows.

As a good graphical FTP client for Linux, gFTP is recommended (which is either provided by distributors or can be obtained from http://gftp.seul.org/). gFTP also supports the SSH protocol (see 9.4, "sftp (secure FTP)" on page 220).

FTP is not used as a principal method to deploy a file server, even though Microsoft Windows offers this type of connection. While FTP is very efficient in transferring single large files due to the small protocol overhead, other protocols like Network File System (NFS) or Server Message Block (SMB) offer the features that are necessary to open multiple files simultaneously, hold them open, and lock them against write access of other users (which is where we speak of file sharing, as opposed to file transfer).

You should be aware that FTP connections can be established in two different "modes": *normal mode* and *passive mode*. In both cases, the client first opens a TCP connection to the FTP control port of the server (port 21). Normal mode means that for data (file listings, files) the server opens a TCP connection back to the client (which is unusual for a TCP/IP client-server protocol). This can prevent communication because often the FTP client is located behind a firewall or a NAT router that would not let the ingoing connection pass.

Fortunately, most clients (and servers) can use the passive mode instead. This is initiated by the client by sending the PASV keyword, thereby instructing the server to accept another connection made by the client as data connection. To use the passive mode, the client has to be configured accordingly (there is a configuration setting in most browsers, or they might use passive mode by default). In a Linux command-line FTP client, you could use the **passive** command (or simpler, use **pftp** instead of **ftp**). You should note that the DOS command line FTP client does not support passive mode.

Passive mode FTP is becoming more and more common, and its use is recommended wherever feasible, for example, when the client and server support it; it has no known disadvantages.

10.2 Security considerations

FTP is inherently insecure since all commands, user names, passwords, and data are transmitted in cleartext. Therefore, authenticated FTP (user login), where a password is needed, is usually deployed only in small trusted intranets, while anonymous FTP is common on the Internet.

Some people achieve a somewhat similar effect to “user FTP” by using hidden directories in the publicly accessible file area with names that are difficult to guess. However, directories that are world writable (that would be needed for write access) can be (and usually are) exploited by other people to distribute their files (often of illegal content, while you are the one who is liable for files on your server).

If it is sufficient for your needs to transmit the password securely. However, to have the files and directories transferred unencrypted, you could get by with ssh port forwarding the control channel of the FTP connection. Use the following command:

```
ssh -L2001:server:21 server
```

Then, connect (in passive mode) with:

```
pftp client 2001
```

On a Windows PC, you can achieve the same effect with the free ssh client PuTTY. Refer to the documentation on how to set it up (http://www.tartarus.org/~simon/puttydoc/output.txt).

Important: If files are to be transferred over untrusted networks, an ssh-based method like scp (see 9.3, “scp (secure copy)” on page 219), rsync, or sftp (see 9.4, “sftp (secure FTP)” on page 220), or HTTPS is recommended. These services allow for authentication, encryption, and data integrity checking.

10.3 Available FTP servers

Some of the most widely used FTP daemons on the Linux platform are:

- BSD-derived in.ftpd
- WU-FTPD
- ProFTPD
- vsftpd

Each of these are briefly described in the following sections. Please refer to Table 10-1 on page 227 for an overview on the configuration files and other relevant files.

10.3.1 The BSD-derived FTPD

The BSD-derived FTP daemon is a simple, but solid implementation that is distributed with Linux NetKit (http://www.eleves.ens.fr:8080/home/madore/programs/#prog_ftpd-BSD). In its variants, it has been in use on most UNIX systems for many years. The binary is usually called in.ftpd, like other Internet servers (in.fingerd, in.telnetd, ...).

The principal source of information is the manual page (seen via **man 8 ftpd**). in.ftpd (the actual the name of the binary) is spawned as one process per client by the inet daemon and configured by the command line arguments that it is given. You can change its behavior by editing the configuration file of your Internet “super server” (inetd/xinetd). It can also be run in standalone mode via the **-D** command line switch. Like the other servers, this daemon supports user (authenticated) FTP and anonymous, FTP depending on its configuration.

10.3.2 WU-FTPD

WU-FTPD (originally maintained by Washington University in St. Louis, Missouri) is the most used FTP server on Linux and UNIX servers (http://www.wu-ftpd.org/). It can be run by inetd or in standalone mode and provides excellent performance. WU-FTPD is a much larger program than in.ftpd. It offers many more features, like on-the-fly compressions and decompression.

Just as more complex programs, in general, tend to have more bugs, WU-FTPD has a long history of security vulnerabilities. You should always make sure that you run the latest version and watch the public security mailing lists. In the wuftpd package, SuSE ships both the current version and an older version (2.4.2). The latter does not have the newest features, and can’t run in standalone mode, but it went through an extensive security audit in 1998, and no problems have been found since then. Both servers are contained in the same package, and all files of the newer version are suffixed with -2.6.

10.3.3 ProFTPD

ProFTPD is a newer server. The project (http://www.proftpd.net/) was started with the intention of a complete redesign of WU-FTPD to implement more features, more security, and better configurability. Many people like it for its Apache-style configuration file. The feature list includes multiple virtual FTP servers and per directory “.ftpaccess” configuration similar to Apache's “.htaccess” files. Although it is meant to be more secure than WU-FTPD, ProFTPD has, in fact, also had some security problems in the past.

Unfortunately, due to large system call overhead ProFTPD does not scale as well as WU-FTPD. It is also said to have some memory leaks that can render the standalone mode unusable. Therefore, ProFTPD can maybe not be recommended for very high traffic sites.

10.3.4 vsftpd

vsftpd (“very secure FTP daemon”, http://freshmeat.net/projects/vsftpd/) is an FTP server that was newly written with solid security. It supports user and anonymous FTP and offers bandwidth limiting.

10.3.5 Summary comparison

Table 10-1 shows details about of the four common FTP daemons.

Table 10-1 Configuration files and other files relevant to the four common FTP daemons

name of the binary	in.ftpd	wu.ftpd	proftpd	vsftpd
Running mode	inetd or standalone	inetd or standalone	inetd or standalone	inetd
Configuration files	none (in.ftpd is configured via command line options) /etc/ftpchrootd /etc/nologin	/etc/ftpaccess /etc/ftpconversions /etc/ftphosts /etc/ftpgroups	/etc/proftpd.conf (may also be located in /etc/proftpd/)	vsftpd.conf vsftpd.banned_emails vsftpd.chroot_list (all in /etc)
List of unwelcome or restricted users	/etc/ftpusers			Can be set with the userlist_* directives
Message files (if a path name is not preceded by / , it is relative)	/etc/ftpwelcome /etc/motd .message msgs/*	/etc/ftpmsg.dead /welcome.msg .message	msgs/welcome.msg msgs/msg.dead msgs/welcome.msg .message	.message (if enabled)
Log files	/var/log/messages (/var/log/ftpd)	/var/log/messages /var/log/xferlog	/var/log/messages and others, as specified by the TransferLog and ExtendedLog directives	/var/log/vsftpd.log
Additional programs		ckconfig ftpcount ftpwho ftpshut ftprestart xferstats_wuftpd	ftpcount ftpwho ftpshut	

10.4 Configuration files

The configuration files of the FTP servers are located in the /etc directory. You will probably see the file first in /etc/inetd.conf. All editing should be done using a text editor as described in 3.1.2, “Editing files using text editors” on page 88.

After editing /etc/inetd.conf, you need to make the (running) inetd reread it by running the command:

```
killall -HUP inetd
```

This is actually the command that would be run for you when you use the init scripts from the distributors, which would be:

- On a SuSE installation: `rcinetd reload`
- On a Turbolinux installation: `/etc/rc.d/init.d/inet reload`

10.5 Default setup in different Linux distributions

When Linux is installed on the iSeries server, an FTP server is usually automatically installed and configured, and sometimes also started.

Which server is installed by default, and how it is configured, depends on the distribution you use and the choices you made (the distributor might offer all three servers, but install only one). SuSE installs in.ftpd, while Turbolinux installs proFTPD.

To find out about the default setup, you can either refer to the distributor's documentation, or look into /etc/inetd.conf, the server's configuration file.

You can use the command **rpm -qa | grep ftp** to find all packages that have "ftp" in their name. Get information about each package with **rpm -qi packagename**. It is a good idea to read the documentation that comes with the package (**rpm -qd packagename** tells you which files these are).

Depending on the distribution's defaults, an inet daemon (inetd) could be up and running, listening on port 21 (ftp-control). The following command would tell you if any process is listening on the FTP port:

```
lsof -i tcp | grep ftp
```

If you see "inetd", the inet daemon is running and configured to listen on the FTP port. If you see "proftpd", then proftpd is running in standalone mode and listening on port 21 itself.

To see which FTP daemon the inet daemon is configured to start and which command line arguments to use, enter this command:

```
grep -v ^# /etc/inetd.conf | grep ftp
```

10.6 Testing your server

The configuration of any running FTP service is of course determined in the configuration file. But to actually confirm that the server runs as intended, you should check the configuration by connecting to the server and trying. There is a simple way to do this. If you connect to localhost and try to authenticate yourself as a user known to the system or as "ftp" (shorter synonym for "anonymous"), you will see how the server reacts. Presumably, it will never be possible to authenticate as user *root*; therefore, use the account of an ordinary user. After trying this from localhost (the machine itself), you can repeat it from an external host as well.

While testing, you might also want to keep an eye on /var/log/messages, where the system log daemon writes to (and most FTP daemons use the syslog facility to hand over their messages). To watch the file, you can use the command **tail -f /var/log/messages** on another terminal (to quit the process, press Ctrl-C).

10.7 The OS/400 FTP daemon: Transfer tips

Linux on the iSeries and the native OS/400 FTP daemon and client can be used to transfer a file into, or out of, the OS/400 IFS. Non-text files should be transferred in binary mode. Use the **bin** FTP command to set this mode.

Not all FTP clients default to binary. To switch back to text mode, use the **ascii** or **ebcdic** FTP command, depending on the source encoding type.

Placing files into the IFS requires the use of naming format 1 of the iSeries FTP server. This format allows the use of UNIX path names.

To enter into naming format 1, you can either:

- Execute the FTP command: **`cd /`**
- Use the FTP command **`site namefmt 1`** for a remote client, or use **`namefmt 1`** as a local client.

Use the FTP command **`cd`** to set the remote (iSeries IFS) directory. Remember to set the transfer mode (**`binary`**, **`ascii`**, or **`ebcdic`**) as necessary before using **`put`** or **`get`**.

10.8 Problem determination

This section is not a full description of the problems that you could run into. It is just reviews the most obvious ones that most users usually encounter. For other problems, we recommend that you consult the documentation that came with your distribution.

- **Host does not respond**
  - *Cause*: You most likely typed in the wrong hostname or host address, or the server is not started.
  - *Solution*: Verify your server is started.
- **Access denied**
  - *Cause*: You probably have an authority problem on the file you are trying to put or get.
  - *Solution*: Try to use **`ls -l`** in the directory and check if the file has the right authority. Authority can be changed with the **`chmod`** command, for example:

    ```
    chmod 755 [filename/directoryname]
    ```

11

Samba

This chapter provides a framework for using Samba and Linux on the iSeries. You'll see that using Linux and Samba can very effectively extend your desktop environment with minimal effort and expense. The topics covered are:

- Understanding Samba
- Preparing to use Samba
- Configuring Samba to extend desktop printing services
- Configuring Samba to provide additional file space for your users
- Problem determination

11.1 Understanding Samba

Samba is an open source/free software suite that provides seamless file and print services to Server Message Block (SMB) clients. The SMB protocol is used by PC-related machines to perform peer-to-peer and client/server related sharing of network resources. These resources can be printers, files, folders, and other resources that need to be shared and can be supported by SMB (Samba).

Microsoft Windows uses the protocol natively. All of these capabilities allow you to extend the Windows LAN environment as if you had just installed another Windows-based server with one small difference. This server operating system is free and just as robust, if not more, than Microsoft Windows. In combination with the iSeries NetServer (that runs in OS/400) and LPR/LPD, you can extend your network to meet your needs with very little additional software and hardware costs.

You can use a Samba server to run printers and authenticate users, share files, and directories, just like Microsoft Windows. Samba can even act as a Primary Domain Controller (PDC) or as a Backup Domain Controller (BDC) in your Windows network. You can use it to run OpenLDAP and add LDAP function to your Windows Network without the expense.

You can use Samba and the iSeries NetServer product to share printers and files on the iSeries server as well as on your Linux partition.

You can find detailed information about Samba in the IBM Redpaper *Implementing Linux in your Network using Samba*, REDP0023. This document references how to install and configure Samba in Linux on an IBM Netfinity Series server. The instructions are the same for Linux on iSeries.

You should also refer to the Samba home page to learn more about Samba, its history, and the current and planned implementations: `http://www.samba.org/`

11.2 Preparing to use Samba

To use Samba on your Linux partition, Samba must be installed. There are two parts of Samba: a Samba client and a Samba server. Both have their own RPM packages called `smbclntXXXX.ppc.rpm` and `sambaXXXX.ppc.rpm` respectively. As of Samba Version 2.2, these packages were merged into `sambaXXX.ppc.rpm`. To determine whether they are installed, type the following command:

```
iserieslinux:~ # rpm -q samba
package samba is not installed
iserieslinux:~ # rpm -q smbclnt
package smbclnt is not installed
```

If the package is installed, **rpm** would return the version:

```
iserieslinux:~ # rpm -q samba
samba-2.0.7-95
iserieslinux:~ # rpm -q smbclnt
smbclnt-2.0.7-95
```

If Samba is not installed, follow the instructions from your particular distributor to install the package. Once Samba is installed, you can then configure it for your particular network needs.

11.2.1 Using Linux as a Samba client

To use the Linux server as a Samba client, you need another Samba server that shares a directory. This could be an Microsoft Windows machine or an OS/400 partition, for example.

As an example, we share a directory on a Windows PC, and try to mount it from the Linux partition. Imagine the Windows PC is called `P23BK53A`, which shares a directory under the name *public*. We connect using the username `itscid02` and password `itsolock`. And we make the contents of this share available in an empty directory in the Linux system, for example `/mnt/pcpublic` (we say we "mount" the shared directory in that `/mnt/pcpublic` directory). This can be accomplished using the command:

```
mount -t smbfs -o username=itscid02,password=itsolock //p23bk53a/public /mnt/pcpublic
```

If we then list the contents of /mnt/pcpublic, we actually see the contents of the shared directory on the PC. We can unmount the directory again using:

```
umount /mnt/pcpublic
```

Note that you must be *root* to use those commands. Only if the mount point is listed in /etc/fstab, a regular user can do this. See 3.3.7, "Automatically mounting a disk partition at boot time" on page 100.

smbclient

The program **`smbclient`** is used to perform SMB Client command. **`smbclient`** is used for an FTP-like connection to the server, except that the SMB protocol is used rather than FTP. You use it to get, put, and print files, among many other things.

For example, to list the shares that were shared from that same PC, use the command:

```
root@iserieslinux:/ # smbclient -U itscid02 -L p23bk53a
added interface ip=9.5.92.90 bcast=9.5.92.95 nmask=255.255.255.248
Password: <enter password>
Domain=[ITSO] OS=[Windows 5.0] Server=[Windows 2000 LAN Manager]

        Sharename      Type      Comment
        ---------      ----      -------
        Public         Disk
        IPC$           IPC       Remote IPC
        print$         Disk      Printer Drivers
        IBMNetwo       Printer   IBM Network Printer 12 PS
        ADMIN$         Disk      Remote Admin
        C$             Disk      Default share

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
```

The **`smbclient`** tool can be used to test the configuration of your own server as well.

11.2.2 Connecting to the iSeries NetServer

To connect to an OS/400 system using Samba, you need an updated version of NetServer that supports the Linux Samba client. This update should be available as a PTF. At the time this redbook was written, we needed to install these PTFs in order to connect to the iSeries NetServer in V5R1:

- MF27247
- MF27248
- MF27249
- MF27294
- MF27295
- SI02179

11.2.3 General things you should know about the Samba server

A *share* is a resource on the network to which access is given. The rights to the resource are determined by the rights granted to the user profile of the person logging in. In the case of a file share, this is viewed by the users as an extension of their native file system. You can specify guest services where no password is required; in this case, a guest account is used to define access to the share being used.

Before you start configuring Samba, plan your installation and view the documentation for your particular distribution of Linux.

11.3 Configuring the Samba server

The configuration file is usually located in `/etc/smb.conf` or `/etc/samba/smb.conf`. For information on editing files, refer to 3.1.2, “Editing files using text editors” on page 88. You can also use SWAT to configure your Samba server, as explained in 11.3.3, “Samba Web Administration Tool (SWAT)” on page 240.

If you need more information, see the man page of smb.conf (by entering **`man smb.conf`**), or refer to the Samba project Web site at: `http://www.samba.org/`

The configuration file smb.conf is divided into two main sections:

- **Global settings**: These affect the connection of your Samba server to your network.
- **Shares**: This is where you define what resources are going to be shared on the network and accessible among the users. This section includes:
  - *Homes*: Defines the user’s home directories.
  - *Printers*: Defines the available printers.
  - *Shares*: Has an entry for each directory you want to define and share.

The Samba configuration file is logically divided. The start of each section is marked with a headline in brackets, like `[printers]`. So if we mention the `[printers]` section, we refer to the lines below the `[printers]` headline in the configuration file. For an exact description of the parameters in the `smb.conf` file, refer to the man page using the **`man smb.conf`** command.

Sections other than those designated as guest services require a password to be used. The client provides the username and password, although some older clients only provide a password. For these older clients, you have to specify a list of user names to check against this password with the `user=` option for the share you are implementing. Usually with Windows clients, this is not an issue.

The [homes] section in Samba configuration file is a default section that is used to create a file share for each person that logs in. As a user logs in, the existing sections are scanned. If there is a match, the section is used; if not, the name is treated as a user name. It is authenticated against the local password file. If the name and password are correct, a share is created by duplicating the [homes] section. The share name is changed from [homes] to the located user name. If a local directory is set up in the user account, then that directory is used. Consider these items:

- If you have set up a guest account in the [homes] section, all home directories will be visible to all clients with no security needed.
- The browseable parameter for home directories created automatically will be inherited from the browseable flag in the [global] section of the configuration file, if none is given in the [homes] section. If you state browseable=no in the homes section, this will hide the [homes] share but make any auto home directories visible.

In the following discussion, we describe how to modify smb.conf to use the Samba directory server and a Windows NT participant. We cover only the parameters you use to get these functions working.

In the configuration file, all text behind a semicolon is interpreted as comment. To use those items or sections that are comments, remove the semicolon from the first position in the line and make the required changes to the default provided.

As an example, here is a possible configuration file:

```
;
; /etc/smb.conf
;
; Copyright (c) 1999 SuSE GmbH Nuernberg, Germany.
;
[global]
   workgroup = ITSO
   guest account = nobody
   keep alive = 30
   os level = 2
   kernel oplocks = false
   security = user
   netbios name=iserieslinux

; Uncomment the following, if you want to use an existing
; NT-Server to authenticate users, but don't forget that
; you also have to create them locally!!!
;  security = server
;  password server = 192.168.1.10
  encrypt passwords = yes

   printing = bsd
   printcap name = /etc/printcap
   load printers = yes

   socket options = TCP_NODELAY

   map to guest = Bad User

; Uncomment this, if you want to integrate your server
; into an existing net e.g. with NT-WS to prevent nettraffic
;  local master = no

; Please uncomment the following entry and replace the
```

```
; ip number and netmask with the correct numbers for
; your ethernet interface.
;   interfaces = 192.168.1.1/255.255.255.0

; If you want Samba to act as a wins server, please set
; 'wins support = yes'
   wins support = no

; If you want Samba to use an existing wins server,
; please uncomment the following line and replace
; the dummy with the wins server's ip number.
;   wins server = 192.168.1.1

; Do you wan't samba to act as a logon-server for
; your windows 95/98 clients, if so uncomment the
; following:
;   logon script =%U.bat
;   domain logons = yes
;   domain master = yes
; [netlogon]
;   path = /netlogon


[homes]
   comment = Home Directory
   browseable = yes
   read only = no
   create mode = 0750

[tmp]
   comment = Temp Directory
   browsable = yes
   read only = no
   create mode = 750

; The following share gives all users access to the Server's CD drive,
; assuming it is mounted under /cd. To enable this share, please remove
; the semicolons before the lines
;
; [cdrom]
;   comment = Linux CD-ROM
;   path = /cdrom
;   read only = yes
;   locking = no

[printers]
   comment = All Printers
   browseable = no
   printable = yes
   public = no
   read only = yes
   create mode = 0700
   directory = /tmp
```

The [global] section

The first section to be concerned with is the `[global]` section. This section defines how your Samba server will connect to and behave in the network. The first setting to change is:

```
workgroup = ITSO
```

This is the name of the workgroup or the name of your current Windows NT domain. The workgroup in our example is ITSO. You should use the name of the workgroup this server will be in.

The next parameter to change is `netbios name`. This should be the name you want your users to see when this server is on the network. In the example, the name we want users to see on the network for this server is `iSerieslinux`:

```
netbios name = iSerieslinux
```

Passwords and Samba

Handling passwords in Samba is quite a complicated topic. Samba has a separate `/etc/smbpasswd` file, which is quite similar to the normal /etc/passwd file. (Refer to 3.6.6, “Layout of /etc/passwd” on page 107.) Basically it contains a line per user, and that line contains the encrypted passwords. It is only useful when encrypted passwords are used.

Only when the option `encrypt passwords = true` is set, the clients are forced to send encrypted passwords. This means the Samba server does not know the password anymore, but only the encrypted value of the passwords. These encrypted values can only be compared to the values in `/etc/smbpasswd` (where they are stored in the same encrypted form) and the users that are only in `/etc/passwd` cannot use the Samba server anymore. In the other case, when they are sent non-encrypted, the Samba server can check the password against the passwords in the `/etc/passwd`.

To set the passwords in this file, use the **smbpasswd** command. This command can be used for several purposes, but the most important is discussed here. As *root*, you can specify the **-a** option to **smbpasswd** to add a user to the `smbpasswd` file. (It should already be in the normal `passwd` file, as explained in 3.6.4, “Managing users” on page 104.) For example to add the user test and set the password of that user in the `smbpasswd` file, do this:

```
root@iserieslinux:/etc # smbpasswd -a test
New SMB password:
Retype new SMB password:
Added user test.
```

If a normal user runs the **smbpasswd** command, it only asks to change the password. Users cannot add themselves to the `smbpasswd` file. In addition, the Samba daemon should be running while a user changes their password. For *root*, this requirement is not there because they have the authority to change the /etc/smbpasswd file directly.

The /etc/smbpasswd file is used next to the normal /etc/passwd file. If a user does not exist in /etc/smbpasswd, they are looked up in the /etc/passwd file and their password is validated.

For more information on the layout of `/etc/smbpasswd`, refer to the man page with the **man 5 smbpasswd** command. For more information on the **smbpasswd** command, see **man 8 smbpasswd**.

WINS

Attention: When using the option `encrypt passwords = true`, only users listed in /etc/smbpasswd can see the Samba server. If no encrypted passwords are used, users listed in /etc/passwd can also use the Samba server when they are not in /etc/smbpasswd.

WINS is a sort of Dynamic Name Server (DNS) that maps names to IP addresses, but it is a Microsoft-specific version of DNS. If you want your Samba server to act as a WINS name server on you network, set the WINS support option to “yes”. If there is already a WINS server in your network that you want to use to resolve names of other hosts, set it with the WINS server option.

These two options (`wins server` and `wins support`) are mutually exclusive (only one or the other should be set), but they allow clients outside the local network segment to connect (if DNS is not sufficient).

Verifying the configuration file with testparm

To verify the configuration file, run the **`testparm`** command. This command interprets the configuration file and outputs the values it detected. This is a very handy tool to determine whether your configuration file is correct and all parameters specified are indeed picked up by the Samba server.

11.3.1 Configuring Samba for directory shares

You can set up Samba to allow public and private directories for users and groups of users. A Samba server can be useful when you offer resources to the users on your network. This section explains how to create a share. In this scenario, we configure a share for the Microsoft Word processing group. The sample share section in the smb.conf file looks similar to this:

```
[wordprocessing]
comment = word processing files
path = /documents
browseable = yes
printable = no
writable = yes
write list = @users
```

This section is set up to have a directory with the name of `/documents`. It must already exist; it can be browsed and is writable. This share is seen by users as wordprocessing; they can write and browse the contents of their share.

The `write list = @users` parameter tells us there is a group (denoted by the @ symbol) called *users* that have read-write access to the service, regardless of how the writable option is set.

You can name individual users and groups in the same parameter as in:

```
write list = admin, tommy, rockopera, @accounting
```

In this example, there are three users, admin, tommy, and rockopera along with a group, the accounting group, that have write access to the wordprocessing share.

Before you implement user directory shares, you should really understand the implications of share permissions. (See 3.8, “Security issues” on page 112, for more information on file permissions.) Although you can control the share permissions with share parameters, UNIX permissions are applied before a user can access files on the share. You need to set the UNIX file and directory permissions. When a user creates a new file on the shared directory, the default create mask used is 0744. For directory creation, the default create mask is 0755. If you want, you can force a different creation mask. The parameters for doing this are in Table 11-1.

Table 11-1 File and directory mask parameters

Parameter	Description
Create mask	This is the permissions settings for files. A mask of 0777 allows read, write, and execute actions on a file for the owner, the group, and all others. A mask of 0755 allows read, write, and execute for the owner of the file, read and execute for the group and read, and execute for all others.
Directory mask	This is the permissions settings for directories. A mask of 0777 allows read, write, and execute actions on a directory for the owner, the group, and all others. A mask of 0755 allows read, write, and execute for the owner of the directory, read and execute for the group, and read and execute for all others.

An example of setting up a share where only the owner has permissions for new files and directories created in their share is:

```
[homes]
comment = Home Directories
path = %H
valid users = %S
browseable = no
writable = yes
create mode = 0700
directory mode = 0700
```

This example uses variables %H and %S. The %H variable represents the home directory of the user; this is usually derived from the home directory setup in the users configuration file in LINUX security.

The %S variable is the actual service the user is requesting. These values are derived when the user logs into the system and the smb daemon passes the values to these variables. In the case of the [homes] service, it becomes the users name.

11.3.2 Connecting to the Samba server

It usually takes a while for a Samba server to show up in larger networks. A quick way to find your Samba server once you have reconfigured and restarted the smb and nmb daemons is to use the Microsoft Windows Find: Computer dialog (click **Start->Find->Computer**). Then, you should see the display shown in Figure 11-1.

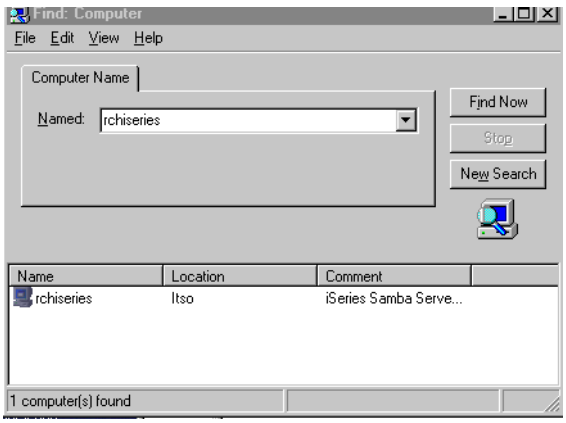


Figure 11-1 Sample windows Find computer session

Fill in the netbios name that you assigned to your server and click **Search Now**. Your Samba server is displayed in the bottom box, just as in the previous figure. From here, simply double-click the server name and you are prompted for a user name and password. Once you enter in the proper entries, you see a list of all the current shares, both files and printers. See the example in Figure 11-2.

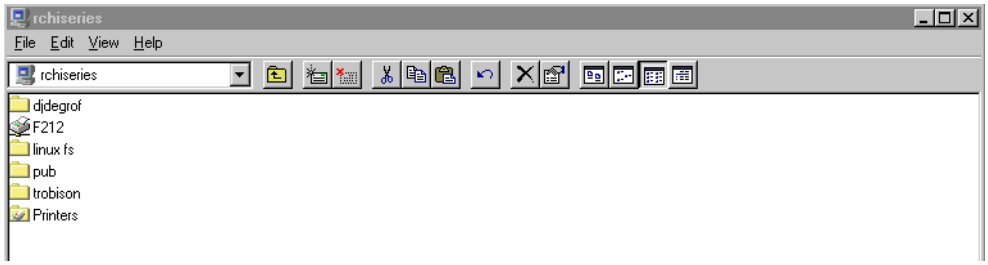


Figure 11-2 Listing of shares available

Only those shares you are authorized to see are displayed.

11.3.3 Samba Web Administration Tool (SWAT)

The Samba Web Administration Tool allows the remote configuration of the smb.conf configuration file through a Web browser. That means you can configure Samba in a GUI-like environment and maintain users access through a GUI interface.

SWAT itself is a small Web server and CGI scripting application, designed to run from inetd and provides access to the smb.conf configuration file. An authorized user with the *root* password can configure the smb.conf configuration file via Web pages. SWAT also places help links to all configurable options on every page, which lets an administrator easily understand the effect of the changes. It is automatically included in the Samba RPM package.

Make sure SWAT is automatically started. See your distributors documentation for information on how to do this.

- In the /etc/services file, you must have the following line:

  ```
  swat 901/tcp
  ```

- If xinetd is used instead of inetd, in the /etc/xinetd.d directory, you must have the SWAT file with the entry:

  ```
  swat disable = no
  ```

You can control who can access the SWAT service with the /etc/hosts.allow and /etc/hosts.deny files (see **`man 5 hosts_access`**).

Now you are ready to use SWAT. To start SWAT, point your favorite Web browser to the Internet address of your Samba server on port 901:

```
http://hostname:901
```

After you load the home page of SWAT, you see a window similar to the example in Figure 11-3.

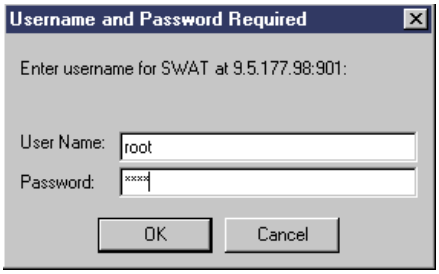


Figure 11-3 SWAT log in

This is the login prompt to gain access to SWAT. You can login with any users account, but only logging in as a user root allows you to make changes. Please note that the password is transmitted over the network in plaintext. This can be a security issue, so we recommend that you do SWAT administration only over a trusted network connection. For information on how to make SWAT secure, see: `http://www.samba.org/samba/docs/swat_ssl.html`

Once you are correctly logged in, you see a Web page similar to the example in Figure 11-4.

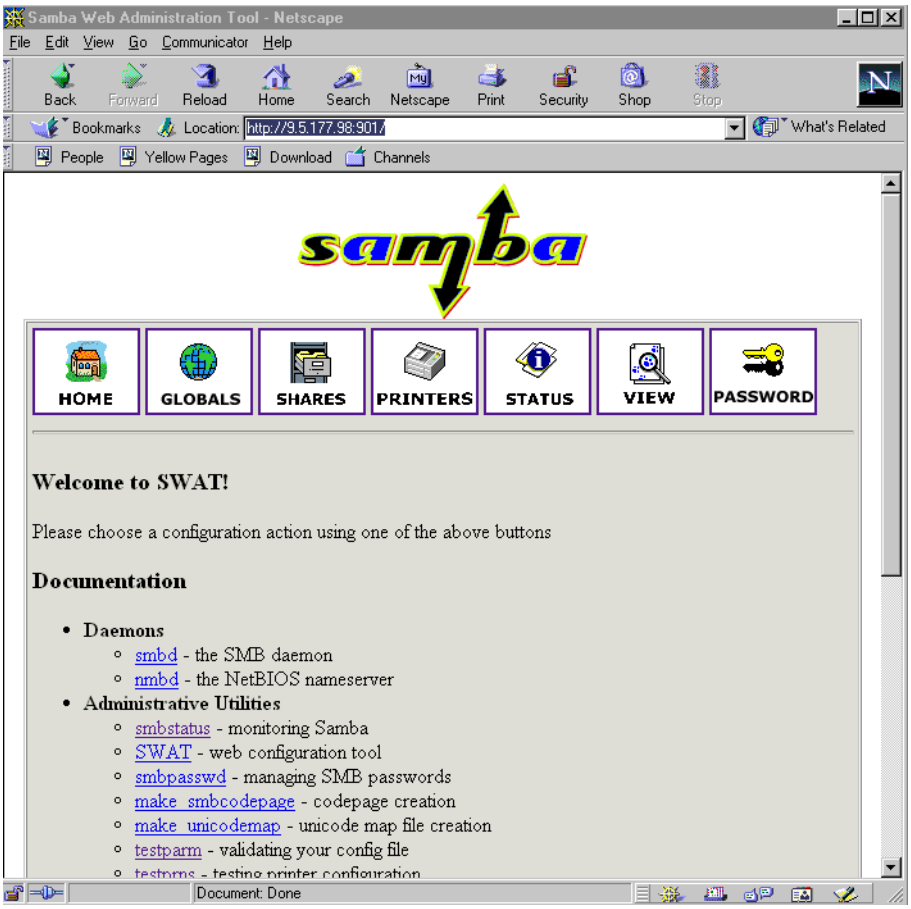


Figure 11-4 SWAT start page

From here, you can configure Samba shares and users.

11.4 Problem determination

If your Samba server does not show up on your network (it may take some time depending on how busy your network is), you may have one of several problems:

- Did you configure Samba to use WINS support? If you did and you are not using WINS (which may be the problem)?
- Can you use the Find: Computer option in the start menu and use the IP address of the server to find the Samba server? If yes, then you should check the logs found in /var/logs/log.smbd to see what problems are listed. If no problems are listed there, try restarting the samba daemons with the command **/etc/rc.d/init.d/samba restart** or something similar like **/etc/rc.d/samba restart** or **/etc/rc.d/smb restart**.

Use the **smbclient** command to determine what's wrong, because it typically gives better error messages than Windows servers do.

You can also join the Samba mailing list. The people on the mailing list are very helpful and are quite experienced. To subscribe, go to:
http://lists.samba.org/mailman/listinfo/samba

12

Network File System

This chapter provides basic information about Network File System and also provides links to other IBM Redbooks and World Wide Web resources. It includes the following topics:

- What NFS is
- Security issues with NFS
- Other resources for NFS
- Advantages of NFS
- Installing NFS
- Configuring NFS

12.1 What NFS is

NFS is a way to access file space on other systems. The NFS makes remote objects stored in file systems appear to be local, as if they reside in the local host. With NFS, all the systems in a network can share a single set of files. This eliminates the need for duplicate file copies on every network system. NFS gives users and administrators the ability to distribute data across a network by exporting local file systems, from a local server for access by remote clients, and mounting remote server file systems over local client directories. The biggest advantage of NFS is its ability to consolidate file system sharing without extra licensing issues that exist with other file systems such as the Windows file system.

12.1.1 The history of NFS

Sun Microsystems released NFS in 1984. Sun introduced NFS Version 2 in 1985. In 1989, the Request For Comments (RFC) standard 1094 describing NFS Version 2 was published. X/Open published a compatible version that was a standard for NFS in 1992. Sun published the NFS Version 3 protocol in 1993. Sun developed NFS in a UNIX environment and, therefore, many UNIX concepts were integrated into the final protocol. Yet, the NFS protocol remains platform independent. Today, almost all UNIX platforms use NFS, as do many PCs, mainframes, and workstations.

Most implementations of NFS are of Version 2, although several vendors are already offering products that combine Version 3 and Version 2. The basis for the iSeries implementation of the Network File System is Version 2. Linux can be both an NFS server and an NFS client, which means that it can “export” file systems to other systems, and “mount” file systems exported from other machines as local drives.

12.1.2 Other resources for NFS

When you determine that you need to use NFS on your network, it is a good idea to make sure you have current information concerning the version you are using. A good resource concerning the iSeries implementation of NFS is *Exploring NFS on AS/400,* SG24-2158. You can find a complete HowTo at: http://nfs.sourceforge.net/nfs-howto/

12.1.3 Two daemons: knfsd and nfsd

On Linux, there are two different implementations of the NFS daemon. One runs as a self-contained user space program. The other one is partly implemented in the kernel, and therefore, it is faster. Both daemons can be used on iSeries Linux. For iSeries users, it is very interesting that the users space-based daemon has some additional capabilities; it can map user IDs according to a map file.

12.2 Advantages of NFS

NFS allows access to data on remote hosts in exactly the same way a user would access their local files. This is made possible by a mixture of functions on the client side (that use the remote file system) and an NFS server that provides the data. This file access is completely transparent to the client and works across a variety of server and host architectures. NFS offers a number of advantages:

- Data accessed by all users can be kept on a central host, with clients mounting this directory at boot time. For example, you can keep all user accounts on one host and have

all hosts on your network mount /home from that host. If installed alongside with NFS, users can then log into any system and still work on one set of files.

- Data consuming large amounts of disk space may be kept on a single host. For example, all files and programs relating to LaTeX (a typesetting system) could be kept and maintained in one place.
- Administrative data may be kept on a single host. All administrative data can be kept in a single database and shared to the appropriate users.
- Entire trees of the file system that are the same on each and every machine can be NFS mounted from one server, where they can be administered at one central place. For example, it is possible to mount /usr over NFS. /usr contains architecture dependant files, but /usr/share contains only architecture-independent data and could even be shared between (same versions of) Linux installations on different platforms. With some adaptations, it is even possible to mount the root file system (/) over NFS, which is common practice to run to, called *thin clients*. For more information, see:
  http://www.linuxdoc.org/HOWTO/mini/NFS-Root.html
  http://www.linuxdoc.org/HOWTO/mini/NFS-Root-Client-mini-HOWTO/

12.3 Security issues with NFS

While NFS adds functionality to your network in terms of file sharing, it also adds complexity. The complexity comes in the form of added security precautions that you must take to ensure that your network is still secure.

You can find more information to help you understand the security issues and some fixes for those issues at: http://www.ibiblio.org/mdw/HOWTO/NFS-HOWTO/

Check with the distributor for up-to-date versions of the kernel and NFS packages. Another source for security issues is to subscribe to or go to the Web site:
http://www.securityfocus.com/

12.4 User IDs

It is important that the user ID (UID, the unique number of the user) needs to be the same or to be mapped to a user. In OS/400, the UID can be set in the creation of a user profile (CRTUSRPRF) or can be changed (CHGUSRPRF). In iSeries for example, the QSECOFR user profile has a UID of 0 (which is convenient because in Linux the UID of root is also 0); the QSYS user profile has a UID of 101. When changing the UID, the user cannot own any files in the Integrated File System (IFS). For general information on UIDs in Linux, see 3.6, “Users and groups” on page 103. For the iSeries, see *Exploring NFS on AS/400,* SG24-2158.

Important: The matching of UID numbers is very important for file access. Refer to *Exploring NFS on AS/400,* SG24-2158, for more information on this.

In a network of Linux hosts, a common way of keeping user IDs consistent is to use NIS (Network Information Service). This is a method to administer users on one central server (that can be backed up by so-called slave servers). On the client machines, there are no local users; instead they query the NIS server for the so-called user and password “maps”. In addition, if the Linux automounter is combined with NIS, and the home directories of the users

can automatically be mounted via NFS when a user logs in at a client machine. For more information about NIS on Linux, refer to:
http://www.linuxdoc.org/LDP/nag2/x-087-2-nis.html
http://www.linuxdoc.org/HOWTO/NIS-HOWTO/

Unfortunately there is no NIS implementation for OS/400, so that you either have to use the manual approach in keeping the user IDs consistent between the systems, or use the special mapping options that the user space NFS server offers.

Therefore, if you users exist on your OS/400 installation, it makes a lot of sense to use their user IDs when you create users on your Linux partition. This can be done with the **-u** switch of the **useradd** command. The **useradd** command is explained in more depth in 3.6.4, "Managing users" on page 104.

12.5 Installing NFS servers in Linux

First determine if NFS-related packages are installed. In your Linux command line, type the command:

```
rpm -qa | grep nfs
```

Press the Enter key. This generates a list of packages with NFS in the package name. The version of the package or packages is also listed.

Next you can find out what the installed packages contain, by entering the command:

```
rpm -qi package_name
```

Package_name would be one of the retrieved names from the query command (for example, **knfsd**). If no NFS server package is installed, then you need to install one. If the package is installed, go to 12.6, "Configuring NFS" on page 247.

To install NFS, retrieve the documentation from the distributor and follow the instructions to install from their distribution media. If the distribution has NFS installed, you could query the package information with RPM similar to the display in Figure 12-1.

```
[root@MO3H /root]# rpm -qi knfsd                                          Name
: knfsd                        Relocations: (not relocateable)
Version     : 1.4.7                             Vendor: TurboLinux
Release     : 7                             Build Date: Fri Jun  1 06:03:40 2001
Install date: Mon Oct  1 10:29:34 2001      Build Host: vafel.dev.si.tlan
Group       : System Environment/Daemons    Source RPM: knfsd-1.4.7-7.src.rpm
Size        : 476266                           License: GPL
Packager    : TurboLinux
Summary     : The kernel NFS server.
Description :
The knfsd package provides the kernel NFS server and related tools,
which provides a much higher level of performance than the traditional
Linux NFS server used by most users.
```

Figure 12-1 NFS is installed

12.6 Configuring NFS

Both the iSeries and Linux have NFS clients and NFS servers.

12.6.1 Linux as an NFS server

NFS has a configuration file that tells it what to share and who can share it. In order for a directory to be accessed by a client, it has to be exported. This is done via the /etc/exports file. The description of the file can be read by entering the **man 5 exports** command. The contents of this file determine what is shared, who can share it, and their access privileges. Here is a sample of a /etc/exports file:

```
# See exports(5) for a description.
# This file contains a list of all directories exported to other computers.
# It is used by rpc.nfsd and rpc.mountd.

/usr/local 10.5.177.0/255.255.255.0(ro)
/home/trobison 10.5.62.56(rw)
/books *.example.com(ro)
/mounts/work/ 10.5.0.0/255.255.0.0 *.example.com(ro)
```

This sets up sharing of the /usr/local directory so that anyone in the 10.5.177/24 network can read only the share, no writing or modifying of the directories or their contents. The second line set up a share so that the /home/trobison directory is shared to a specific IP address of 10.5.62.56. This share has access privileges of read and write. This person can read directories and files and also write directories and files in the shared directory. The third line defines /books as to be shared read only with all hosts in the example.com DNS domain (the * serves as a placeholder for any hostname). In the last line, we grant read only access to /mounts/work to all hosts in the 10.5 network with a 16 bit netmask, as well as to all hosts in the example.com domain, separated by a space.

Now you know how to set up an /etc/exports file on the exporting server. See 3.1.2, “Editing files using text editors” on page 88, for practical information on editing files.

So far, we have only showed the ro (read only) option. There are a number of other options you can set up in the /etc/exports file. For example, consider this entry:

```
/usr/local/share myserver.mydomain.com(rw,no_root_squash)
```

This specifies that the directory /usr/local/share is accessible to the server myserver.mydomain.com for read/write access. In addition, we allow root on the mounting system to have root access to the files on the exporting system. This is not the default, since in many (if not most) cases, it is not desirable that the administrator of one hosts can automatically access the files of another host as root. More often, write permission for normal users is required, but not for root. The default (root_sqash) is that the requests of *root* are mapped to the user nobody (UID=65534=–1).

More options are listed in Table 12-1. To some extent, the options of knfsd and the ones of the userspace NFS daemon differ. They can also differ by version, so you should look them up in the man page of exports.

Table 12-1 NFS options

Option	Description
ro	Read only. Only permits reading.
rw	Read/Write. Permits reading and writing. If both ro and rw are specified, rw takes priority.

Option	Description
root_squash	Very often, it is not desirable that the root user on a client machine is also treated as root when accessing files on the NFS server. With this option, requests are mapped from uid/gid 0 to the anonymous uid/gid.
no_root_squash	Turn off root squashing. This option is mainly useful for diskless clients.
squash_uids and squash_gids	Specify a list of UIDs or GIDs that should be subject to anonymous mapping. A valid list of IDs looks like this: squash_uids=0-15,20,25-50
all_squash	Processes all requests for access as anonymous user.
anonuid=uid	root_squash or all_squash when options are set will assign a user ID to an anonymous user request.
anongid=gid	root_squash or all_squash when options are set will assign a group ID to an anonymous user request.

Once the changes have been made, make sure the NFS server is started. Check the distributor's documentation for the how-to information. To make the NFS server aware of the changed export file run the **exportfs -arv** command to update the NFS server (or use the distributors scripts that might be supplied, like **rcnfsserver restart**). The above setup now allows other systems that support NFS to connect to this Linux system.

12.6.2 iSeries as an NFS client

For the iSeries to share files with Linux, enter **mount** on a command line. Use F4 to prompt for this command and fill in the type, shared directory or file location, and the directory the iSeries will use to mount the files as shown in Figure 12-2.

```
                          Add Mounted FS (MOUNT)

Type choices, press Enter.

Type of file system  . . . . . .   *nfs          *NFS, *UDFS, *NETWARE
File system to mount . . . . . .   /fonts

Directory to mount over  . . . .   /mnt/fonts

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 12-2 Mount command

12.6.3 iSeries NFS server

To use the iSeries as a NFS server, there are several steps to verify whether the server is started and to set up the server. In the IFS, there are several different directories and file types. The UNIX like file system is in QOpenSys, which is the directory a Linux user should use. If the files are Windows-based, the user can use the Window file directories. For verification and configuration, the user may use the command line format or Operations Navigator. For the commands, see Figure 12-3.

```
CMDNFS                  Network File System Commands

Select one of the following:

  Commands
     1. Change NFS Export                                      CHGNFSEXP
     2. Convert RPC Source                                     CVTRPCSRC
     3. Display Mounted FS Information                         DSPMFSINF
     4. End NFS Server                                         ENDNFSSVR
     5. End RPC Binder Daemon                                  ENDRPCBIND
     6. Change NFS Export                                      EXPORTFS
     7. Start RPC Binder Daemon                                RPCBIND
     8. Convert RPC Source                                     RPCGEN
     9. Display Mounted FS Information                         STATFS
    10. Start NFS Server                                       STRNFSSVR

  Related Command Menus
    11. Java Commands                                          CMDJVA
                                                                  More...
Selection or command
===>

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F16=Major menu
(C) COPYRIGHT IBM CORP. 1980, 2000.
```

Figure 12-3 Command line commands for NFS

This example uses Operations Navigator. The first step is to verify if the host servers are started:

1. Start Operations Navigator and select **Network-> Servers->TCP/IP** (Figure 12-4).

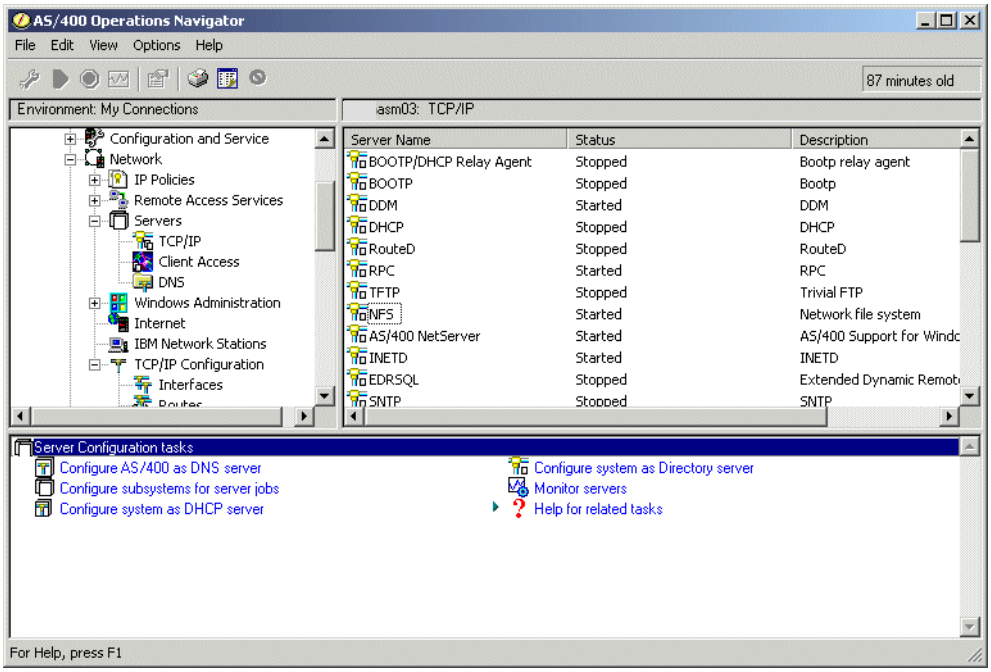


Figure 12-4 Checking the status of the NFS server

2. If the server is stopped, right-click to see the pop-up screen (Figure 12-5).

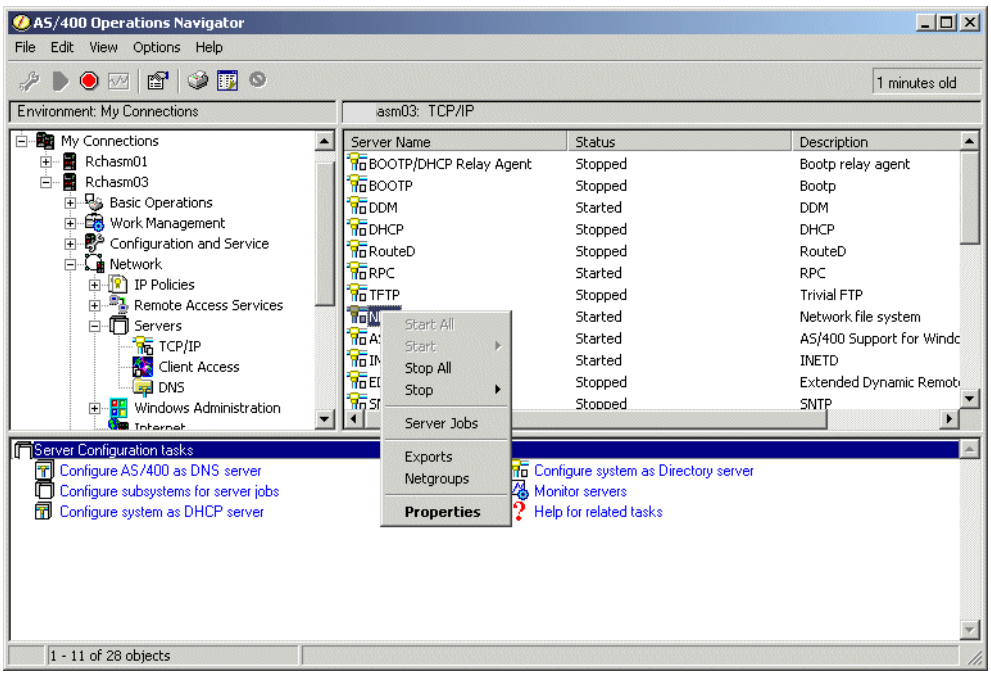


Figure 12-5 Making sure the server is started

3. To allow another machine to use the NFS service, open the properties. Check the status of the started daemons (Figure 12-6).

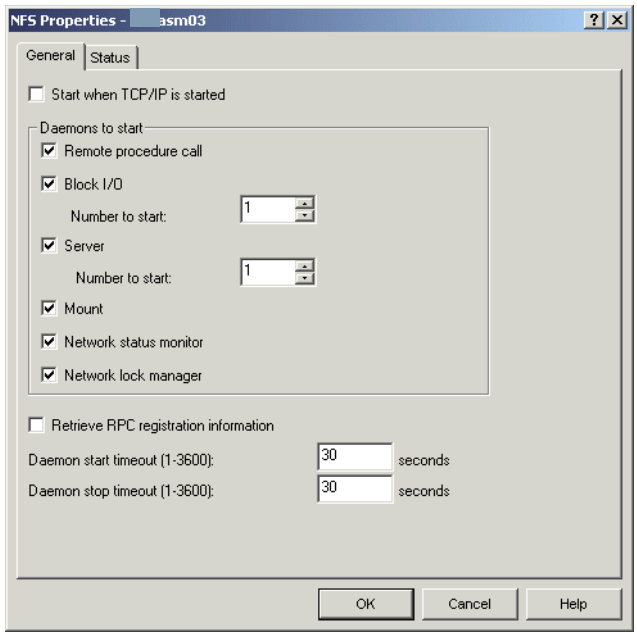


Figure 12-6 Daemons Status

4. Check the status of the started daemons (Figure 12-7).

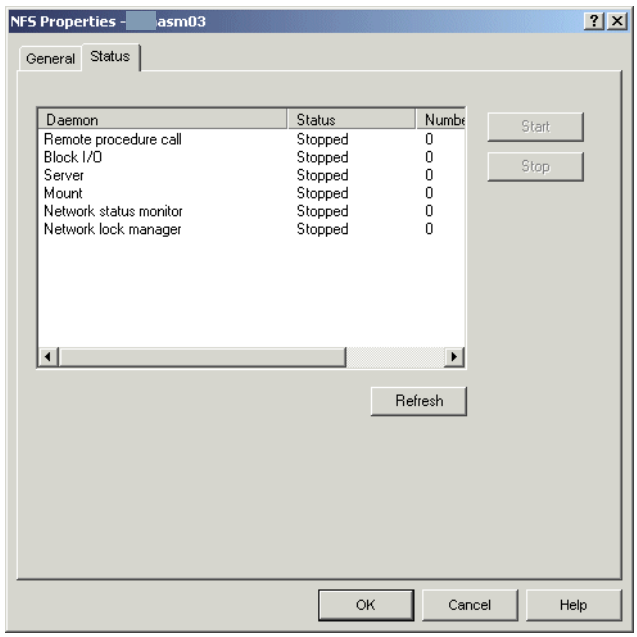


Figure 12-7 Daemons started

5. The systems are added to the export file, and authority is granted or revoked. If correct, click **OK** and return to previous screen. Right-click and choose **Export** (Figure 12-8).

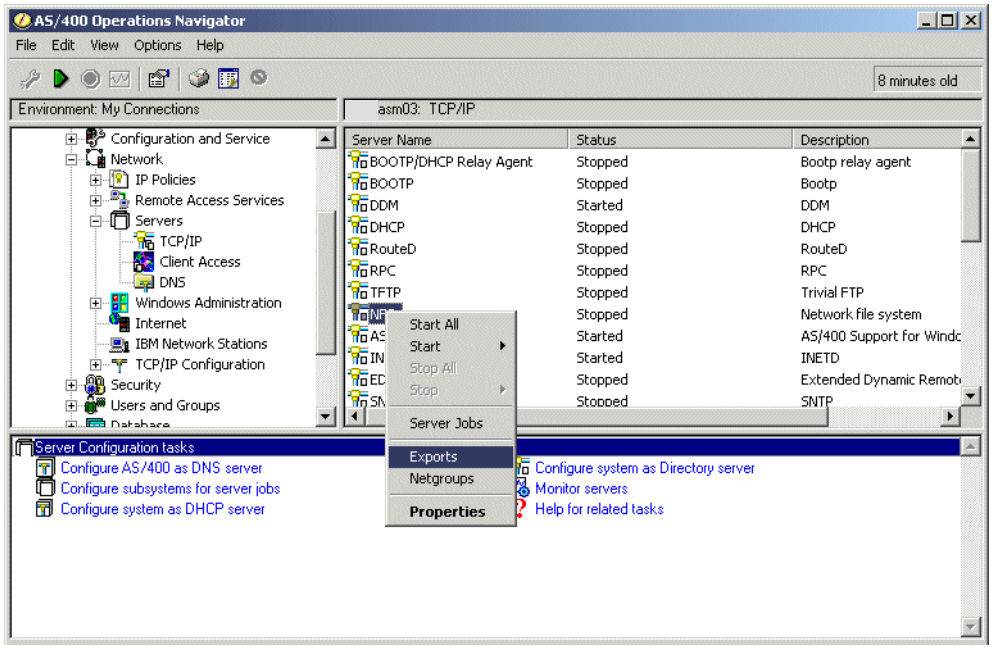


Figure 12-8 Choosing export

6. Select the files that the selected systems can use. Click the **New** button and enter the directory or directories that are to be shared (Figure 12-9).

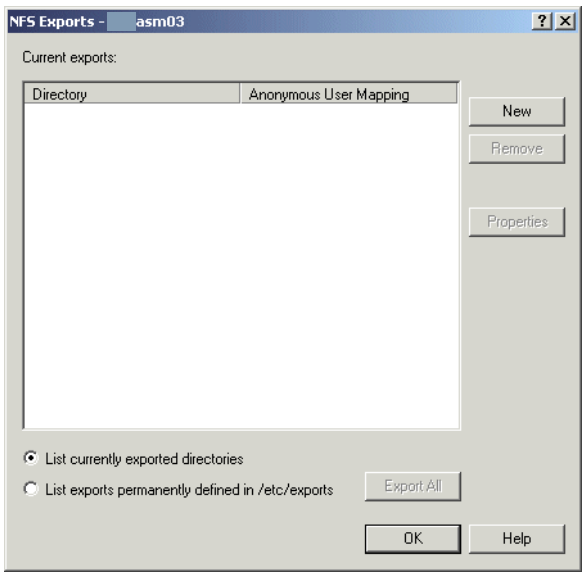


Figure 12-9 Selecting New

7. Select the files and the desired mapping (Figure 12-10).

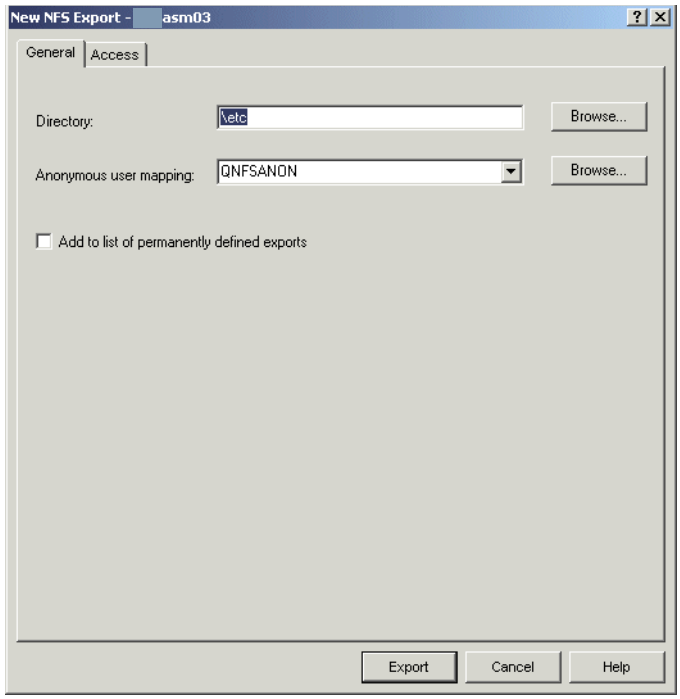


Figure 12-10 Shared file and mapping

8. To allow access to the file, select the **Access** tab. The default is **Public**. If this file is to be restricted, then click the **Add Host/Netgroup** button. A group of systems in the host table will be listed (Figure 12-11 and Figure 12-12).

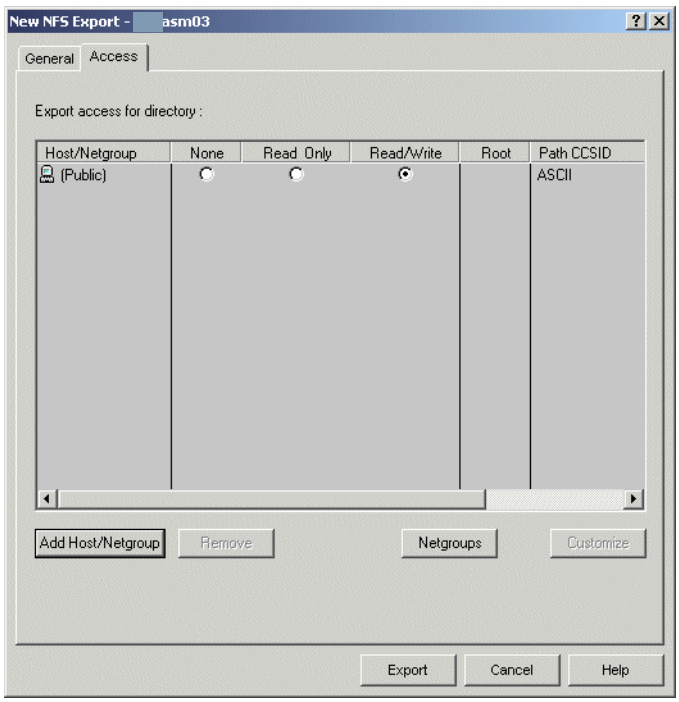


Figure 12-11 Determining who can use the file

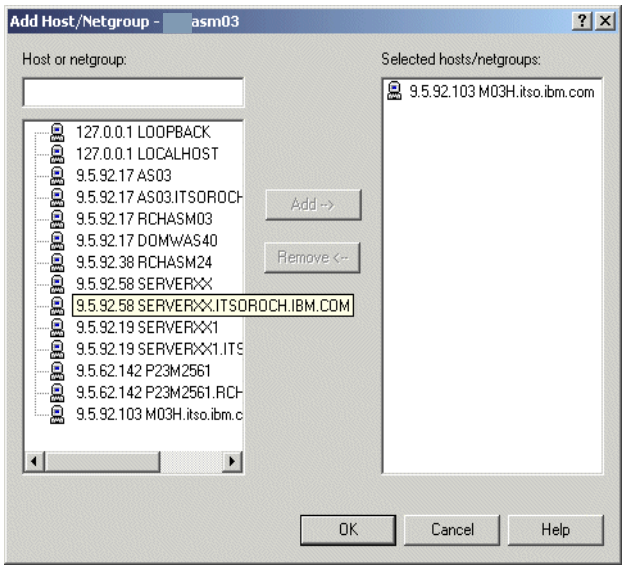


Figure 12-12 Selecting from systems who are entered in host table

9. If the system is not in the list, it must be added to the host table entries. The system can be added through the Operations Navigator. Right-click **TCP/IP configurations** and select **Host tables**. Add the system name to the host table. If the system is in the host table, select the system. When you are finished adding the host, click the **Export** button (Figure 12-13).

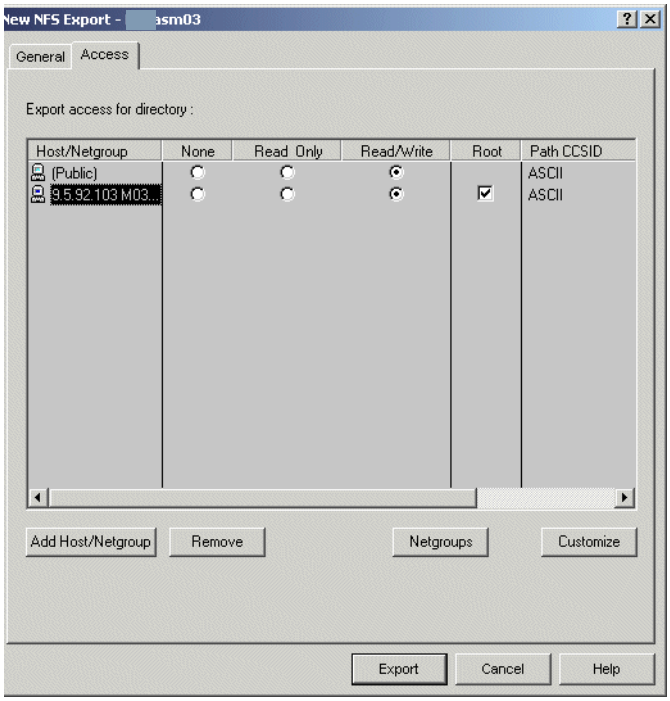


Figure 12-13 Selecting read/write authority

This completes the NFS setup on the iSeries.

12.6.4 Linux NFS client

On the Linux system, the files need to be mounted to have access to the iSeries files. This section looks at the Linux commands to mount the files.

Make sure the directory for the mount is created (the so-called “mount point”). Also make sure that the portmapper is running, a service that is needed for NFS. You can either issue the command **portmap** to start it, or use a supplied start script of your distributor, usually **/etc/init.d/portmap start**.

If needed, enter the command to create a directory to be used as mount point (this is where the file system will be mounted):

```
mkdir /mnt/images
```

Mount the directory of the iSeries with the command. Either the IP address or the system name may be used if they are entered in the host table:

```
mount 10.5.92.58:/usr/local/images /mnt/images
```

Verify that the connection has been made with this command:

```
ls /mnt/etc
```

If the response shows the files in the directory, NFS is connected.

For general help on mounting and unmounting file systems, see 3.3.5, “Mounting a disk partition” on page 99, and 3.3.6, “Unmounting a disk partition” on page 99.

See http://www.linuxdoc.org/LDP/nag/node142.html for more information on options that can be used for NFS mounts.

12.6.5 Diagnostics

The **showmount** command is a useful command that should be mentioned. It is helpful to determine which directories are exported by a host, with the **-e** switch:

```
mac@iserieslinux ~/ > showmount -e 10.5.70.90
Export list for 10.5.70.90:
/73/CD1 *
/73/CD2 *
/73/CD3 *
/73/CD4 *
mac@iserieslinux ~/ >
```

There are four directories that can be mounted by anyone.

If we replace the **-e** command-line switch with **-a**, the information that **showmount** outputs is different. It then displays whether there are hosts that have something mounted from the server in question and what they have mounted:

```
mac@iserieslinux ~/ > showmount -a 10.5.70.90
All mount points on 10.5.70.90:
10.5.70.51:/73/CD1
mac@iserieslinux ~/ >
```

This tells us that one host (10.5.70.51) has mounted the /73/CD1 directory on host 10.5.70.90.

12.7 Problem determination

One possible cause when having trouble mounting an NFS share is to change the read size and write size:

```
mount -t nfs -o rsize=1024,wsize=1024 10.5.92.58:/usr/local/images /mnt/images
```

A smaller size often works, but a larger size may give more performance. See **man 5 nfs** for more information on NFS mount options.

If you are unable to connect, make sure that the portmapper and NFS daemons are started on the remote machine.

13

Linux print support

This chapter covers some basic information about printing using Linux on the iSeries. Directly attached printers are not supported because there is no parallel or serial card support in a Linux partition. However, there are other ways to produce printed material both locally and remotely. Several standards have evolved for UNIX and Linux. This chapter covers:

- Line print requester (LPR) and line printer daemon (LPD)
- LPRng (adds additional function to LPR/LPD)
- Common UNIX Printing (CUPS)

13.1 What LPR/LPD is

Line print requester (LPR) and line print daemon (LPD) are platform independent that were added to the Berkeley version of UNIX. TCP/IP is the protocol that is used to communicate with the server (LPD) and the client (LPR). The client sends a job to the printer and the host spools the job to a queue that the client specified. The host can control multiple printers. It uses filters to determine the format of the output for the printer.

The LPR/LPD protocol is specified in RFC 1179, a standards document for UNIX and Linux. Each distributor should have documentation to set up the LPR or LPD. However, you may find additional information at: http://www.LinuxPrinting.org

13.1.1 Using LPR/LPD for printing

This native TCP/IP method of printing only requires that the printer be used has an IP address and a spool space allocated in Linux. Using this process for printing requires:

- The proper software installed on the LINUX server
- A printer with an IP address (internal or from external source)
- A method to define printers in the /etc/printcap.local file

To determine if LPD is installed on your server, type (see the example in Figure 13-1):

```
whereis lpd
```

```
[root@MO3H /root]# whereis lpd
```

Figure 13-1 Determining if LPD is installed

Press the Enter key to see a list of where the components are installed. It should look similar to the example in Figure 13-2.

```
[root@MO3H /root]# whereis lpd
lpd: /usr/sbin/lpd /etc/lpd.conf /etc/lpd.perms /usr/man/man8/lpd.8.gz
/usr/share/man/man8/lpd.8.gz
[root@MO3H /root]#
```

Figure 13-2 Installed components list

Next check the distributor's documentation to see if have they have provided a print tool to help with the printer configuration. To configure a remote printer, you need its IP address and should know the type of printer it is. It also helps to know the type of printing the printer can do. Can the printer do PostScript printing, HP-PCL 4 or 5? The exact printer may not be available in the tool. However, if the type of printer format stream is known, choose a similiar printer from the supported list, and it should work.

13.1.2 Configuring the /etc/printcap.local file manually

The printer configuration may be done manually. Edit the configuration file /etc/printcap.local on the local system using the following example:

```
# printcap.local
#
# This file is included by printconf's generated printcap,
# and can be used to specify custom hand edited printers.
```

```
REMOTE POSTSCRIPT 600X600 letter {} Postscript Default {}
F212   :\
   :sd=/tmp:\
   :mx#0:\
   :sh:\
   :rm=F212:\
   :rp=P782XWBB:\
   :if=/var/spool/lpd/
```

This printcap.local file is a database of the printers to be used on the network. A line-by-line explanation of the file follows:

- # = Comment
- `REMOTE POSTSCRIPT 600X600 letter {} Postscript Default {}` = This line defines that this is a remote printer; it uses PostScript printing on letter size paper and the PostScript printer language is the default
- `F212   :\` = This is the name of the local print queue
- `:sd=/tmp:\` = This is the shared directory for the print queue
- `:mx#0:\` = This sets the maximum job size to accept; 0 means no maximum
- `:sh:\` = Suppress headers and banner pages
- `:rm=F212:\` = This is the name of the remote host to which the printer is attached
- `:rp=P782XWBB:\` = This is the name of the printer to use on the remote host
- `:if=/var/spool/lpd/` = This is the input filter to use

A good way to find out what items can be customized in the printcap.local file is to use the **`man printcap`** command from a Telnet session.

Once the the printcap database is setup, enable TCP Printing Services for Windows NT or higher client. For Windows 95 or 98, find a Windows-based LPR/LPD program. A good place to start looking is: `http://download.tucows.com/perl/window95.html?Target=window95.html`

Then search for `LPR` in the search box. Select the software to try and download it. Follow the setup instructions. If the /etc/printcap.local file is set up correctly, printing will start.

13.2 LPRng

Line Printer Request next generation (LPRng) is an improved version LPR. You can find a definition of LPRng at: `http://www.astart.com/prng/LPRng.html`

Attention: Interoperability with older LPD implementations requires privileged access to ports. Therefore all affected client-programs ship with SUID root privileges!

This site also provides a toolkit and a documentation on LPRng. With this extension of the older LPR/LPD, the distributors have enhanced the security and the function of the standard printing on Linux. The same configuration tools will work with LPRng.

13.3 Common UNIX Printing System

Common UNIX Printing System (CUPS) uses the Internet Printing Protocol 1.1 to provide a complete printing system for Linux. CUPS is the latest in printing software to provide support for new printers and protocols while maintaining compatibility to the existing Linux environments. CUPS is free software provided under the terms of the GNU General Public License and GNU Library General Public License.

CUPS is designed around a central print scheduler. This scheduler dispatches print, manages print jobs, processes administrative commands, and sends information on printer status to local or remote programs. It can also inform users of the status of print jobs.

CUPS consists of several parts:

- Scheduler
- Configuration files
- CUPS API
- Filters
- CUPS imaging
- LPR/LPD command

CUPS is free software that can be downloaded with documentation from:
http://www.cups.org

To use CUPS, LPR, configuration tools and print filters need to be removed before you install CUPS.

There are two graphical interfaces provided for CUPS. One is a KDE interface located at the KUPS project (http://cups.sourceforge.net/kups/). The other is X printing panel (XPP), which can be located at http://cups.sourceforge.net/xpp/

The Web sites have the screen shots for both KUPS and XPP.

13.4 What to use

The decision on which printer server to use is based on the users familiarity with the different products. Also the user needs to determine if the distributor has added the package to their distribution. SuSE provides all three printer solutions in their distribution. TurboLinux provides for LPR/LPD and LPRng. Red Hat provides support for all three. The iSeries provides support for LPR/LPD. CUPS is the newest and provides the richest function. However, the list of printers that have been defined is limited. If the printer has not been defined, the user would be required to define the script. This could be a problem for a novice.

14

E-mail systems on Linux

The chapter provides resources about e-mail servers that are available on Linux for e-mail. Its purpose is to provide references to these resources, not to give a detailed description of all programs and their configuration.

The topics that are covered in this chapter include:

- E-mail basics and how e-mail works
- What a mail transfer agent (MTA) is and which ones are available in Linux
- Which programs can be used to set up e-mail in a Linux system

14.1 E-mail basics

E-mail (electronic mail) is basically the storing and forwarding of a file, with a mechanism to deliver the file to a recipient. The basic concepts include:

- **Mailbox**: A file, directory, or set of directories where e-mails are stored.
- **Mail user agent (MUA)**: A mail user agent or e-mail client is an application that the user runs directly. E-mail clients are used to compose and send outgoing messages and display file or print messages that have arrived in their mailbox.
- **Mail transfer agents (MTAs):** Mail transfer agents are software programs used to transfer messages between machines on the same or different networks. User agent programs give the message to the transfer agent, who may pass it on to another transfer agent, or possibly many other transfer agents. In the end, the receiving MUA program contacts its server, requesting the messages, and shows the e-mail to the end-user.

  Transfer agents are responsible for properly routing messages to their destination. This is the most complex portion of an e-mail system.
- **Headers and bodies**: Each message consists of two basic parts: the headers (like the envelope) and the body of the message (the letter you put inside). The headers contain information about who originated the message, the recipients, a date and time stamp, the subject of the message, and delivery stamps. You can find more information about message headers in RFC 822 and RFC 1123, Section 5. You should also refer to 16.2.2, “RFC” on page 276, for information on RFCs. A blank line always separates the headers from the body.

  The body contains the information the sender is trying to communicate. This is the “message” you are trying to send.
- **Domain Name Server (DNS)**: DNS servers are used to convert from a hostname to an IP address, and back. However, most people are not aware of the fact that DNS entries contain more information. For example, they contain information on which mail server is to be used for a certain domain, such as which mail server is willing to accept mail for a domain. So, instead of requesting an IP address for a host (which is already known by name) in a certain domain, a host could also ask “which mail server could I try to contact in order to deliver mail to the ... domain?”.

  Most of the time, a host requests an IP address for a hostname, which is specified as an “A record”. When it looks for the DNS server of that domain, it retrieves an “NS record”. For e-mail, there are specific “MX records”. These records can also be queried with the host command like this:

```
poeml@bro ~/ > host -t mx de.ibm.com
de.ibm.com mail is handled (pri=20) by d12lmsgate-3.de.ibm.com
de.ibm.com mail is handled (pri=30) by d06lmsgate.uk.ibm.com
de.ibm.com mail is handled (pri=20) by d12lmsgate.de.ibm.com
de.ibm.com mail is handled (pri=20) by d12lmsgate-2.de.ibm.com
poeml@bro ~/ >
```

  There are four mail server that would accept mail for the de.ibm.com domain. Servers with lower priority numbers (`pri=20`) would be contacted first. See `http://www.nimbledomain.com/FAQs/dns-basics.htm` for more information.

14.2 How e-mail works

E-mail works just like the post office, only much faster and no postage stamps are required.

First, the user types the address of who will be receiving the e-mail (they do this in their e-mail client, or MUA). They actually fill in the e-mail header, just as you would create an envelope for postal mail. Next, the body of the message is filled in, just like writing the letter for postal mail.

Once the user is satisfied with the message, they use the send function to send it to the server. It is actually put in their mail folder, the same as when you put your regular postal mail in the mailbox. Next, the sendmail daemon looks in all the mail folders for mail that is to be sent out. When it finds items to send, it looks at the headers to determine what to do next. If it's addressed to someone locally, on the same server, it transfers the message to the appropriate mailbox.

If it goes to some other domain, it uses a Domain Name Server to look up the address of the mail server at that domain (actually it looks for an MX record for that domain). Once it finds the MX record in the DNS for that domain, it establishes a connection to the mail daemon on that site and transfers the mail destined for that domain. The server at that domain puts the mail in the appropriate users mailbox. When the receiver of the mail logs into their mail account (using an MUA that supports one of the protocols or manners for reading mail), the mail can be read.

14.3 The whole picture

Several MTAs are available in Linux. Mail transfer agents are used to route e-mails, using a store and forward mechanism to send the messages. They use the SMTP Protocol to send messages to each other. When a message arrives in a intermediate server (that has to forward the message further on), it is temporarily stored in /var/spool/mqueue (normally).

14.3.1 Mail store directories

When the messages finally arrive at the destination MTA, it stores them in a specific file in the file system on the Linux system. There are two standards of file structures for the mail store directory:

- **mbox:** Stores the messages for a user named *username* in the */var/spool/mail/username* file. This is a flat file where all messages are stored after each other. This is a very simple structure and is older than maildir.
- **maildir:** MTAs that use this mail store directory standard create a */var/spool/mail/username* directory for each user, in which they store each message in its own file. This structure has the advantage that when a file is broken (by an error or locking condition that might happen with NFS for example), only one e-mail is lost.

Depending on the mail transfer agent, one or both of these mail store directory formats is supported.

14.3.2 Mail transfer agents

The three most popular MTAs are Sendmail, Postfix, and qmail as explained here:

- **Sendmail:** This is the most popular MTA and is estimated to handle 50% of all Internet e-mail traffic. It has been around for a very long time. Prior to the 1990s, it handled almost all Internet e-mail traffic.

However, its popularity is decreasing, partly because of security problems that were found. It also tends to be slower compared to other MTAs, but it really has all the features that an MTA should have. It can rewrite headers, do forwarding, and pass mail to specific accounts to programs that handle the mail (for example for mailing list software).

Sendmail has certainly proven to be a very good server, and it's still getting better. For more detailed information, refer to http://www.sendmail.org or one of the many books on this topic. The RPM package name is *sendmailXXXX.ppc.rpm*.

- **qmail:** The primary design goal for qmail is to replace the sendmail functionality, giving more security and performance in the process. It has very good security, mainly because all the different tasks have been split up into different programs. Refer to its Web page at http://www.qmail.org and note that the RPM package is named *qmailXXXX.ppc.rpm*. You won't find qmail on the Linux distribution, because the license basically forbids every form of binary distribution of the software.
- **postfix:** Postfix is another MTA that has been specifically designed to run fast and secure. It also consists of many programs, but not as many as qmail. It is specifically secure because it can run in a **chroot** environment. This means it cannot access any other locations in the file system other than its own directory (which implies that all libraries and configuration files it needs have to be available in that directory). If someone eventually can break in into the postfix server, they will only be able to access the files of Postfix (and no system libraries).

  Postfix appears to run very fast and can handle many more messages than sendmail does. The postfix home page is at http://www.postfix.org and the RPM package is called *postfixXXX.ppc.rpm*.

For a comparison of these MTAs, refer to the following Web pages:

- http://www.kyoto.wide.ad.jp/mta/eval1/eindex.html
- http://shearer.org/en/writing/mtacomparison.html
- http://linuxperf.nl.linux.org/mailserving/

In general, sendmail is the standard. Postfix offers good compatibility to it, while qmail is a bit less compatible.

Security issues

Make sure that when you run a MTA on a publicly available server, you disable mail relaying. Otherwise anyone from the Internet can use your mail server to send spam mail, and you would be "held responsible" because you did not protect your mail server. Today, most e-mail servers are configured to deny relaying by default. To check whether your server is relay-safe, you can try the online mail relay testers that will inform you of any holes in the MTA configuration. You can find an online mail relay tester at: http://www.abuse.net/relay.html

14.3.3 Retrieving messages

Some mail user agents running on the Linux system can read the mail store directories directly. Consequently they don't need to retrieve the messages from the server in order to read them. For example, **pine** or **mutt** can do this. Apart from this, **pine** and **mutt** also have the possibility to copy or move over the files from the mail store directory and store the messages in their own format.

POP3 and IMAP4

Most mail user agents use a protocol like POP3 or IMAP4 to download the messages. Each of these protocols is explained here:

- **POP3**: This is a protocol to retrieve e-mail from a mailbox. The purpose is to retrieve messages as a whole, but does not provide more functionality. If you want to use more advanced features like subfolders to store mail, or save your sent mail, this needs to be implemented at the client side, which typically stores all of the required data. For more information, refer to RFC 1081.
- **IMAP4**: This is a protocol to "control" e-mail in a mailbox. You can read it, but there are many more features like subfolders that are all known by the server. Of course, the e-mail client must implement the tools to access those features on the server, but it does not store the messages locally. For more information, refer to RFC 2060.

You would expect mail transfer agents to implement the POP3 and IMAP4 servers to access mail files using POP3 or IMAP4 clients. However, mail transfer agents do not provide this functionality. Specific server daemons are needed to do this.

14.3.4 POP3 and IMAP4 servers

Several POP3 and IMAP4 servers are available for Linux. They read the mail store directory themselves and make them available to POP3 or IMAP4 clients. Most of them also support encrypted connections and other protocols. Here are some examples:

- **qpopper:** This is a POP3 (no IMAP4) server from Qualcomm, but it has been released as open source. Its home page is `http://www.eudora.com/qpopper_general/`. The RPM package is called *qpopperXXX.ppc.rpm*.
- **imap:** This package is a POP3 and IMAP4 server. Its package name is *imapXXX.ppc.rpm* or *imap-2000XXX.ppc.rpm*.

You don't need to install them if you're not going to use POP3 or IMAP4 clients.

14.3.5 Fetchmail

Fetchmail is a full-featured POP3/IMAP4 mail retrieval daemon. This means it acts as a POP3 or IMAP4 client. Instead of storing messages (like normal e-mail clients do), it delivers the mail to another MTA using SMTP, most probably the mail transfer agent on your local PC. In a way, it is a POP3/IMAP4 to SMTP gateway.

Fetchmail is a very handy tool if you want to have multiple pop accounts and want them to be stored in your local mail file. This is what it is primarily used for, but there are many more possible scenarios where it can fit in.

For more information, refer to the fetchmail home page at:
`http://www.tuxedo.org/~esr/fetchmail/`

The RPM packages you need for fetchmail are called *fetchmailXXX.ppc.rpm* and the configuration tool for fetchmail can be found in the *fetchmailconfXXX.ppc.rpm* package.

14.3.6 Mail user agents (e-mail clients)

Examples of popular e-mail clients in Microsoft Windows are Outlook Express, Netscape Messenger, or Eudora Pro. They normally support IMAP4 and POP3.

In Linux, there are both text-based and graphical based e-mail clients. Some example of text-based clients are **`pine`**, **`elm`**, or **`mutt`**. The most popular graphical e-mail clients are probably Netscape Messenger, balsa (GNOME), or kmail (KDE).

The Lotus Notes client can also be a Mail User Agent, although communication between the Notes Client and the Domino Server are normally done with other protocols than we discuss.

14.3.7 Mailing list software

It's also possible to set up mailing lists using specific mailing list programs. This enables you to configure mailing lists and mailing list subscribers. Several programs are available, the most popular ones are:

- **majordomo:** This is the most popular mailing list program. It supports various kinds of mailing lists: closed or open, moderated or non-moderated, hidden or non-hidden mailing lists, etc. The subscription process is fairly easy although not effortless. You have to create a message with a specific SUBSCRIBE command as its body. (There are some Web front ends available, but they are not included in majordomo itself.) The RPM package name is *mdomoXXX.ppc.rpm* and its Web site is: http://www.greatcircle.com/majordomo/
- **mailman:** This mailing list package is very flexible and provides *Web pages* to all configurations for an administrator as well as subscribe/unsubscribe for users, which is an advantage if you want your mailing list easily accessible for non-technical users. This is easier than subscribing by sending an e-mail (which is also supported by mailman).

  Apart from that, mailman provides you with similar features as majordomo. The RPM package is called *mailmanXXX.ppc.rpm*. Visit its Web site on http://www.list.org/ for more information.

For more information on various other mailing list packages, see "The Mailing List Software Inventory" at: http://listes.cru.fr/sympa/robots.html

14.4 Problem determination

There can be so many different causes of problems. If you cannot send mail to your Linux server, try to find the Frequently Asked Questions page of your favorite MTA program or look at your distributors documentation.

Apart from misconfiguring the MTA, the most likely the underlying cause of mail delivery problems is a wrong DNS configuration.

15

System time (NTP) and job scheduling (cron and at)

This chapter provides information on time-related tools. One of them is NTP, which is used to correct the time on iSeries Linux when it is inaccurate. The other tools are **cron** and **at**, which can be used to schedule jobs. This chapter examines the following topics:

- The inaccurate clock that Linux uses
- Estimating its accuracy
- Correcting time instantly using **ntpdate**
- Correcting time gradually using **xntpd**
- Scheduling jobs using **cron**
- Scheduling jobs using **at**

15.1 Accuracy of time in iSeries Linux

The clock of the iSeries is accurate because it is driven by a service processor that was specifically designed for long term accurate time-information. OS/400 itself uses that clock to ensure time in OS/400 partitions is very accurate. Linux, however, does not recognize that clock, and it uses the regular PowerPC clock, which unfortunately is often inaccurate.

The following sections discuss solutions for improved accuracy.

15.1.1 Determining the problem

The following output illustrates how fast the clock degrades. Each time the current time is shown, the difference between the current time and a time server is calculated. The time server should have the exact time by a hardware device that receives the time from atomic clocks, or by synchronizing itself with another server that has this device.

```
[root@turbo init.d]# ntpdate -q time.mynetwork.com
server 10.5.39.33, stratum 2, offset -0.653686, delay 0.03067
 3 Oct 14:25:57 ntpdate[5536]: step time server 10.5.39.33 offset -0.653686 sec
[root@turbo init.d]# ntpdate -q time.mynetwork.com
server 10.5.39.33, stratum 2, offset -0.963649, delay 0.03882
 3 Oct 14:38:02 ntpdate[5566]: step time server 10.5.39.33 offset -0.963649 sec
```

You can see that in a time frame of 15 minutes, the clock degraded 0.3 seconds, which is quite inaccurate. And it keeps degrading. So after 30 minutes, the difference is 0.6 seconds, and so on. The **ntpdate** command is used for this quick check. You can find more information on **ntpdate** in the following sections.

15.1.2 System clock and hardware clock

When we talk about time, we mean the time as stored in the system clock. The hardware clock is not used at all while the system is running and is only used to initialize the system clock when the system is booted (or IPLed).

You can show the current system clock using the **date** command as in:

```
root@iserieslinux:/etc # date
Wed Oct  3 15:00:30 CDT 2001
```

Retrieve the current hardware clock using the **hwclock** command:

```
root@iserieslinux:/etc # hwclock
Wed Oct  3 15:01:13 2001  -0.279296 seconds
```

hwclock also gives you the difference it encountered with the system clock, which is *-0.279296* in our example.

To adjust your hardware clock, you should set it from the system clock. You can do this by using **hwclock --systohc**. It might be needed to explicitly specify whether the hardware clock is in local time or in Greenwich Mean Time (GMT). Use **hwclock --systohc --localtime** or **hwclock --systohc --utc** for these purposes. Server systems usually have their clock set to GMT.

15.2 Network Time Protocol (NTP)

NTP is a protocol that enables you to synchronize the time of your system in a variety of ways. There is an NTP daemon (called **xntpd**) that is active all the time and contacts other servers to find out the exact time. The more servers it can contact, the better the time estimation will be. **xntpd** can also be contacted by other systems to send its own time so other systems can adjust their time.

xntpd is a very advanced tool using special algorithms, and it does not set the clock in one step. It sets the clock gradually so there are no real jumps in time. This ensures that you don't encounter two moments in the log files with the same time, for example, when the clock is set back a few seconds in one step. **xntpd** gives hints to the Linux kernel on how to adjust its time, so its clock is constantly fine-tuned.

For detailed information about this protocol, refer to http://www.ntp.org where you can also find a list of public time servers that you can use if there is none in your local network.

The Network Time Synchronization Project home page at http://www.eecis.udel.edu/~mills/ntp.htm offers more practical information, including explanations of commands. It is certainly worth a visit.

15.2.1 ntpdate

To retrieve the current time of an NTP server, use the following command (the **-q** options stands for *query only*):

```
root@iserieslinux:/etc # ntpdate -q time.mynetwork.com
server 10.5.39.33, stratum 2, offset -0.002179, delay 0.03882
 3 Oct 15:42:38 ntpdate[9552]: adjust time server 10.5.39.33 offset -0.002179 sec
```

Here *time.mynetwork.com* is the NTP server that we used when writing this book.

Setting time with ntpdate

ntpdate can also set the system time from a specified NTP server. Just specify **ntpdate** without any options, for example:

```
iserieslinux:~ # ntpdate time.mynetwork.com        << set the time
 9 Oct 17:01:04 ntpdate[30120]: step time server 10.5.39.33 offset 59.524240 sec

iserieslinux:~ # ntpdate -q time.mynetwork.com     << query the time to see the difference
server 10.5.39.33, stratum 2, offset -0.008414, delay 0.02829
 9 Oct 17:01:13 ntpdate[30123]: adjust time server 9.5.39.33 offset -0.008414 sec
```

You probably want to synchronize the hardware clock afterward. Refer to "System clock and hardware clock" on page 268 for more information.

Remember that when using **ntpdate**, the clock is set instantly, causing possible overlaps or leaps in time. The **ntpdate** function can be used for initialization, for example, every time the system is restarted, to set the initial time. In general, it is best to use **xntpd** to keep your time accurate.

15.2.2 xntpd

To set the time of your machine and have it corrected continuously, use the NTP daemon **xntpd**. Obtain the package *xntp* or *xntp3* or something similar, which should be included with your distribution.

There is one configuration file that should be modified. Add a line in */etc/ntp.conf* for each NTP server you want to use. (For more information on how to edit a file, refer to 3.1.2, “Editing files using text editors” on page 88.) For example, we used *time.mynetwork.com* and *tock.mynetwork.com*, so we added the following two lines:

```
server time.mynetwork.com
server tock.mynetwork.com
```

Make sure **xntpd** is started when your system is started. This can be done by adding a link named *S15xntpd* in the appropriate */etc/init.d/rcX.d* directory to `../ntpd`, where X is the run level in which you want it to start. If this sounds unfamiliar to you, use the distributors own provided tool to accomplish the autostart of the NTP daemon.

Note: **xntpd** does not start adjusting the time immediately when it’s started. It takes a few minutes to acquire stable information and then starts adjusting gradually. If there is a large difference between the clock stated time and the actual time (for example, more than half an hour), it’s a good idea to manually initialize the time using **ntpdate** before starting **xntpd** (see 15.2.1, “ntpdate” on page 269). This is because it might take too long before the clock is synchronized since corrections occur gradually.

xntpd itself is an NTP server to other NTP clients in your network. Typically you synchronize the time of your server with a public time server, and then all hosts in your network can use your time server to synchronize. This is needed to reduce load of the public time server and to reduce the bandwidth you use to the Internet. Most of the time, because of the firewall, the NTP clients in your local network don’t have a direct connection to the Internet anyway.

15.2.3 ntptrace

Use the **ntptrace** command to see whether your host is synchronized to one or several time servers. The **ntptrace** determines from where a given Network Time Protocol server gets its time and follows the chain of NTP servers back to their master time source. If no arguments are given, it starts with localhost, for example:

```
iserieslinux:/dev # ntptrace
localhost: stratum 3, offset 0.000008, synch distance 0.11079
tock.mynetwork.com: stratum 2, offset -0.003359, synch distance 0.03815
rchntp.otherdomain.com: stratum 1, offset -0.004399, synch distance 0.00000, refid 'GPS'
```

This output shows that the Linux system (localhost) is synchronized with *tock.mydomain.com* and that NTP server again is synchronized with *rchntp.mydomain.com* that uses a GPS device to retrieve its time.

15.2.4 ntpq

The **ntpq** utility program is used to query NTP servers about their current state and to request changes in that state. It can be used as an important debugging tool to see which peers could not be reached, for example:

```
iserieslinux:/dev # ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
+time.mynetwork.c rchntp.otherdom 2 u   11  128  377    7.304   -0.521   3.029
*tock.mynetwork.c rchntp.otherdom 2 u   14  128  377    8.467    2.466   4.329
```

You can see the two time servers we specified in the */etc/ntpd.conf* file. The currently selected peer is marked with an asterisk (*), while additional peers designated acceptable for synchronization, but not currently selected, are marked with a plus sign (+).

See http://www.eecis.udel.edu/~ntp/ntp_spool/html/debug.htm for more information.

15.2.5 Integration with OS/400

Although NTP works very well, it is important to make sure you are using a good NTP server to retrieve the correct time. Some servers sometimes have no fixed Internet connection, or the iSeries Linux system cannot directly connect to public time servers on the Internet. If this is the case, you need a different solution.

Since OS/400 uses the accurate iSeries clock, it would be a good time server for isolated systems to synchronize with. You need a NTP server program on your OS/400 partition to synchronize with. This is not a standard product with a Product ID (PID). Therefore, you need to contact the Customer Technology Center to access this NTP server. Actually it's a Simple Network Time Protocol (SNTP) server and client. This means it does not use the complicated algorithms a real NTP package is based on to calculate the time using several other time servers. But they are adequate for most environments.

For more information on the SNTP server (and SNTP client), contact the Customer Technology Center (CTC) via their Web site at: http://www.ibm.com/iseries/service/ctc/

Or you can directly go to the SNTP page of the CTC at:
http://www.ibm.com/iseries/service/ctc/details/a2_sntp.htm

15.3 cron

cron is used to execute a repetitive task on a specific moment in time. Scheduling information is placed in the /etc/crontab file, using the following format:

```
minute hour day month year username command
```

You can specify each time component as an integer number. For example, use 1 through 12 for the months of January through December. Or you can specify one or more components as "*" characters, which will be treated as wildcards. For example, * in the month component means the command will run at the given day and time in every month. (For information on how to edit files, refer to 3.1.2, "Editing files using text editors" on page 88.) The username parameter should be the name of the user that executes the command, which might be different for security or authority reasons. For example, a backup process must be run by a user that can read all files in all directories.

There is also the possibility for each user to put a crontab file in their home directory. For more information, refer to http://www.linuxdoc.org

Or you can run either of the following command, which will also provide more information:

```
man crontab
man 5 crontab
```

However, you don't have to use this text file; there is a simpler approach. Several directories exist in /etc where you can place a script file that will automatically, executed on specific moments in time. These directories are called:

- /etc/cron.hourly
- /etc/cron.daily
- /etc/cron.weekly
- /etc/cron.monthly

The script is then executed every hour, day, week, or month.

There are some default commands that are run daily, for example **updatedb** to update the filename database of **locate**. (For information on **locate**, see 16.5.1, “Finding a file with find and locate” on page 277.)

15.3.1 Updating the hardware clock every hour

cron, for example, can be used to run a script every hour that will write the system clock into the hardware clock. (See 15.1.2, “System clock and hardware clock” on page 268.) This can be important if you run **xntpd** and want the time to be accurate when the server is restarted.

To accomplish this, create a script called **sethwclock** (for example) and place it into */etc/cron.hourly* with the following content (see 4.1.3, “Scripts” on page 139):

```
#!/bin/sh
hwclock --systohc --localtime
```

Afterwards, make the script executable by running:

```
chmod +x sethwclock
```

15.4 The at daemon

For one-time jobs that you want to run at some specific moment in the future, you can use **at** to schedule them.

The at daemon (**atd**) has to be running in order for the commands to be executed finally. You can find out whether it is running by issuing the **pidof atd** command. If the command prints out a number, the at daemon is running. If not, refer to your distributors documentation for information on how to start it.

For example, to launch the script named *run_backup* in exactly three hours, run:

```
iserieslinux:~ # at now + 3 hours
warning: commands will be executed using /bin/sh
at> run_backup
at> <Ctrl-D>
job 4 at 2001-10-09 17:25
```

See the **at** man page for more detailed information (run **man at**). You can list all queued jobs with **atq** and remove jobs with **atrm**.

16

Help information

This chapter provides information on additional resources that can be used to obtain help with Linux. It includes the following topics:

- Online help
- Additional information on the system
- Resources on the Internet
- Related redbooks
- Finding a file on the system

16.1 Online help

Several commands can be used to find information on your Linux system.

16.1.1 Manual (man) pages

Manual (known as man) pages are installed on every UNIX (Linux) system. They are read using the **man** command. You can enter the following statement to obtain information about how to use the **man** command:

```
man man
```

Man pages are divided into the following sections:

- Executable programs or shell commands
- System calls (functions provided by the kernel)
- Library calls (functions within system libraries)
- Special files (usually found in /dev)
- File formats and conventions, for example, /etc/passwd
- Games
- Macro packages and conventions (for example, man(7), groff(7))
- System administration commands (usually only for root)
- Kernel routines [Non-standard]

Usually there is only one man page for the name you enter. However, some names have man pages in two or more sections. For example, the following command shows a list of all installed man pages for the keyword "`write`":

```
man -f write
```

In this case, there were two man pages: one in section 1 and one in section 2. The following command explicitly selects the man page from section 2:

```
man 2 write
```

Reference to a man page in a particular section may look like this example:

```
see man write(2)
```

Tip: These are the commands you need to know to navigate inside man pages (in fact, they are displayed with the text pager **less**, so the man page of **less** can tell you more key bindings):

g	Top
G	Bottom
/<keyword>	Search for <keyword>
n	Search next occurrence
N	Search previous occurrence
q	Quit

Apart from the classification into sections, the collection of man pages is relatively unorganized. The **apropos** command searches the manual page names and descriptions for a keyword.

For example, you might want to search for man pages related to the Samba daemon **smbd** as follows:

```
$ apropos smbd
smbd                 (8)  - server to provide SMB/CIFS services to clients
testprns             (1)  - check printer name for validity with smbd
```

The results show that besides **smbd**, the **testrpns** man page is related.

16.1.2 info

The structure of info is different from man in that it has a tree structure through which you can navigate, depending on which program you use for browsing the info pages.

GNU textinfo is invoked with:

```
info
```

This produces a black and white text display. In contrast, the following line is an info reader that adds color and lynx-style navigation (right arrow to follow link, left arrow to go to upper level), but does not have name completion:

```
pinfo
```

Emacs is pretty comfortable for reading info pages. If you use emacs, you can use it as your info browser by typing `ctrl-h i` inside emacs. If you prefer vi, you'll use plain info anyway. You can also read the html versions of the info pages with your favorite browser.

16.1.3 help

help is a built-in shell (that is, a command that is part of the shell) and gives help about the other functions that are shell built-ins. You invoke it by entering:

```
help
```

Among the different shells, only bash offers help; others usually say "`help: not found.`"

16.2 Additional information

There is an enormous amount of additional information to be found in various documents. Ranging from simple to very technical, you can to find answers to your questions through the following sources.

16.2.1 Howto

Howtos should reside in the /usr/doc/howto directory. They should exist in (gzip compressed, plain ASCII) text format and probably also as HTML. To read the compressed ASCII format, it is not necessary to uncompress the files first because **less** does it automatically:

```
less /usr/doc/howto/Highly-Cryptic-And-Powerful-HOWTO.gz
```

Howtos vary somewhat in style, from collections of pointers to further information to tutorials, and can be used in the following manner. Suppose you want to set up some service on your system that you've never installed before. If you do not have a package from a Linux distribution that does the installation and configuration automatically, then you can consult the appropriate howtos.

Howtos also often provide useful examples.

16.2.2 RFC

A Request For Comments (RFC) document is also a source of information. If you do not have the RFCs local on your machine, go to the following site: http://www.rfc-editor.org/

If you installed a complete Linux distribution, you should have about 2700 files named *rfc*.txt.gz* in the /usr/doc/rfc/ directory. In this case, the asterisk (*) stands for a number from 1 to 2792 (at the time of writing).

The *rfc-index.txt.gz* file contains an index (by number) of what is covered in each RFC. The entry 2696 reads as follows:

```
2696 LDAP Control Extension for Simple Paged Results Manipulation. C.
Weider, A. Herron, A. Anantha, T. Howes. September 1999. (Format:
TXT=12809 bytes) (Status: INFORMATIONAL)
```

An arbitrarily chosen paragraph reads:

```
If the page size is greater than or equal to the sizeLimit value, the
server should ignore the control as the request can be satisfied in a
single page. If the server does not support this control, the server
MUST return an error of unsupportedCriticalExtension if the client
requested it as critical, otherwise the server SHOULD ignore the
control. The remainder of this section assumes the server does not
ignore the client's pagedResultsControl.
```

This is complex information about a rather specific topic, so less experienced users may find using RFCs a bit daunting, but they can be useful. For example, if an RFC exists about a protocol you want to program for, it may give you all the information you need in a well-structured form.

16.3 The Internet

The Internet contains numerous sites that have the kind of information you are interested in. We recommend you bookmark a few favorites and check for newsgroups and forums in your interest area that will be helpful to you.

- A good Web site that has books available for download is at: http://www.linuxdoc.org

  Several of the books available for download include:

  - Linux From Scratch
  - Linux Kernel Internals
  - Linux System Administration Made Easy

- The official IBM iSeries Linux Web site: http://www.ibm.com/iseries/linux
- IBM Linux on PowerPC Project is a “developer-minded” site that provides PowerPC patches for kernel sources, etc. Low-level programming experience is required: http://oss.software.ibm.com/developerworks/opensource/linux/projects/ppc/
- Linux Technology Center mainly has technical contributions about Linux and IBM in general: http://oss.software.ibm.com/developerworks/opensource/linux/
- LinuxPPC.org is another technical site. Although it is more application-oriented (instead of kernel oriented), it contains information about PowerPC-specific builds of various open source programs: http://linuxppc.org
- This Web site offers an on-line forum, news, and general information for the iSeries Linux community: http://www.iserieslinux.com

16.4 Books

There are plenty of books about Linux and related subjects that differ widely in style and quality. Luckily, choosing the right one requires no wizardry. By reading online recommendations and reviews (online bookstores usually have them), you can choose the right book for you. As you're likely to discover, for a particular topic, an authoritative book usually exists.

16.4.1 IBM Redbooks

There are several IBM Redbooks that contain related information. Here are some examples:

- *TCP/IP Tutorial and Technical Overview*, GG24-3376
- *Linux for S/390*, SG24-4987
- *Red Hat Linux Integration Guide for IBM @server xSeries and Netfinity,* SG24-5853
- *Implementing Linux in your Network using SAMBA*, REDP0023
- *Exploring NFS on AS/400*, SG24-2158
- *Slicing the AS/400 with Logical Partitioning: A How to Guide*, SG24-5439

16.5 Useful tools

This section discusses useful tools. Very often you know the name of the file, but not where it is located. Or you might know the name of the command, but don't know whether it's a built-in shell command or a separate program and from where that program is executed.

16.5.1 Finding a file with find and locate

To access a list of all files (and directories) whose names contain "blah" you could use the **find** command, as follows:

```
find / -name "*blah*" -print
```

This goes through your entire file system and finds all the files that are specified:

```
/usr/X11R6/include/X11/pixmaps/mini.blah.xpm
/usr/X11R6/include/X11/3dpixmaps/normal/blah_3d.xpm
/usr/X11R6/include/X11/3dpixmaps/small/small.blah_3d.xpm
```

However, there are two reasons why you do not really want to use **find**:

- It causes a massive system load.
- It takes a long time.

Instead, you can set up a database of filenames and use the **locate** command to access that database, as follows:

```
locate blah
```

The output is the same as above. Updating the database is usually done daily by a **cron** job that launches the respective process, **updatedb**. **updatedb** can also be intelligent and, for example, spare outfile systems that are mounted via the network when searching. Note, however, that files created after the last update will not be found using **locate**.

16.5.2 Determining the type of commands

Commands can be of different types. They can be executable files on disk, shell built-ins, aliases, functions, or keywords. To find out what type a command is (for example, **pwd**), you can enter:

```
$ type pwd
```

This returns the answer:

```
pwd is a shell builtin
```

But wasn't there an executable named **pwd** in **/bin**? If this is the case, then the following command format will clarify the matter:

```
$ type -a pwd
```

This returns the answer:

```
pwd is a shell builtin
pwd is /bin/pwd
```

So there are actually two **pwd**s. To execute the binary, use the full path:

```
/bin/pwd
```

To determine the type of **type**, you enter:

```
type type
```

This returns the answer, courtesy of bash:

```
type is a shell builtin
```

16.6 Other resources

For technical support or training, IBM has several contact points.

E-mail contact

For technical, marketing, planning, and other questions related Linux on iSeries, send an e-mail to rchlinux@us.ibm.com

Linux on iSeries education

The iTC group in Rochester, Minnesota, runs hands-on Linux on iSeries education. You can check the Linux on iSeries Web site at http://www.ibm.com/iseries/linux for more information. To get more details on this education, send e-mail to rchlinux@us.ibm.com

A

Application scenarios

This appendix provides some examples of possible application scenarios where Linux on iSeries is used in cooperation with OS/400 to accomplish tasks that cannot be accomplished by using one of the operating systems alone. When looking at these examples, the added value of Linux running on an iSeries server can be discovered. This appendix includes the following situations:

- A theoretical (though realistic) situation, where some Linux partitions cooperate with OS/400
- An example of a real world application using the above situation
- A situation where clustering is used to give the ability to fail over to another hosting partition in case of a planned or unplanned outage

Base scenario

When deploying Linux on an iSeries server, the same base configuration is going to be used each time. Of course, Linux is run in an LPAR, so you always need an OS/400 primary partition. When you have a hosted Linux partition, the primary partition could be the hosting partition. But when the resources are available, we advise that you set up an independent OS/400 partition that will provide the hosting functionality for the Linux partition.

The base scenario would, therefore, be an iSeries server with a heterogeneous combination of Linux and OS/400 partitions,

All-in-one-box solution

It is now possible to deliver a reliable all-in-one-box solution to end users. Figure A-1 shows an example of such an implementation, in which each operating system is used for its intended purpose. OS/400 is definitely the best platform for business applications. Through the years, it has proven to be very reliable. Linux on the other hand, has proven to be a very stable operating system for (inter) networking in general, like providing firewall functions.

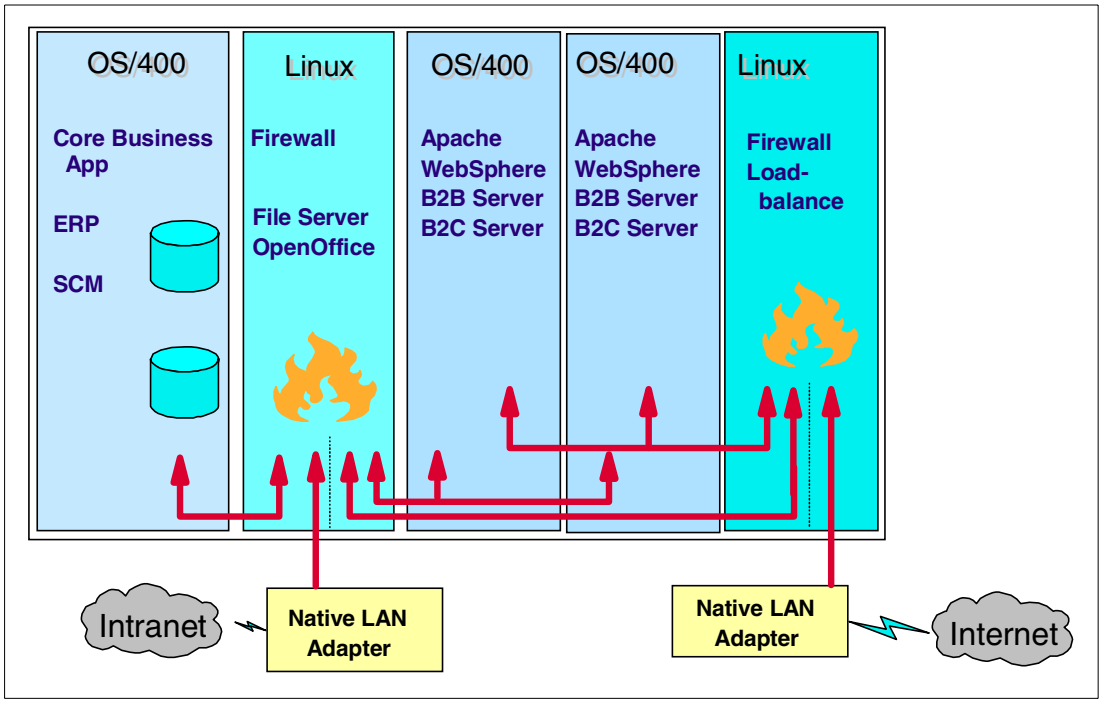


Figure A-1 All in one box example

In this example, LPAR is leveraged to support numerous operating environments on one iSeries server. Typically environments like firewalls and Internet servers are installed on their own servers. Figure A-1 shows one iSeries server can support five different servers.

This example also shows how virtual and direct I/O in a Linux environment can be leveraged. The firewall that interfaces with the Internet has a dedicated LAN adapter. The other resources needed by this partition (disk, tape, CD-ROM, and LAN) are virtual. Virtual LAN is also used to provide a direct connection to the Web serving partitions.

Two Web-serving partitions running a Commerce or Business-to-Business application are used to provide a load balancing and high availability environment. The Web-serving partitions are then connected to a second firewall via a different virtual LAN connection. The multiple virtual LAN connections provide another level of security between the outside and inside environments.

The second firewall uses virtual I/O disk, tape, CD-ROM, and LAN resources. It does not have any direct I/O devices except for the native LAN. It could be possible to connect the native LAN adapter to the first OS/400 partition. The advantage of this approach (with the native LAN adapter connected to the Linux partition) is that even internal LAN computers must pass the firewall. This means you can even stop access to the core business software for some parts of the internal network or log activity to that system.

The OS/400 partition runs the line of business applications and contains the data on which the Web applications are based.

These five partitions could be set up on a small iSeries server with two processors (for example, a Model 270). The OS/400 and Linux partitions could share these two processors and resources can be moved between the partitions based on the workload demands. This same solution could be set up on a very large iSeries server with 24 processors.

Intranet application server

You can use the second Linux system (the one connected to the internal LAN) as an X Windows Server. Using this could enable you to do application serving to the internal network. You simply have to install an X server on all PCs in the internal network and then install all programs you need at the server side once. When upgrading these applications, simply upgrade that server side.

Several programs are available that could be available in a business environment, but the most important one is probably OpenOffice. See Chapter 5, “X Windows and OpenOffice” on page 165, for more information.

There is no plan to release a Linux version of the Notes client. This was stated on the Web at: http://www.lotus.com/developers/itcentralnew.nsf/allpublic/2D96D32F0ED19D26852569DD0067B3D0?opendocument

Windows NT domain controller

Using Samba (Chapter 11, “Samba” on page 231), you can achieve complete file and printer sharing for a Windows NT domain (or a Windows workgroup of course). There even is the possibility to run a domain controller for a Windows NT domain. At the time this redbook was written, the general beta version of Samba 1.2 was available. This version officially implements a domain controller (this function was around for a long time in the alpha version).

Web server with Apache and Hypertext Preprocessor (PHP)

Instead of running all Web server tasks in OS/400, you could also do this in Linux partitions, using the Apache Web server on Linux (see Chapter 6, “Apache HTTP server on Linux” on page 183). This can be combined with PHP, which is a strong way to embed scripts at the server side to accomplish a dynamic Web site with user profiles, log on, session management, database access, etc. The access to the OS/400s DB2 database can be accomplished by using the Java Toolbox for iSeries. Learn more about the IBM Toolbox for Java and JTOpen at: http://www.ibm.com/servers/eserver/iseries/toolbox/

The ODBC drivers for the DB2 database is a work in progress right now, but it will be available some time after the general availability release of Linux on iSeries.

Magic eMerchant

Magic ported a part of their CRM eMerchant application to Linux on iSeries. Magic eMerchant is a business to business (B2B) and business to customer (B2C) e-commerce application. An overview of this solution is shown in Figure A-2.

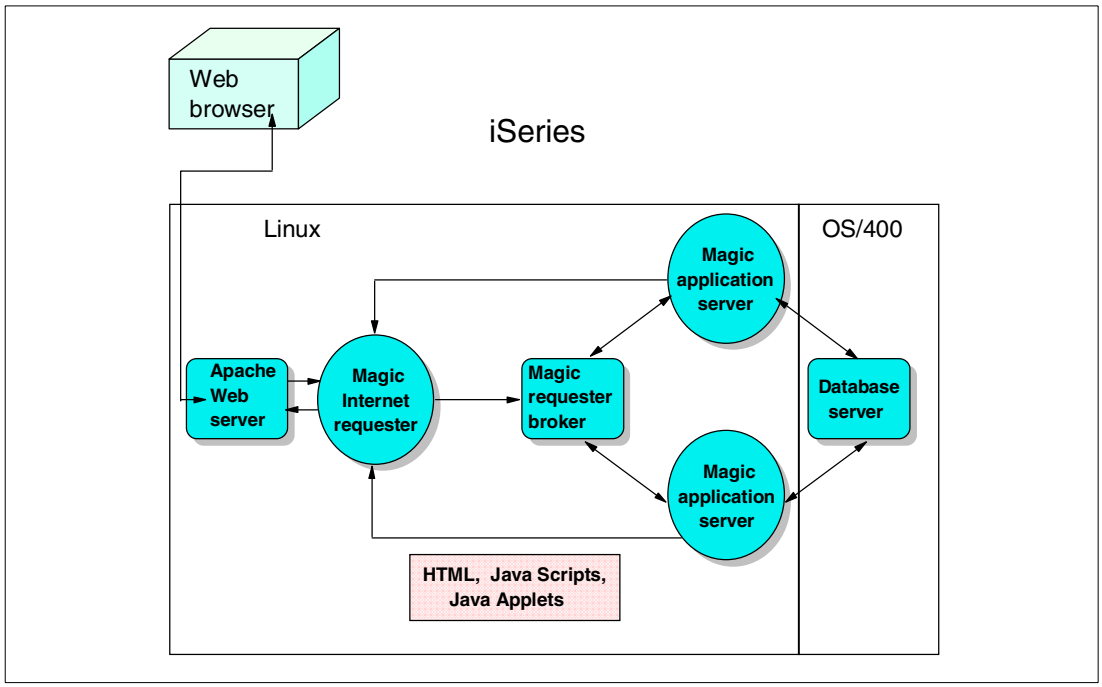


Figure A-2 The Magic eMerchant solution

The benefits of using the eMerchant application running on Linux on iSeries include:

- *Security*: By using a native Linux Apache Web server in a different partition from the e-commerce application, this enables a secure environment.
- *Fast development*: By using the Magic eDevelopment Platform, this enables faster development of a full solution.
- *Real interoperability*: Application server and database access are running on OS/400 partitions and the Internet requestor and Web server are running on Linux partitions.
- *Expansion*: It has the ability to offer the Magic application server running natively on Linux and to offer Magic's CRM and eBusiness applications on this platform.

Linux cluster for availability

The core of this scenario is driven by the Independent Auxiliary Storage Pools (IASPs) that can offer protection for planned and unplanned iSeries outages. The point is that it is now possible to mount a physical hard disk to a specified directory in the iSeries IFS, without making it a complete part of the system. To the programs using the files in that directory, this is all done transparently. The phrase "not making it part of the system" means that it is now possible to mount that same physical disk to another iSeries server.

Previously this was never possible because of the single-level storage where iSeries objects are located. You could never tell which hard disk an object resided. Even if you could, there was no way to remove the harddisk from that system and plug it into another iSeries server. This is now possible with IASP.

Now, because the virtual disks (see 2.4.5, “Create Network Server Storage Space (CRTNWSSTG)” on page 40, for information about virtual disks) that a Linux system uses are actually network server storage spaces (NWSSTGs), you can place them on such an IASP. This is because these NWSSTGs reside in subdirectories of /qfpnwsspc in the IFS. Therefore, you can place these storage spaces on such an IASP. From a Linux point of view, this is done transparently. Only the IASP needs to be mounted whenever the Linux system is varied on.

Figure A-3 shows an example where an iSeries server has a partition running Linux. iSeries A is connected to iSeries B and the I/O tower via high speed link (HSL) OptiConnect.

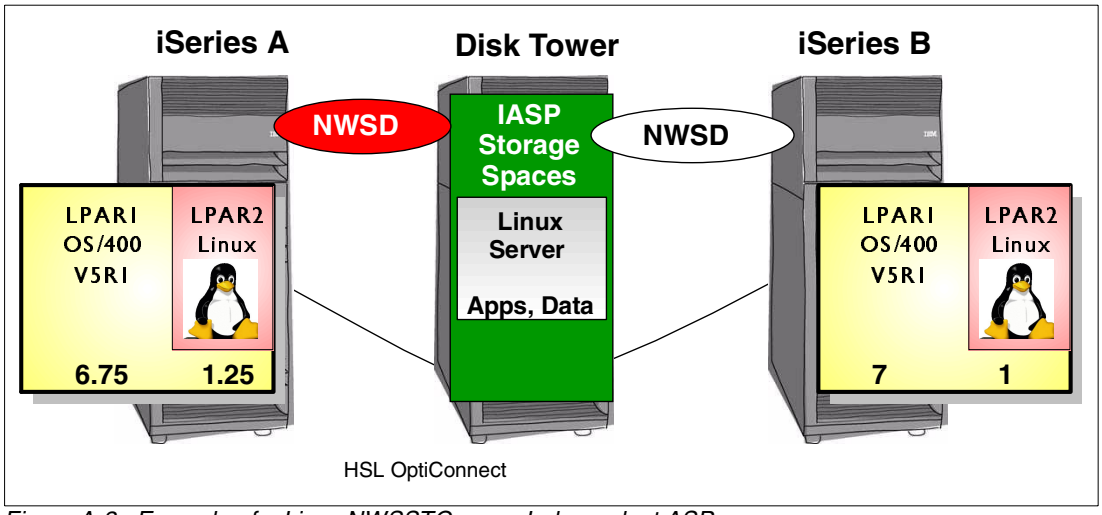


Figure A-3 Example of a Linux NWSSTG on an Independent ASP

Server A is hosting a Linux partition and the associated network server storage space (NWSSTG). The colored circle (on the left) marks which network server description NWSD is active.

At some point when there is a planned or unplanned outage on iSeries A, the iSeries HSL OptiConnect facilities detect the outage and the I/O Tower is switched to iSeries B. At that time, an administrator can move the processor and memory resources to the Linux partition and IPL the Linux server, so that the Linux server is now online on iSeries B.

This construction can also be used in one machine using LPAR. You just set up two OS/400 partitions, both with a NWSD. If the currently hosting partition fails or simply needs maintenance, the other partition can now host the Linux system after the IASP has been switched.

B

ODBC

This appendix provides basic information about Open Database Connectivity (ODBC) and references to other Redbooks and World Wide Web resources. The topics that are covered include:

- What ODBC is
- Linux implementation on the iSeries

What is ODBC

Open Database Connectivity (ODBC) was part of Microsoft Windows Open Services Architecture (WOSA). The first release came out in 1993 and has been adopted as a standard to connect to different databases. The current defined version is 3.5.1. ODBC is an application programming interface (API) that allows the programmer to concentrate on the task of getting data.

Several companies make ODBC driver for Windows-based PCs and for UNIX machines. Each ODBC drive is unique to the connected database. The iSeries Access for Windows licensed program provides an ODBC driver so Windows applications can access DB2 database data on the iSeries. The current iSeries driver is compliant with the 3.5.1 standard. You can find additional information on the standard at:
http://www.microsoft.com/data/odbc/

You can find more information on the iSeries Access for Windows driver in the iSeries Information Center (search for **ODBC**) at: http://www.ibm.com/eserver/iseries/infocenter

Linux on iSeries implementation

The iSeries Access for Windows ODBC driver was ported to Linux. This product, called *iSeries ODBC Driver for Linux* enables iSeries database access from a Linux application via the ODBC API. The driver is delivered via a Web download. To download the driver and access information on the driver, see the iSeries ODBC Driver for Linux home page at:
http://www.ibm.com/eserver/iseries/linux/odbc

The ODBC driver requires a driver manager. Linux has no ODBC driver manager built into the operating system like Windows. Instead, the UNIX ODBC driver manager is used. This is a popular open source ODBC driver manager. The user is required to obtain and install the driver manager from: http://www.unixodbc.org

Figure B-1 shows the parts of the ODBC connection.

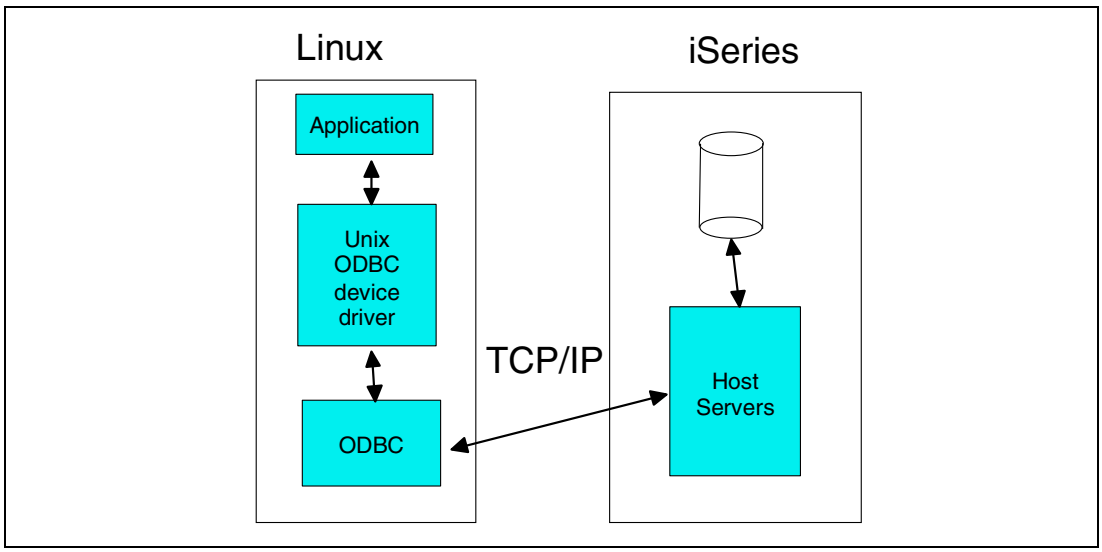


Figure B-1 IBM Linux implementation on the iSeries server

The connection of the Linux ODBC driver is through TCP/IP sockets to the OS/400 host database server. This is the same connection used by iSeries Access for Windows ODBC driver. Certain features of the Windows ODBC driver are not supported in Linux. For example, the Microsoft Transaction Server (MTS) and prompting for a user ID and password are not supported in the Linux ODBC driver. For a complete list of differences, see the iSeries ODBC Driver for Linux home page at: http://www.ibm.com/eserver/iseries/linux/odbc

There are other methods to connect to the iSeries database. For example, another way to connect is through the Java JDBC connection. However, this requires the Java Virtual Machine on the Linux distribution to use JDBC. It also require a JDBC driver such as the one provided by the IBM Toolkit for JAVA. See Chapter 7, “Tomcat Web Application Server (using Java) on Linux” on page 189, for more information.

C

Hardware planning, ordering, and configuration examples

This appendix provides the requirements and points to consider when planning and ordering Linux on iSeries. This appendix examines:

- Supported machines and processors
- Supported adapters
- LPAR planning for Linux
- Configuration examples
- Software ordering

Hardware planning

If you are planning a new system, you need to consider the following items:

- Will you use shared processors?
- Will the Linux partition use virtual I/O or native I/O or a combination of virtual I/O and native I/O? Refer to 1.4.2, “Definitions” on page 7, for a definition of virtual I/O.
- The primary partition must run OS/400 V5R1.

Supported machines and processors

The following tables show the current machines (as of V5R1) and whether they support Linux. Remember that shared processors are only available on selected models. Check the column “Supports Linux Shared processors” to ensure the model you are considering supports shared processors.

For existing n-way processor features for iSeries Model 820, 830, and 840 servers, Linux requires a minimum of one processor per Linux partition.

Table C-1 Model 2xx servers

Model	CPW	CPUs	Supports LPAR	Supports Linux	Supports Linux Shared Processor
Introduced at V5R1					
270-2431	465	1	Yes	Yes	Yes
270-2432	1070	1	Yes	Yes	Yes
270-2434	2350	2	Yes	Yes	Yes
270-2452 (DSD)	100 *	1	Yes	Yes	Yes
270-2454(DSD)	240 *	2	Yes	Yes	Yes
Note: * Dedicated Server for Domino					
Introduced at V4R5					
270-2248	150	1	No	No	No
270-2250	370	1	No	No	No
270-2252	950	1	No	No	No
270-2253	2000	2	No	No	No
270-2422 (DSD)	50 *	1	No	No	No
270-2423 (DSD)	100 *	1	No	No	No
270-2424 (DSD)	200 *	2	No	No	No
Note: * Dedicated Server for Domino					

Table C-2 Model 820 servers

Model	CPW	CPUs	Supports LPAR	Supports Linux	Supports Linux Shared Processor
Introduced at V5R1					
820-0150	1100	1	Yes	Yes	Yes
820-0151	2350	2	Yes	Yes	Yes
820-0152	3700	4	Yes	Yes	Yes
820-2435	600	1	Yes	Yes	Yes
820-2436	1100	1	Yes	Yes	Yes
820-2437	2350	2	Yes	Yes	Yes
820-2438	3700	4	Yes	Yes	Yes
820-2456 (DSD)	120 *	1	Yes	Yes	Yes
820-2457 (DSD)	240 *	2	Yes	Yes	Yes
820-2458 (DSD)	380 *	4	Yes	Yes	Yes
Introduced at V4R5					
820-2395	370	1	Yes	No	No
820-2396	950	1	Yes	No	No
820-2397	2000	2	Yes	Yes	No
820-2398	3200	4	Yes	Yes	No
820-2425 (DSD)	120 *	1	Yes	No	No
820-2426 (DSD)	240 *	2	Yes	Yes	No
820-2427 (DSD)	380 *	4	Yes	Yes	No
Note: * Dedicated Server for Domino					

Table C-3 Model 830 servers

Model	CPW	CPUs	Supports LPAR	Supports Linux	Supports Linux Shared Processor
Introduced at V4R5					
830-2400	1850	2	Yes	Yes	No
830-2402	4200	4	Yes	Yes	No
830-2403	7350	8	Yes	Yes	No
830-2351	3700	1/8	No	No	No

Table C-4 Model 840 servers

Model	CPW	CPUs	Supports LPAR	Supports Linux	Supports Linux Shared Processor
Introduced at V5R1 (includes processors on demand)					
840-2461	20200	24	Yes	Yes	Yes
840-2352	9000	8-12	Yes	Yes	Yes
840-2353	12000	8-12	Yes	Yes	Yes
840-2354	20200	18-4	Yes	Yes	Yes
Introduced at V4R5 (includes processors on demand)					
840-2418	10000	12	Yes	Yes	No
840-2420	16500	24	Yes	Yes	No
840-2416	10000	8-12	Yes	Yes	No
840-2417	10000	12-18	Yes	Yes	No
840-2419	13200	18-24	Yes	Yes	No

Once the machine model and processor are selected, you must consider the next item on the Hardware Planning list:

Will the Linux partition use virtual I/O or native I/O or a combination of virtual I/O and native I/O?

Virtual I/O and native I/O are discussed briefly. For a more information, refer to 1.4.6, “Native I/O: IOAs directly attached to the Linux partition” on page 12.

Virtual I/O

Virtual I/O functions include virtual console, virtual disk, virtual CD, and virtual tape. These virtual services are provided by communication between device drivers on the Linux partition and the hosting OS/400 partition.

Native I/O

Native IOAs, including network, DASD, tape, and optical devices, are assigned “natively” to the Linux partition. These are standalone IOAs; they are not under the control of an IOP.

The supported I/O features are shown in Table C-5.

Table C-5 Linux supported I/O features

Feature Code	Adapter	Description
0601	2743	1 Gbps Ethernet
0607	4838	10/100 Mbps Ethernet
0602	2760	1 Gbps Ethernet UTP
0603	2744	16/4/100 Mbps Token Ring
0604	2763	Ultra-2 SCSI disk and tape controller - 10MB Cache

Feature Code	Adapter	Description
0605	2748	Ultra-2 SCSI disk and tape controller - 26MB Cache
0606	2778	Ultra-2 SCSI disk and tape controller - ~104MB Cache (Note2)
Note 1: The disk controllers do not support hardware RAID or compression under Linux Native attach. **Note 2**: Adapter 2778 has approximately a 104 MB cache, depending on the compression ratio of the fast write cache.		

Now that you have examined the models, machines, and adapters that support Linux, it is time to plan and configure your logical partitions in more detail.

LPAR planning

Planning and configuring a machine for LPAR requires that you have an understanding of the following concepts:

- Bus-level and IOP-level partitioning
- Dedicated and switchable IOPs and devices
- Dedicated processors and shared processors
- Dynamic movement of resources
- Hardware configuration

If you require additional information on LPAR, there are numerous resources available to you:

- The iSeries LPAR Web site: http://www.iseries.ibm.com/lpar
- The redbook *Slicing the AS/400 with Logical Partitioning: A How to Guide*, SG24-5439
- The iSeries Information Center: http://publib.boulder.ibm.com/pubs/html/as400/infocenter.htm

LPAR planning process

LPAR planning is described in detail at the above Web sites. The following section includes an outline of these steps taken from the iSeries LPAR Web site at: http://www.iseries.ibm.com/lpar

Note: This is the process as at V4R5. This process will change slightly at V5R1, but the basic steps are similar.

LPAR planning for V4R5

To plan for LPAR with V4R5, you must follow these steps:

1. Learn about logical partitioning.
2. Have your IBM Marketing Representative or Business Partner use normal support channels for assistance.
3. Complete the system design phase. This includes completing the Configuration Planning Worksheet.
4. Conduct a Solution Assurance Review with your IBM Marketing Representative or Business Partner.
5. Order the necessary hardware and software based on the validated worksheets.

6. Install V4R5M0 or V4R4M0. We recommended that you install V4R5M0 or V4R4M0. You should also have it operating for one complete business cycle prior to setting up LPAR to allow it to stabilize in the customer environment.
7. Install new hardware based on the validated work sheets. This may require a CE to move existing hardware or install new hardware. If the CE needs to move existing hardware to new locations, a relocation or rearrangement contract is required.
8. Set up logical partitions.

LPAR planning at V5R1

The Configuration Planning Worksheet that was a requirement at V4R5 has been replaced with a downloadable tool called *LPAR Validation*. This product is for 270 and 8xx systems. It does not include the facilities for the migration tower.

The tool acts as the validation mechanism, so it is no longer necessary to complete the "Logical Partition Hardware Planning Worksheet".

LPAR Validation Tool

The LPAR Validation Tool provides a graphical user interface to logical partition planning on the iSeries. The tool includes support for a Linux partition, so you can add supported Linux adapters to a Linux partition. Other models and machines with SPD cards will continue to use the LPAR Planning Worksheet available at: http://www.iseries.ibm.com/lpar

It is very useful to understand the hardware requirements of the iSeries model that has been selected, before using the LPAR Validation Tool. For more information, refer to the following books from the Redbooks Web site (http://www.redbooks.ibm.com):

- *IBM @server iSeries and AS/400e Builder*, SG24-2155
- *IBM @server iSeries Handbook*, GA19-5486

A user manual is available for the LPAR Validator Web site that provides information on how to use the product. The following process gives you an idea about how to create a valid LPAR configuration:

1. Create a new file. Next, select the OS/400 level, system model, and memory. Then enter the number of partitions you want on the machine as shown in Figure C-1.

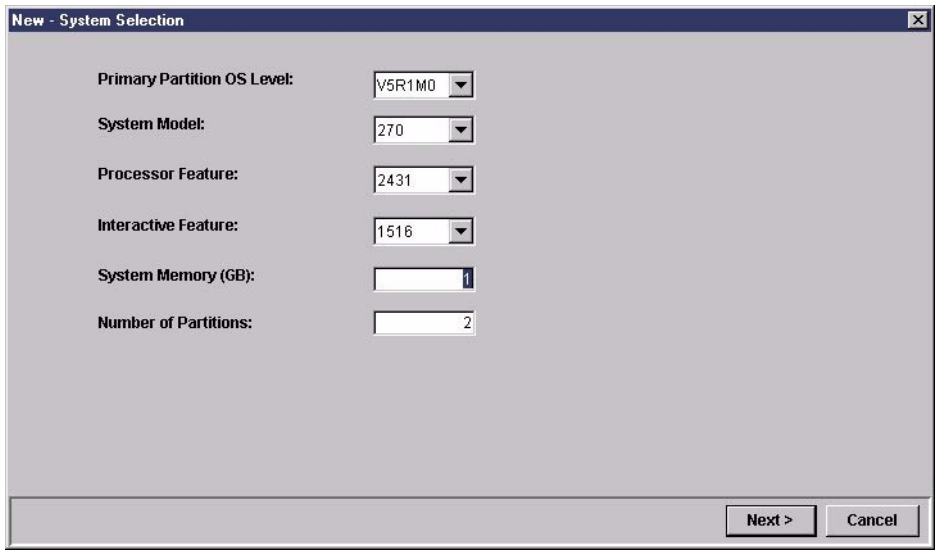


Figure C-1 LPAR Validation Tool - Selection

2. Enter the partition specifications, showing the percentage or amount of processors, memory, and interactive workload to be dedicated to each partition. Make sure that you select **Linux** under the OS Version column for a Linux partition as shown in Figure C-2.

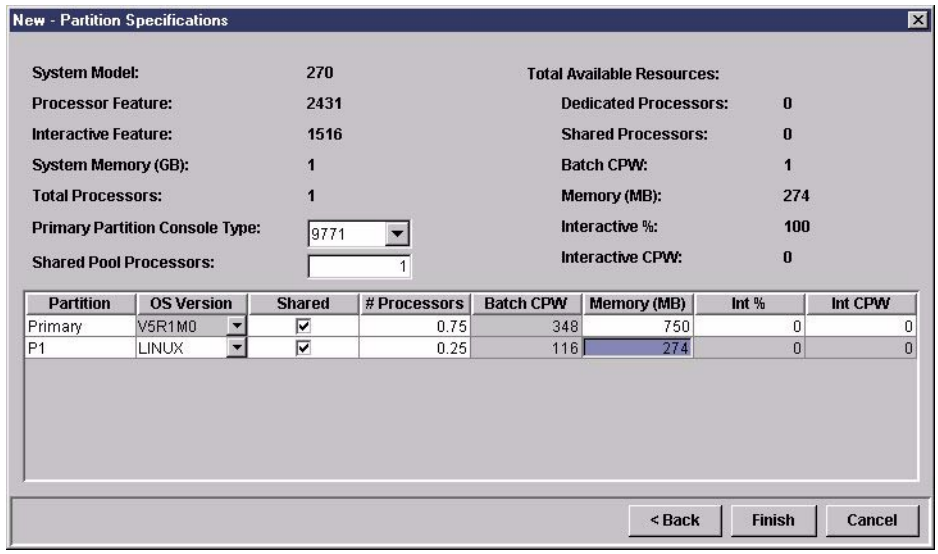


Figure C-2 LPAR Validation Tool - Partition Specifications

3. Complete the allocation of IOAs, IOPs, drives, and Linux adapters to the respective partitions. Remember if you decide to used native I/O for Linux, you need to select the relevant Linux adapters. See Figure C-3.

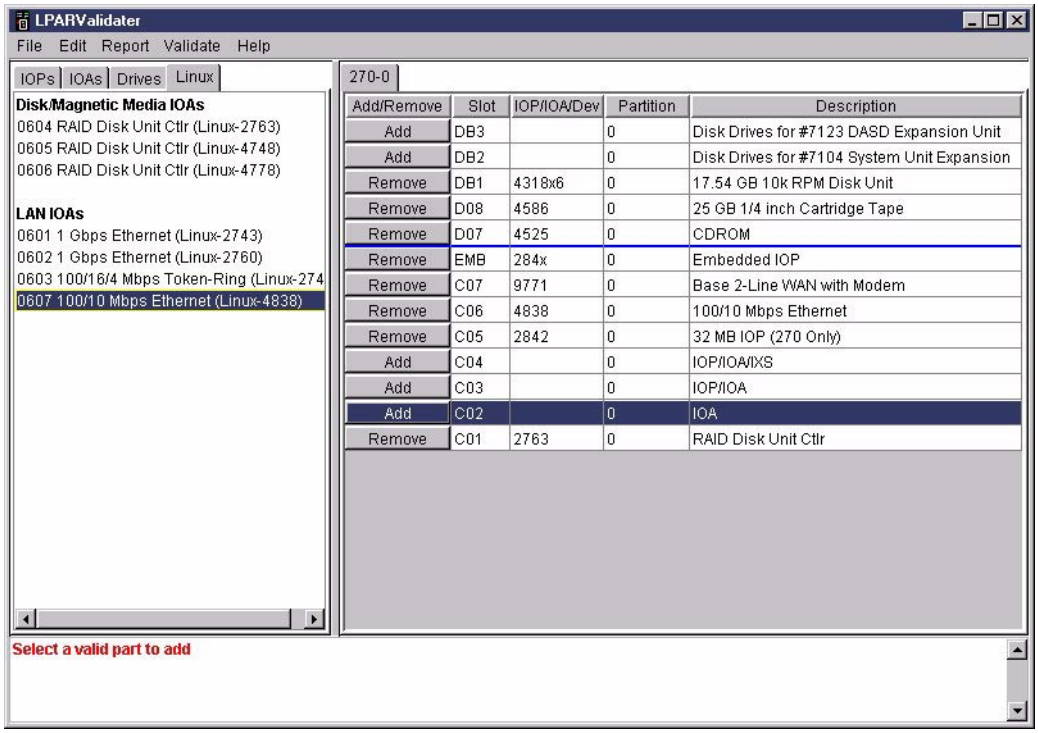


Figure C-3 LPAR Validation Tool - IOA, IOP, drive, and Linux IOA allocation

4. After your partitions are validated, print the validated report. This provides card placements for the partitions on your machine. The output of the report is very similar to the LPAR Planning Worksheet that you were previously required to complete.

Configuration example

For this configuration example, we consider the scenario of planning a new system with the intention of installing Linux on iSeries in a partition.

Planning a new system

A machine with the following requirements will be ordered:

- Two partitions
- 820-2436 (1 way)
- 1024 Mb memory
- 1 Ethernet card on an OS/400 partition
- 1 dedicated Ethernet card on the Linux partition
- Linux partition will use a virtual disk
- Linux partition will use virtual tape/CD

The display in Figure C-4 shows the system model, processor feature, interactive feature, total memory, and the number of partitions that we specified.

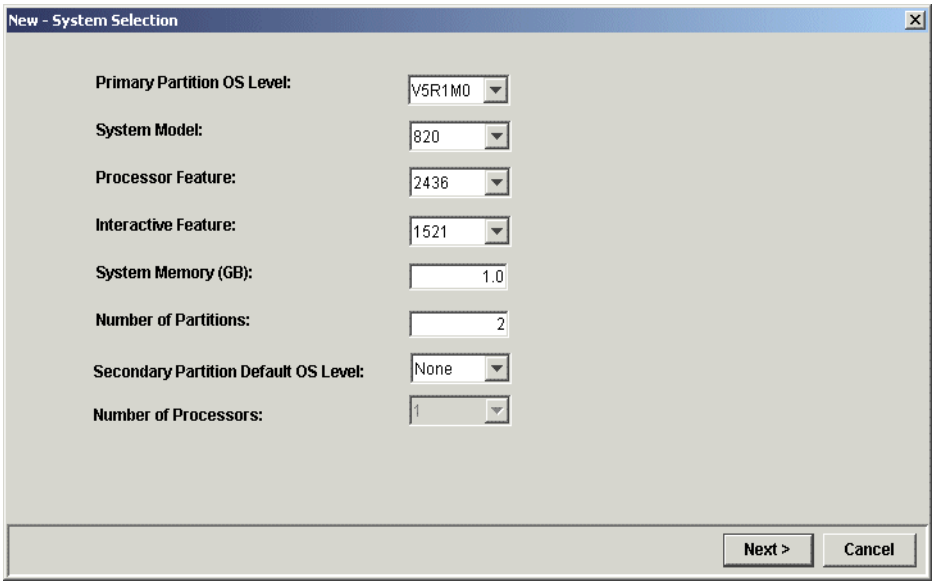


Figure C-4 LPAR validation processor selection

In Figure C-5, we specified one OS/400 partition and one Linux partition (Linux is selected from the drop-down box on the P1 partition). We made the processor a shared processor and allocated 0.75 of the processor to the OS/400 partition and 0.25 of the processor to the Linux partition.

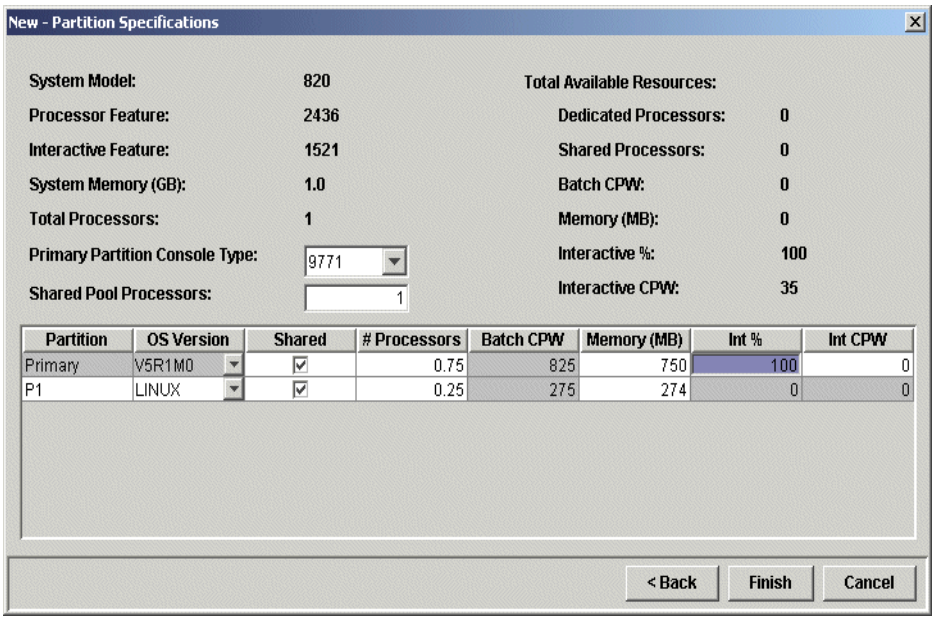


Figure C-5 LPAR Validation Tool - Sample 820

In Figure C-6, we allocated the required IOPs and IOAs to card slots. Our disk units and tape drive have been added. Note that in position C09, we have added the Linux supported Ethernet adapter 0607 (Linux-4838) for Linux partition 1.

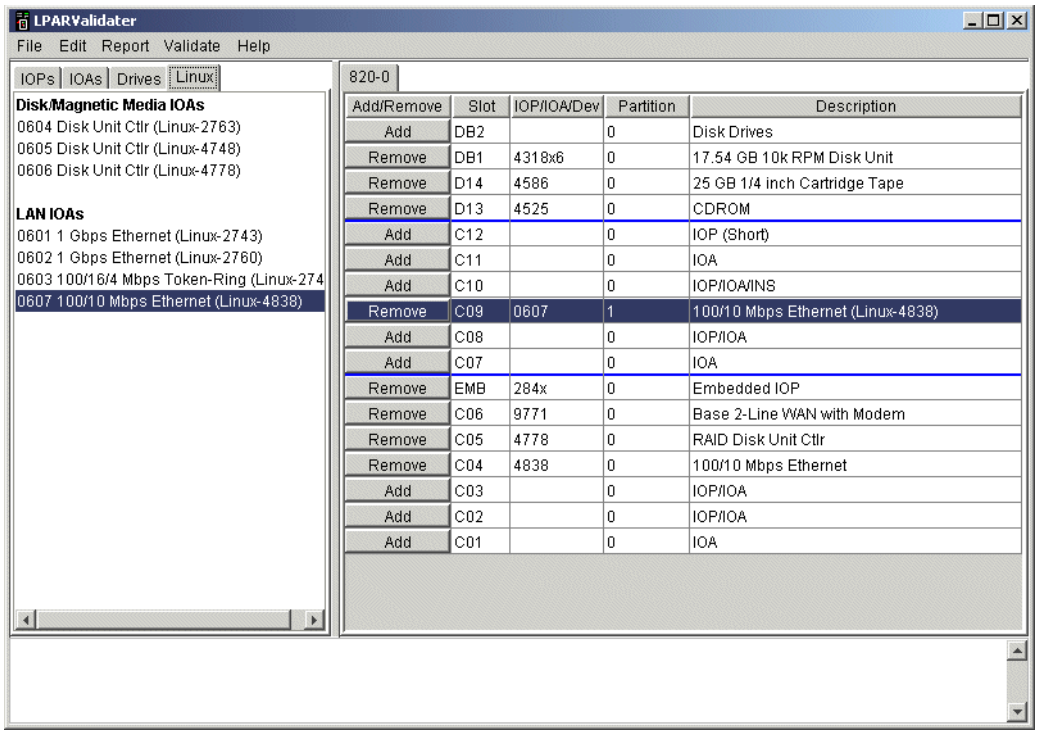


Figure C-6 LPAR Validation Tool - Sample 820 card placement

The final step is to validate the partitions. Select **Validate** from the toolbar. If your configuration passed validation, you will receive the message “All partition requirements have been validated” as shown in Figure C-7.

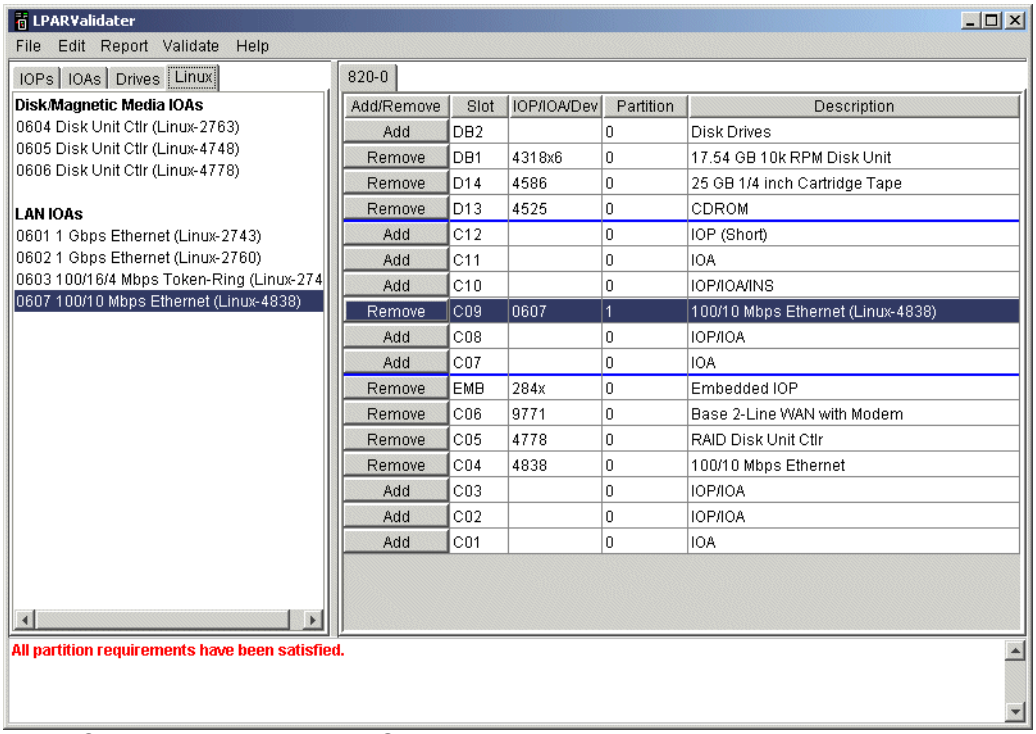


Figure C-7 LPAR Validation Tool - Sample 820 card placement

Ordering

The hardware and system software is placed through IBM either direct or a business partner. The Linux software is ordered from the distributor.

Hardware ordering

The following steps are really guidelines that you should use when ordering an iSeries machine with a Linux partition:

1. Perform proper Capacity Planning for each logical system, new or consolidated, to determine the number of partitions needed and the size of each.
2. Plan your LPAR configuration.
3. Perform LPAR validation using the LPAR Validation Tool.
4. Review requirements with Business Partner or IBM Representative.
5. Place the order with IBM.

As you may have already disclosed, this is the existing ordering process. The important step is for LPAR configuration, where you must understand the virtual I/O or native I/O requirements of your Linux partition.

Here are some points to note:

- When a machine is ordered with a Linux partition using native I/O, the adapters will not be installed on the machine. The adapters will be shipped in a separate box.
- The adapters are generally Customer Setup Up (CSU) features.
- The primary partition must be V5R1.

Figure C-8 shows a sample order of the features from the LPAR Validator.

```
9406-820 iSeries 400 Server 1
   0006 LPAR Restrict Build Process 1
   0041 Device Parity Protection-All 1
   0140 Logical Partitioning Specify 1
   0142 Linux Partition Specify 1
   0367 Operations Console PCI Cable 1
   0607 Linux Dir Attach-4838 1
   0829 #4318 Load Source Specify 1
   1413 125V 14-Ft Line Cord 1
   1521 Interactive Capacity Card 1
   2436 Model 820 Processor 1
   2924 English 1
   3006 512MB Main Store 2
   4318 17.54GB 10k rpm Disk Unit 6
   4525 CD-ROM 1
   4586 25GB 1/4-Inch Cartridge Tape 1
   4778 PCI RAID Disk Unit Ctlr 1
   4838 PCI 100/10Mbps Ethernet IOA 1
   5028 Software Version V5R1 1
   5155 Redundant Power and Cooling 1
   5157 Feature Power Supply 1
   5536 Alt IPL Spec for 25GB Tape 1
   5544 Sys Console on OP Console 1
   9002 Dual Line Cord Enabler 1
   9771 Base PCI 2-Line WAN w/Modem 1
```

Figure C-8 Order configuration

Software ordering

There should be no implications to the software ordering process when ordering a partitioned machine for Linux in iSeries. Here are some items that you need to consider:

- You need to obtain a distribution that supplies the iSeries PowerPC distribution and installation instructions. At the time this redbook was written, Red Hat, SuSE, and Turbolinux were the distributors working with iSeries.
- If you plan to use the Native attached DASD, you must have an IBM SCSI driver (IBMSIS.0) that is proprietary. It will be available on an IBM Web site and is also provided by the distributors listed above. The details are not known at this time, but they should be available on the iSeries Web site when finalized.
- The primary partition must be V5R1.
- There should not be any licensing implications to running Linux on iSeries. OS/400 will not manage license information for products running in a guest partition.

Getting help

There are several avenues available for you to find help with hardware planning or ordering:

- iSeries LPAR Web site (http://www.iseries.ibm.com/lpar)
- Business Partner
- IBM Representative
- IBM ITC located in Rochester (e-mail: rchlinux@us.ibm.com)

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications see “How to get IBM Redbooks” on page 302.

- *Exploring NFS on AS/400,* SG24-2158
- *Linux on IBM Netfinity Servers: A Collection of Papers*, SG24-5994
- *TCP/IP Tutorial and Technical Overview*, GG24-3376
- *Linux for S/390*, SG24-4987
- *Red Hat Linux Integration Guide for IBM @server xSeries and Netfinity,* SG24-5853
- *Implementing Linux in your Network using Samba*, REDP0023
- *Exploring NFS on AS/400*, SG24-2158
- *Slicing the AS/400 with Logical Partitioning: A How to Guide*: SG24-5439
- *AS/400 Internet Security Scenarios: A Practical Approach,* SG24-5954
- *All You Need to Know When Migrating from IBM Firewall for AS/400,* SG24-6152
- *IBM @server iSeries and AS/400e Builder*, SG24-2155
- *IBM @server iSeries Handbook*, GA19-5486

Other resources

These publications are also relevant as further information sources:

- *iSeries Performance Capabilities Reference V5R1*, SC41-0607
- Zwicky, Elizabeth D., et al. *Building Internet Firewalls, 2nd Edition*. O’Reilly & Associates, 2000, ISBN 1565928717
- Barrett, Daniel J. and Silverman, Richard. *SSH, The Secure Shell: The Definitive Guide*. O’Reilly & Associates, 2001, ISBN 0596000111

Referenced Web sites

These Web sites are also relevant as further information sources:

- Network Time Protocol site: http://www.ntp.org
- PowerPC Linux project site: http://linuxppc.org
- IBM Toolbox for Java and JTOpen site: http://www.ibm.com/servers/eserver/iseries/toolbox
- iSeries Information Center: http://www.ibm.com/eserver/iseries/infocenter
- iSeries logical partitioning: http://www.iseries.ibm.com/lpar
- Linux on iSeries: http://www.ibm.com/servers/eserver/iseries/linux

- The primary site for the Linux kernel source: http://www.kernel.org
- GNU project site: http://www.gnu.org
- Linux site: http://www.linux.org
- Open source site: http://www.opensource.org
- Linux documentation project: http://www.linuxdoc.org
- Linux standard base site: http://www.linuxbase.org
- SuSE site: http://www.suse.com
- Turbolinux site: http://www.turbolinux.com
- Red Hat site: http://www.redhat.com
- Apache site: http://www.apache.org
- IBM software on Linux - Ready for business: http://www-4.ibm.com/software/is/mp/linux/

How to get IBM Redbooks

Search for additional Redbooks or redpieces, view, download, or order hardcopy from the Redbooks Web Site

ibm.com/redbooks

Also download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web Site for information about all the CD-ROMs offered, updates and formats.

Index

Symbols

#!/bin/sh 139
.tar.gz file 180
/etc/group 108
/etc/pam.conf 116, 118
/etc/pam.d 117
/etc/passwd 107
/etc/printcap.local file 258
/etc/rc.d 176
/lib/modules/2.4.2 162
/var/log 111
/var/log/messages 111
/var/spool/mqueue 263
<codeset> field 118
<language> field 118
<locale> string 118
<modifier> field 118
<territory> field 118
<version> field 118
@euro modifier 118
[global] section 236

Numerics

5250 concept 166

A

A record 262
access control 112, 185
  Ext2 116
adapter status 63
all-in-one-box solution 280
anonymous FTP 224
Apache
  access control 185
  configuration 184
  default Web page 185
  modules 184
  operation 185
  server 183
  virtual named host 185
Apache Software License 190
application-level filter via proxy servers 196
apropos 274, 275
arguments 117
ASCII 144
Aside 162
asymmetrical method 213
at daemon 267, 272
atd 272
atq 272
atrm 272
automatically start XDM 176

B

backup
  commands in Linux 122
  from OS/400 side 131
  network server description 131
  network server storage space 132
  tar command 129
  using cpio command 127
  using dump 123
Backup Domain Controller (BDC) 232
bals 265
bash 139
batch mode 104
big endian 144
Blackdown JVM 191
boot parameters in Linux 38
BSD-derived FTPD 225
Bside 162
byte ordering 144
bzip2 130

C

C library revisions 153
C++ 139
cfdisk 95, 97
CGI (Common Gateway Interface) 187
CGI program 187
chgrp 112
chmod 139
chown 112
chroot environment 264
commands
  cpio 127
  find 128, 130
  pwd 278
  restore 124
  tar 129
  touch 128
comment 107
Common Gateway Interface (CGI) 187
Common UNIX Printing System (CUPS) 260
concurrent maintenance 72
console connection 48
control flag 117
copyleft 3
cpio 127
  incremental backup 128
Create Network Server Storage Space (CRTNWSSTG) 40
creating devices 90
CRM eMerchant 282
cron 267, 271
cron job 277
CRTNWSD 38

CRTNWSSTG 40
CRTUSRPRF 105
csh 139

D

daemon 104
dd 162
ddd 143
Dedicated Service Tools (DST) 21
demilitarized zone (DMZ) 196
denial of service (DoS) attacks 204
desktop environment 169
destination network address translation (DNAT) 200
devfs 90
device name changes 94
devices 90
direct attach adapter 62
direct attached I/O adapter 159
direct attached I/O device 94
direct attached IOA 8
direct attached LAN adapter 159
disk 62
disk partition
  file system 99
  formatting 99
  mounting 99
  mounting at boot time automatically 100
  unmounting 99
diskless workstation 168
DISPLAY variable 168
distributions 4, 5
DLTUSRPRF 106
DMZ (demilitarized zone) 196
DNAT (destination network address translation) 200
Domain Name Server (DNS) 262
DoS attacks 204
DST (Dedicated Service Tools) 21
dump command 123
dynamic content Web pages 186

E

EBCDIC 144
editors 88
elm 265
emacs 88, 89, 138, 275
e-mail client 262, 265
empty position 63
endianness 144
Ethernet line description 52
Eudora Pro 265
exact device name 63
Exceed 167
export 191
ext2 96
Ext2 partition-specific access control 116
external network 196

F

fdisk 95, 97
Fetchmail 265
file as backup 121
file ownership 112
file system 99
  maintenance 103
file-level backup 11
find 277
find a file 277
firewall 177, 196
  application-level filter via proxy servers 196
  ipchains 198
  ipfwadm 198
  iptables 197
  kernel 198
  Netfilter 197
  network address translation (NAT) 196
  no perimeter network in OS/400 203
  on iSeries Linux 195
  performance 199
  perimeter network in OS/400 202
  strategies for Linux on iSeries 199
framebuffer protocol 177
free software 2, 3
fsck 100, 103
FTP
  configuration file 227
  daemon on OS/400 228
  default setup in Linux 228
  protocol 224
  security 225
  servers 223

G

g++ command 139
gcc 138
gdb 138, 142
gFTP 224
glibc 190
Gnome 170
gnome-session 170
GNU C compiler (gcc) 138
GNU General Public License 2, 190
GNU GPL 2
GNU/Runtime System 2
graphical login screen 171
greeting parameter 151
grep 130
group_id 107, 108
groupadd 106, 108
groupdel 106, 108
groupmod 107, 108
groupname 108
groups 103
  managing 106
guest partition 7, 23
  connecting to a LAN 53
gvim 89

gzip 130

H

hard rules 62
hardware clock 268
Hardware Service Manager 72
headers and bodies 262
help 275
home 108
hosted partition 49, 62, 204
howtos 275
HTTP Server 184
Hypervisor 8

I

i18n 118
IBM HTTP Server 183
IBM Java Virtual Machine 190
IBM Netvista Thin Client 167
IBM WebSphere Application Server 194
ibmsis 80, 159
ibmsis.o 19, 81, 160
IFS stream files 42
imap 265
IMAP4 264, 265
incremental backup 128, 130
  using tar 130
Independent Auxiliary Storage Pools 282
info command 275
inittab 176
insmod 163
Intel 144
internal network 196
ipchains 198
ipfwadm 198
iptables 197
  example 204
iSeries
  firewall 195
  firewall security with Linux 197
  firewall strategies with Linux 199
  kernel patch 155
  Linux-specific devices 90
  time in Linux 268
  users with Linux users 108
iSeries Netserver 233
iSeries ODBC Driver for Linux 286
iSeries-specific devices 91
ISO 639 118
ISO C 118
iXplorer 220

J

Java 189
Java Virtual Machine 190
JAVA_HOME 191
JavaServer Pages 188, 190
JSP 190

K

KDE 170
kernel 144, 154, 198
  configuration 155
  patch 155
kernel-source 155
key 213
kill -9 PID 110
kill command 109
killall command 109
kmail 265
knfsd 244
ksh 139

L

LAN adapter requirement 200
LAN console 47
LANG 119
LC_ALL 119
less 88, 275
LILO (Linux Loader) 37
line print daemon (LPD) 258
line print requester (LPR) 258
Line Printer Request next generation (LPRng) 259
Linux
  backup 121
  backup command 122
  boot parameters 38
  default FTP setup 228
  devices for iSeries 90
  firewall 195, 197
  firewall security for iSeries 197
  GNU tools 2
  HTTP Server (Apache) 183
  iSeries time 268
  overview 2
  platforms 5
  print support 257
  Samba client 233
  Tomcat Web Application Server 189
  users with iSeries users 108
  using FTP servers 223
  virtual console 44
  X Windows Server 167
Linux Cluster 282
Linux kernel 2, 122, 131
Linux Loader (LILO) 37
Linux native adapters 62
Linux NetKit 225
Linux on iSeries
  firewall strategies 199
  open source applications 144
Linux partition 49
Linux SCSI driver 80
LinuxConf 89
Linux-PAM 116
listener 174
little endian 145
locale

categories 119
definition file 121
model 118
name 118
settings 119
locate 277
log file 111, 192
logical partition 7
Lotus Notes 266
LPAR Validation 294
LPAR Validation Tool 294
LPD (line print daemon) 258
LPR (line print requester) 258
LPRng 259

M

Magic eMerchant 282
mail relaying 264
mail store directory 263
mail transfer agent (MTA) 262
mail user agent (MUA) 262
Mail user agents 265
mailbox 262
maildir 263
mailing list software 266
mailman 266
majordomo 266
make 140
make Aside 162
make Bside 162
make clean 141
make config 156
make menuconfig 156
make mrproper 162
make runme 141
make xconfig 157
makefile.in file 141
makefiles 138, 140
man 89, 274
man fstab 100
man mount 100
manpages 89
mbox 263
message retrieval 264
Microsoft Windows 167
mkfs.ext2 99
mknod 90
moderated 266
module 184
module-path 117
module-type 117
mount 99, 122
MTA security issues 264
multi-user system 103
mutt 265
MX record 262, 263

N

NAT (network address translation) 56, 196
national language support 118
Native I/O 292
native IOA 8
native SCSI support 80
Netfilter 197
filtering basics 198
Netscape Messenger 265
Netvista Thin Client 167
network address translation (NAT) 54, 56, 196
network device
direct 101
virtual 101
Network File System 243
network server description (NWSD) 7
backup 131
restore 133
network server storage space (NWSSTG) 7, 40, 132
restore 134
Network Time Protocol (NTP) 269
NFS 244
nfsd 244
noarch 146
non-hosted partition 49, 62, 81, 163, 204
normal mode 224
NS record 262
ntpdate 269
ntpq 270
ntptrace 270
NWSD (network server description) 7, 8
NWSSTG (network server storage space) 7

O

occupied position 63
ODBC 286
open source 2, 3
applications 144
program 150
open system interface 4
OpenOffice 179
.tar.gz file 180
for PowerPC Linux 180
installation 180
multi-user installation 181
single user installation 180
starting 182
OpenSSH 210, 217
OS/400
backup 131
FTP daemon 228
HTTP Server 184
partition under firewall control 202, 203
restore 133
Outlook Express 265

P

packet filter function 196
Pageant 219
PAM (Pluggable Authentication Module) 116
library 107

panel 170
partition ID 96
partition remote panel key authority 21
partitioning disks 97
passive mode 224
passphrase 215
passwd 90
password 107, 108, 215
perimeter host 196
perimeter network 201, 202, 203
permission
  bits 112
  setting 113
PHP 187
pico 88, 138
PID (process ID) 108
pidof 176, 272
pine 265
pinfo 275
pipe 215
plug n' play interface 4
Pluggable Authentication Module (PAM) 107, 116
POP3 264, 265
POP3/IMAP4 to SMTP gateway 265
porting to other platforms 144
POSIX 118
Postfix 264
powered by Apache 184
PowerPC 144
PowerPC Linux 180
PReP Boot Partition 131
presentation program 179
Primary Domain Controller (PDC) 232
primary partition 7
print support 257
private key 213
problem determination 164, 188, 222, 229, 242
problem solving 192
process ID (PID) 108
ProFTPD 226
proxy ARP method 54
ps command 109
pscp 220
pseudo terminal 103
public key 213
PuTTY 45, 85, 217

Q

qmail 264
qpopper 265
Qualcomm 265

R

ramdisk 163
r-commands 210
Red Hat 260
Redbooks Web Site 302
  Contact us xv
RedHat Package Manager (RPM) 145
remote login 210
Request For Comments 276
restore
  from the OS/400 side 133
  network server description 133
  network server storage space 134
  tar command 129
  using cpio 127
restore command 123, 124
restricted state 10
restriction deletion flag 113
retrieve message 264
RFC 276
root user 104
RPM (RedHat Package Manager) 145
RPM file 148
rules 140
runme 141

S

Samba 122, 232
  share 234
Samba Web Administration Tool (SWAT) 240
save to file 121
save to tape drive 121
scheduling 267
scp 219
scp command 209, 219
scripts 139
scsi_mod.o 81
sd.o 81
secondary partition 7
secure copy (scp) 219
secure FTP (sftp) 220
secure shell (ssh) 210
SecureCRT 217
SecureFX 221
security 169
  FTP 225
  issues with server system 112
  MTA 264
  Network File System 245
  Telnet 221
Sendmail 263
server-oriented system 104
service tools user ID 21
service-name 117
Servlet 190
setgid bit 113
setting permissions 113
setuid bit 113
sftp command 209, 220
sh 139
Shaolin Secure FTP (GPL) 221
share 234
shared processors 10
shell 108
shutdown.sh 192
sisconfig 81
sisupdate utility 81

slot states 63
smbclient 233
SMTP Protocol 263
soffice 181
soft rules 62
source RPM file 148
spreadsheet 179
squid 196
sr.o 81
src.rpm 148
ssh 209, 210
ssh clients for Windows 217
SSH1 210
SSH2 210
ssh-agent 215
SST (System Service Tools) 21
st.o 81
StarOffice 179
startkde 170
startup.sh 191
static NAT 56
sticky bit 113
store an forward 263
subfolders 265
SUBSCRIBE 266
SuSE glibc 190
SWAT (Samba Web Administration Tool) 240
symbolic mode 113
system clock 268
system log file 111
system scheduling 267
System Service Tools (SST) 21
system time 267

T

tape drive as backup 121
tar 129, 180
  grep 130
TCP/IP routing 54, 59
Telnet 209, 221
  client software 221
  security issues 221
Telnet client
  in OS/400 45
  on a PC 44
Tera Term 217
test command 139
testparm 238
text editor 88
textiles 88
textinfo 275
thin client 168, 245
Tomcat 190
  default home page 193
  home page 193
  Java Virtual Machine 190
  license 190
  log file 192
  starting, stopping 191
  WebSphere Application Server 194
Tomcat Web Application Server 189
top command 109
transparent subnetting 54
troubleshooting 81
TTSSH 217
Tucows 167
TurboLinux 260
type of command 278

U

umount 99
updatedb 277
user ID 104, 245
user_id 107
user_list 108
useradd 89, 104
userdel 105
usermod 106
username 104, 107
users 103
  iSeries and Linux 108
  managing 104
UTF-8 121

V

vi 88, 89, 108, 138, 275
vigr 108
vim 89
vimtutor 89
vipw 108
virtual CD 93
virtual CD-ROM 11, 60
virtual console 11, 44, 47, 92, 103
virtual DASD 11, 93
virtual disk 93
  moving between partitions 101
virtual I/O 8, 11, 51
virtual LAN 11, 20
  configuration 52
  Ethernet line description 52
virtual named host 185
Virtual Network Computing (VNC) 177
Virtual OptiConnect 11
virtual tape 11, 62, 94
virtual tape drive 122
vmlinux 162
VNC (Virtual Network Computing) 177
VNC protocol 177
vncclient 177, 178
vncserver 177, 178
vsftpd 226

W

Web page 185
  with dynamic content 186
WebSphere Application Server 194
window manager 169
Windows 167

WinSCP 220
word processor 179
WS_FTP 224
WU-FTPD 226

X

X Display Manager 172
X Session 175
X terminal 168
X Windows 166
X Windows Server
  for Linux 167
  Microsoft Windows 167
X/Open Portability Guide 118
x86 processors 144
xclock 169
XDM 171, 172
  problems 177
XDMCP-Query 175
xf86 172
X-forwarding 217
XFree86 172
xload 169
xntpd 269
XPG4 118
XPG5 118
xstartup 178
xterm 169
X-Win32 167
XXgdb 143

Y

yank-last-argument 146
YaST 89, 176

Linux on the IBM @server iSeries Server: An Implementation Guide

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

Linux on the IBM @server iSeries Server

An Implementation Guide

Redbooks

Benefit from the advantages of running Linux on the iSeries server

Find out how to install and use Linux on iSeries

Learn which Linux applications can run on iSeries

Running Linux on the IBM @server iSeries server combines the strengths of Linux and OS/400 for an integrated solution. Linux delivers excellent open source solutions, while OS/400 is a premier integrated platform for business solutions. Linux enables a new stream of e-business applications for the iSeries platform that complements its strength as an integrated core business solution. Linux applications benefit from the iSeries platform's ability to provide resource flexibility, security, reliability, and connectivity to other applications on a single server.

This IBM Redbook begins with an overview of Linux, defines what open source means, and explains why using Linux on iSeries is beneficial. Then, it highlights how to install and use Linux on the iSeries server. It discusses the basic system administration tasks and Linux application development to help you manage your system and develop Linux applications on the iSeries server. It also introduces a wide range of services, such as Firewall, Apache, Samba, and e-mail, and explains the capabilities of each.

This redbook is intended to help beginner and intermediate Linux users, with an OS/400 background, to implement Linux on the iSeries server.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-6232-00 ISBN 0738423750