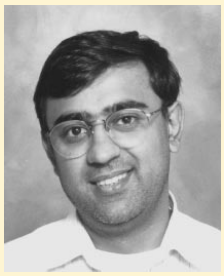
AIX Questions

Compiled by Ismat Dhanjibhai

The AIX Solution Provider Technical Support group in Austin, Texas, supports software vendors who are porting applications to AIX. This article is a compilation of questions that are frequently asked by vendors. The name of the responding Technical Support Group staff member appears after each response.



Ismat Dhanjibhai

I am having a problem making a `mksysb` tape. The errors are as follows:

**Error:*.add not found bosbot:boot image is 3124 512 byte blocks, mkinstal tape failed. 0512-0016 mksysb: attempt to create a bootable tape failed;
mkinstal tape /dev/rmt0.1 failed with return code 5.**

You are using your own `restore` command instead of the system `restore` command. Changing PATH to access the system `restore` (`/etc/restore`) before your own will solve the problem.

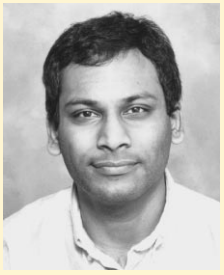
—Sue Lowe

Sue Lowe

Is it possible to increase the 1 MB circular-error log file?

First, run the command `/usr/lib/errdemon -l` to check the current maximum `errlog` size in bytes. Then run the command with the `-s` flag and specify the new size.

—Sue Lowe



Sai P. Ramanath

What is the meaning of the "< >" symbols in the `/bin/oslevel` output (such as < >3250)?

The default output compares the system software maintenance level to the currently applied maintenance level. For example, the shorthand for the 3240 maintenance level is as follows:

Symbol	Description
=3240	Equal to
<3240	Below
>3240	Above
<>3240	At mixed levels

The following options can be specified to determine which products and subsystems differ from the currently applied maintenance level.

`-l`: List products at levels earlier than maintenance level

`-g`: List products at levels later than maintenance level

`-e`: List products at current maintenance level

—Sai P. Ramanath

Is there a penalty for compiling an executable with -bmaxdata to allow 2 GB of data? By "penalty," I mean does the binary start much slower than if it is compiled with -bmaxdata:0x50000000? Or is there no penalty to compile with -bmaxdata:0x80000000? If there is no penalty, why doesn't the default compiler allow every executable to be a maximum of 2 GB in data size? Why do I have to change the switch?

When the option `-bmaxdata:` is used, all static data (whether initialized or not) is grouped together. It is then placed into shared memory segments that begin at address 0x3000000. If the data extends past the maximum size of a memory segment (256 MB), then a second, contiguous segment is constructed beginning at address 0x4000000, and

so on. Up to eight segments (2 GB of virtual memory) can be constructed in this manner. The shared memory segments are private—not accessible to other processes.

Real memory/paging space is assigned when the memory is actually used; if it is never accessed, it will not be assigned. Keep in mind that initialized data is "accessed" when the executable is started.

In general, there is little overhead when using the large data model. More virtual memory objects will be used, but the system should have plenty. The large data model is not the default primarily because the upper memory segments are needed for other purposes, such as shared-memory interprocess communication.

Any application that contains many forks will experience additional overhead because the most expensive part of the fork operation is duplicating the process address space—the `vmm_forkcopy` operation must be done for every memory segment that is allocated.

Note: The `maxdata` option only tells the linker where (potentially) to place the data. The actual data requirements are still the critical issue when running the program.

—Sai P. Ramanath

I am using the C++ compiler with the +- option. A note on page 118 of the C++ compiler manual states that if I use this option, `dbx` will not recognize files ending with ".c".

The `dbx` has been modified with Basic Operating System (BOS) 3.2.5 to work with all extensions except `-o`, `-a`, or `s`. It now allows for debugging programs without a `.c` extension that were previously compiled using the `+-` option.

—Jeff Simon

Is the RISC System/6000 internal CD-ROM compliant with Small Computer Systems Interface-2 (SCSI-2)?

CD-ROM type A is SCSI-1 and type B is SCSI-2. If your RISC System/6000 has a SCSI-1 adapter, both type A and type B drives operate as SCSI-1 devices. If your RISC System/6000 has a SCSI-2 adapter, the type A drive operates as a SCSI-1 device and the type B drive operates as a SCSI-2.

—Priyamvada

How can I determine the size of the dump device?

The `sysdumpdev -e` command returns the size of the dump in bytes. The dump device driver calculates the size of the dump as follows:

- Size of the Virtual Memory Manager (VMM) dump data (0.025 * actual memory)
- Size of memory allocated to VMM segment `(vms_rusage(vmmsrval) * PAGESIZE`
- Size of memory allocated to kernel extension segment `(vms_rusage(kexsrval) * PAGESIZE)`
- Size of memory allocated to kernel segment `(vms_rusage(kernsrval) * PAGESIZE)`

The total of all these sizes is returned to `sysdumpdev`. The estimated size of the system dump is that total plus 25%.

—Priyamvada

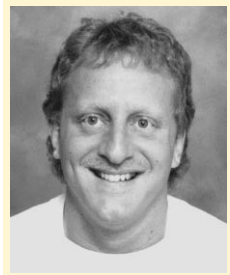
I am using `/etc/security/audit/objects` to perform an audit on some object files. I can't generate `S_PASSWD_WRITE`. I use emacs for day-to-day editing, and this only logs an `S_PASSWD_READ` event.

The `vi` editor available from IBM is integrated with the auditing subsystem. Non-IBM products such as emacs may not fully integrate with the auditing subsystem. Since the emacs you are using is not integrated, you do not get all the events. A workaround is to enable auditing of file writes via the kernel events.

—Ismat Dhanjibhai

The `pcnfsd` is in the `/etc/inetd.conf` file (and it is commented out). When my machine boots (it is running AIX 3.2.5) or when I refresh the `inetd`, the `pcnfsd` does not start. Do I need to do something different?

Figure 2 shows the correct way to verify whether `pcnfsd` (RPC program number 150001) is running. The file `/usr/sbin/rpc.pcnfsd` is part of the `bosnet.nfs.obj` program. Output from Figure 2 should look similar to Figure 3.



Jeff Simon

Priyamvada

1. vi /etc/inetd.conf and uncomment (remove the "#") the following entry:

```
pcnfsd sunrpc_udp udp  wait  root  /etc/rpc.pcnfsd pcnfsd 150001 1
```

2. > refresh -s inetd

3. > rpcinfo -p

Figure 2. Verifying whether pcnfsd is running

```
program vers proto   port
100000    2   tcp    111  portmapper
100000    2   udp    111  portmapper
100001    1   udp   1033  rstatd
100001    2   udp   1033  rstatd
100001    3   udp   1033  rstatd
100024    1   udp    849  status
100024    1   tcp    851  status
300082    1   udp    854
300082    1   tcp    856
100021    1   tcp    683  nlockmgr
100021    1   udp    685  nlockmgr
100021    3   tcp    688  nlockmgr
100021    3   udp    690  nlockmgr
100020    1   udp    693  llockmgr
100020    1   tcp    695  llockmgr
100021    2   tcp    698  nlockmgr
150001    1   udp   1128
```

Figure 3. Output from the rpcinfo -p command

The 150001 entry shows that the `rpc.pcnfsd` is running and listening. To verify this, enter the following command:

```
rpcinfo -u <local hostname> 150001
```

This polls the `rpc.nfsd` process to verify that it is running. The following output is produced:

```
program 150001 version 1
        ready and waiting
program 150001 version 2
        ready and waiting
```

—Rick Malone

When a Network Computing System (NCS) server is busy with an NCS client and a client is waiting, does the client send the data more than once? How frequently does a client send the data to the server? Is it possible when the client resends requests many times that the data size exceeds the number of `mbufs`?

Yes, when a client is waiting, it can send its data more than once. The mechanism is as follows:

When a client request times out, the client NCS runtime attempts to ping the server with a NULL RPC. If the ping succeeds, the client resends the data.

The ping's success or failure is determined differently for two separate cases: when normal (long) timeouts are set and when short timeouts are set. In the former case, the client sends Remote Procedure Calls (RPCs) to ping the server at one-second intervals up to 30 times. In the latter case, only one ping (with a one-second timeout) is sent. A client that times out and resends requests will not cause a server machine to run out of `mbufs` since the vast majority of the RPCs sent are "ping" RPCs, with only a header and no body (data).

—Venkatesh M. S. Iyer

Ismat Dhanjibhai, IBM Corporation, RISC System/6000 Division, 11400 Burnet Road, Austin, TX 78758. Internet: ismat@austin.ibm.com. Mr. Dhanjibhai, a member of the AIX Solution Provider Support Center, currently provides technical support to software vendors. Mr. Dhanjibhai has a degree equivalent to Computer Studies from the Polytechnic of Wales in the United Kingdom.