TTY Subsystem in SMP

By Eddie Ho, Lance Russell, James Partridge, and Derwin Gavin

The AIX 4.1 multi-user terminal I/O subsystem has been reengineered to operate in a Symmetric Multiprocessor (SMP) environment. This includes the overall subsystem migration from BSD to System V Release 4 using the Streams framework. The new Streams-based TTY subsystem conforms to the new interface and standard and offers additional flexibility to meet today's multi-user system requirements.

AIX is a multi-user operating system that can be accessed by hundreds of users. The TTY subsystem is the part of kernel responsible for the data path between the running application and all I/O devices. The originating devices for AIX that use this subsystem can be grouped into three classes:

- **Networked/virtual devices** including `aixterm`, `xterm`, or applications such as `telnet/rlogin`
- **Physical devices** attached directly or through a multiport asynchronous adapter (Physical devices include ASCII terminals, serial printers, plotters, modems, and data collecting devices—any device with an RS-232 interface.)
- **System console** with low-function terminal (lft) emulation, primarily used as a terminal console for booting the system and for initial administration use

AIX 4.1 is SMP-enabled, providing serialized access to common data structures and efficiently using Streams and MP resources. As a result, protocol modules or device drivers have scalable performance on any processor. Other highlights of AIX 4.1 TTY subsystem include the following:

- Supports a single open-system terminal discipline (POSIX™) instead of three different flavors (BSD, AT&T, and POSIX), thereby eliminating confusion for administrators selecting a discipline
- Supports ISA bus-based native ports in the widely used RS/6000 Model 40P
- Supports the three native ports in the SMP hardware with console mirroring
- Supports native serial ports, the 8-port adapter, the 16-port adapter, and the 128-port asynchronous controller that is currently available in AIX 3.2
- Supports both BSD and AT&T pseudo-TTYs and their naming conventions, enabling applications to fully use the AT&T TTY functionality
- Supports the following communication protocols:
  - Terminals, printers, and plotters
  - Modems
  - Communication applications such as Serial Line Internet Protocol (SLIP), Asynchronous Terminal Emulation (ATE), and Communication Utility (CU)

Most existing application interfaces are compatible with AIX 4.1 except those using the IBM PC-RT® or IBM PC-specific `ioctls`. Equivalent functionality is provided using the standard interfaces, which allows your AIX 3.2 terminal I/O application to be migrated to AIX 4.1 without recompiling.

Figure 1 summarizes the AIX multi-user environment.

Streams Overview

Streams is a flexible set of tools for developing UNIX system communication services. It defines a generic message-driven queuing interface and provides a framework and tools for implementing communication services, such as a network pro-

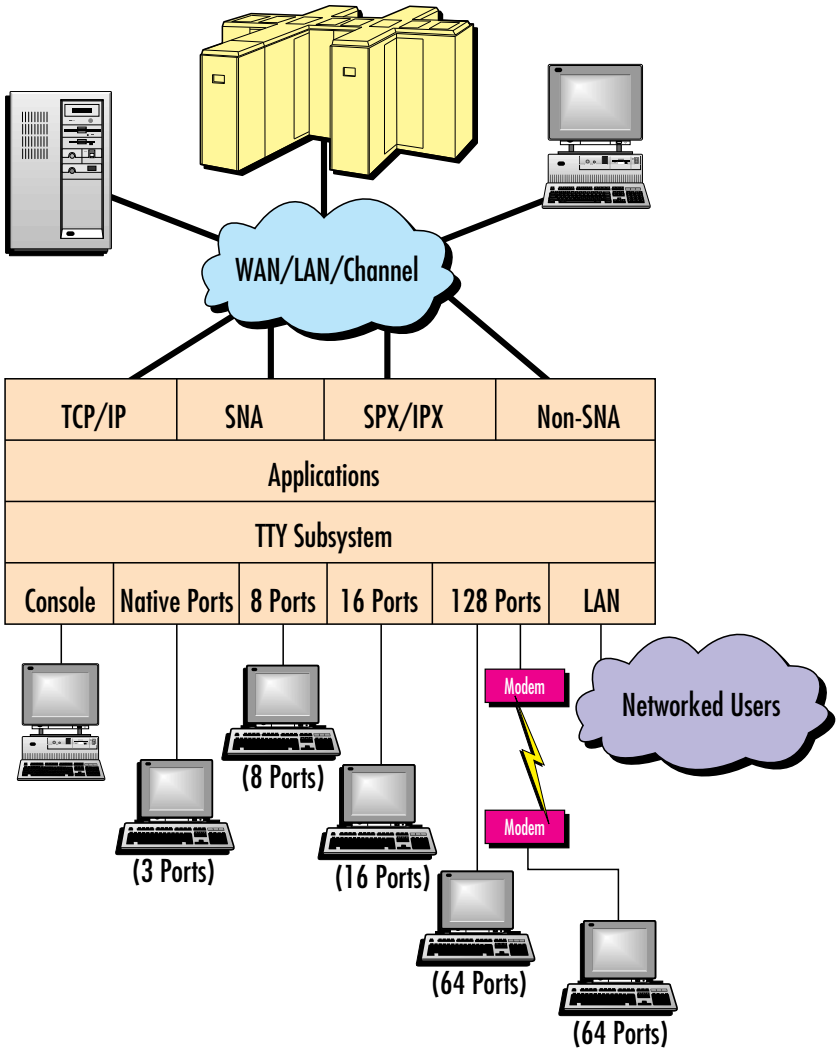

Figure 1. AIX multi-user environment

tocol stack. Streams does not impose any specific network architecture—it is simply a framework. The framework includes the following components:

- **Stream Head**: Part of the stream that interfaces to the user process
- **Stream Modules**: Kernel-level routines that implement the behavior of the supported protocol suite
- **Stream Device Driver**: The driver that handles the data between an external I/O or pseudo device and the AIX kernel

Figure 2 illustrates a simple transport model using Streams.

Benefits of Streams

Streams is a standard interface for developing modular, portable, networking layers that reuse code from different communications protocols. It facilitates the rapid growth of communications services. The Streams framework on AIX 4.1 conforms to OSF/1 Version 1.2. There are several benefits to Streams-based communication:

- The user application can customize the protocol stack in real-time by making the code path more efficient and direct for the application. For example, during printer configuration sessions, the printer discipline module is pushed rather than the terminal discipline module.
- Various protocol layers and device drivers can be shared and mixed without predetermined

restrictions based on the application requirements. For example, all terminal devices share the terminal discipline module, including the lft emulation.

- Runtime protocols can be substituted by the user application. For example, in the remote TCP/IP (SLIP) environment, the discipline module replaces the SLIP module in order to meet the protocol requirement.

TTY Subsystem Infrastructure

The TTY subsystem has many modules that conform to the Streams architecture. Applications define the stream stack based on a set of port requirements during the special file-open phase. Conversely, the stack is destroyed when the special file is closed, such as when the device is disconnected. The stack structure consists of the following:

- **Stream head**: Required for all stream stacks, it processes user requests; it is common to all Streams devices regardless of the nature of the implementation
- **Line discipline module**: Implements the session functions and supports the following types of protocols:
  - Terminal discipline (ldterm)—used by all terminal sessions, including lft emulation
  - Serial printer discipline (sptr)—used by all printer sessions except the parallel printer support
  - TCP/IP discipline for serial lines (SLIP)
- **Streams end**: Represents hardware-specific device drivers, including both Micro Channel® and ISA boards, and pseudo device drivers such as pseudo-TTY

The stream stack can contain other modules:

- **Ioctl processing** provides transparent `ioctl` processing before passing the message downstream.
- **Console mirroring** echos data from one serial port to another serial port (only applicable on SMP hardware with three serial ports).
- **Data mapping** allows I/O data to be remapped to a different value when a mismatch between the terminal hardware and the application's code page occurs. Typical environments include non-ISO-8859-1 ASCII terminals that are incompatible with the application

Streams Model

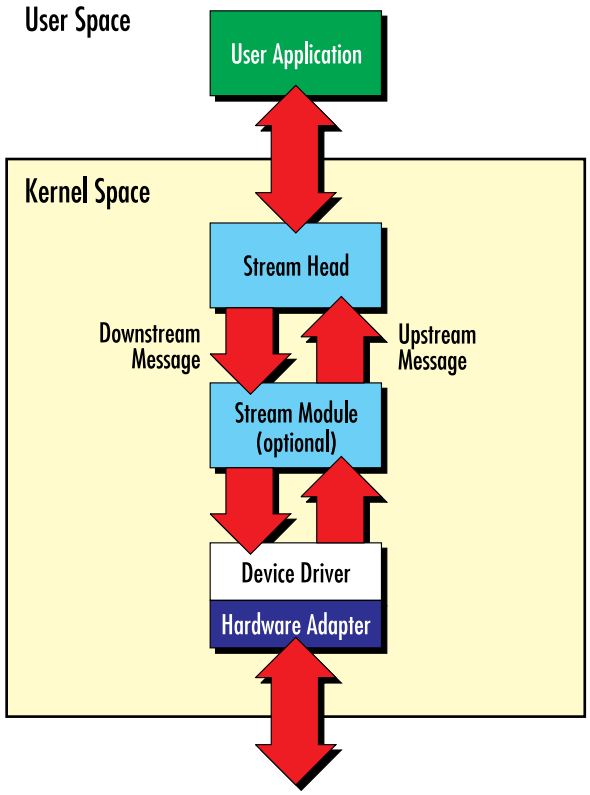

Figure 2. Streams model

or IBM Code Page 850 applications that are being accessed from an ISO 8859-1 terminal.

- **Code Page 932 converter** converts input data from Code Page 932 to IBM EUC-JP and vice versa; these modules will be pushed on the stack whenever the system locale is set to IBM 932 in Japan.

Figure 3 summarizes the construction of each protocol stack during runtime.

Changes to Commands in AIX 4.1

The AIX 4.1 `stty` command supports a single comprehensive discipline (POSIX) as defined by Open Software Foundation® (OSF®). This eliminates the need for AIX Version 3 subcommands, such as `stty-berk`, `stty-att`, and `stty-bsd`, which are therefore removed. Manipulation options (such as `add`, `del`, `disp`, and `get`), that accompany each subcommand are also removed.

An equivalent interface is provided via the Streams framework. For example, instead of using `stty get` to view the protocol stack, use the `strconf` command.

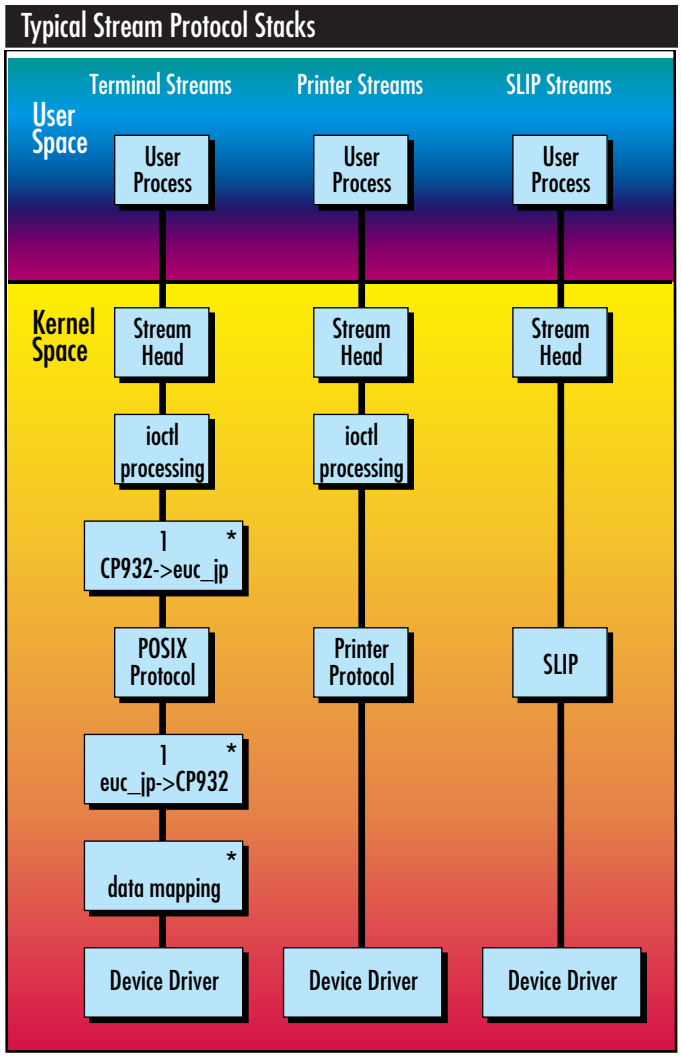

Figure 3. Typical stream protocol stacks for terminal, printer, and SLIP functions. (* *Optional modules*)

AIX 4.1 Header File

The AIX 4.1 header files are the same as in AIX Version 3:

- `ioctl.h` for BSD and AT&T programming interface
- `termios.h` for POSIX interface
- `termio.h` for AIX compatibility interface

An additional header file `termiox.h` is used for the new extensions to supplement the existing header files.

AIX 4.1 Programming Interface

The major programming interface changes are in two areas.

- Removes some PC-RT interfaces (`tcgcsmap`, `tcscsmap`, `txgbaud`, `txsbaud`) if there is an equivalent interface using the POSIX standard.
- Replaces `ioctls` that conflict with the Streams architecture (`txaddcd`, `txdelcd`, `txgetcd`, `txsetld`) with Streams `ioctls` such as `i_push`, `i_pop`, and `i_list`.

Maintaining Port State Information for a TTY

In AIX Version 3, TTY port attributes, such as line speed and character size, were persistent regardless of how the TTY driver specified those attributes. In AIX 4.1, TTY attributes specified using the `ioctl()` system call, the `tcsetattr()` routine, or the `stty` command persist only until the TTY driver processes a `close()` system call. The TTY attributes specified using the System Management Interface Tool (SMIT) or the `chdev` command set new TTY default values that are persistent until replaced by newer default values. The examples in Figure 4, which assume that `tty1` has been defined via SMIT to have a line speed of 38,400 bits per second, illustrate the behavior differences between AIX 3.2 and 4.1.

To summarize, SMIT and the `chdev` command set the default set of attributes, which persists across sessions and during reboots. The `stty` command `tcsetattr()` interface sets the runtime attributes for only as long as the device is open.

Differences in AIX 4.1 Pseudo-TTY (PTY)

Pseudo-TTY is the pseudo-driver used for network terminal communication, such as the `telnet` or `rlogin` command. There are two major differences in PTY usage for Version 4.1: TTY naming and usage conventions, and the configuration of AT&T PTYs.

Pseudo-TTY Naming and Usage Conventions

AIX Version 4.1 supports both BSD and AT&T PTY conventions. The key element of AIX 4.1 PTY is moving toward the OSF®standard and departing from the AIX Version 3 mixed BSD and AT&T naming standard. AIX 4.1 implements clone devices. The AIX Version 3 PTY implementation uses multiplexed files and allows a non-conventional mixing of BSD and AT&T naming.

AT&T Naming Convention and Calling Sequence: AT&T naming has one master—a clone driver—and *n* slaves. The PTY driver manages the connection between master and slave. When a user process opens a PTY line, it must follow this sequence:

1. Open the master (`/dev/ptc`).

Example 1: Changing Port Speed on Current Port

User Actions	Version	Results
1. Log in to `tty1`.	AIX.3.2.5	Line speed is 2400 bps.
2. Issue `stty 2400` from the terminal	AIX 4.1	Line speed defaults to the SMIT attribute—38,400 bps.
3. Log out.		
4. Log in again.		

Example 2: Changing Port Speed on Different Port

User Actions	Version	Results
1. The `tty1` is idle and there is no `getty` process running on it.	AIX 3.2.5	`tty1` is set to 2400 bps by step 2, resulting in the `cat` command displaying the data using 2400 bps.
2. Issue `stty 2400 < /dev/tty1` from User 1 at another terminal.	AIX 4.1	`tty1` is closed in step 2 by the `stty` command. `tty1` is returned to the SMIT default of 38,400 bps; thus, the `cat` command will attempt to display the data using 38,400 bps.
3. Issue `cat /etc/motd > /dev/tty1` from User 2 at another terminal.		

Figure 4. Examples of port state changes

2. Read the name of the slave associated, calling `ttyname()`for example, `/dev/pts/n (n=0,...,n-1` where `n` is a PTY-configurable parameter).
3. Open the slave.

Each time a user process calls `ttyname()`, it gets the first free slave (for example, if the first slave free is the slave number 4, it gets `/dev/pts/4`). The slave cannot be opened before a master because the user process in AT&T naming would not know the name of the slave before opening the master.

BSD Naming Convention: The master and slave have the same number in BSD naming. The connection between master and slave is determined by a naming convention. Since each master has its associated slave, a user process can first open either the master or the slave. The order is not important and there is no clone master.

AIX 4.1 PTY Naming and Usage: All AIX 4.1 applications should be using the AT&T naming convention—the OSF standard. Although AIX 4.1 supports the BSD naming convention to facilitate BSD migration, the BSD and AT&T naming conventions cannot be mixed, even though this is allowed in AIX Version 3.

AT&T PTY Configuration

AIX Version 3 system administrators do not have to control the number of AT&T PTYs from the SMIT PTY configuration panel; they are automatically generated based on demand. The maximum number of PTYs is based on system resources, such as the number of inodes. In AIX 4.1, the SMIT configuration controls the number of AT&T PTYs, with a default of 256. System administrators must allocate more PTYs if needed by the application.

Eddie Ho, IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Mr. Ho is a senior programmer in the AIX Executive Briefing Center. He has a BS in Computer Science from the University of Wisconsin and an MS in Computer Science from North Dakota State University.

Lance Russell, IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Mr. Russell works in AIX kernel development. He has a BS in Computer Science from the University of California at Berkeley and an MS in Computer Science from the University of Minnesota.

James Partridge, IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Mr. Partridge works in AIX communications development. He has a BS in English from the University of Texas in Austin.

Derwin Gavin, IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Mr. Gavin is a senior associate programmer in AIX communications development. He has a BS in Computer Science from Grambling State University in Louisiana.