NAME

pts – STREAMS pseudo-tty slave driver

DESCRIPTION

The pseudo-tty subsystem simulates a terminal connection, where the master side represents the terminal and the slave represents the user process's special device end point. In order to use the pseudo-tty subsystem, a node for the master side driver **/dev/ptmx** and N nodes for the slave driver (N is determined at installation time) must be installed. The names of the slave devices are **/dev/pts/M** where **M** has the values 0 through N-1. When the master device is opened, the corresponding slave device is automatically locked out. No user may open that slave device until its permissions are adjusted and the device unlocked by calling functions **grantpt**(3C) and **unlockpt**(3C). The user can then invoke the open system call with the name that is returned by the **ptsname**(3C) function. See the example below.

Only one open is allowed on a master device. Multiple opens are allowed on the slave device. After both the master and slave have been opened, the user has two file descriptors which are end points of a full duplex connection composed of two streams automatically connected at the master and slave drivers. The user may then push modules onto either side of the stream pair. The user needs to push the **ptem**(7M) and **ldterm**(7M) modules onto the slave side of the pseudo-terminal subsystem to get terminal semantics.

The master and slave drivers pass all messages to their adjacent queues. Only the **M_FLUSH** needs some processing. Because the read queue of one side is connected to the write queue of the other, the **FLUSHR** flag is changed to the **FLUSHW** flag and vice versa. When the master device is closed an **M_HANGUP** message is sent to the slave device which will render the device unusable. The process on the slave side gets the errno **ENXIO** when attempting to write on that stream but it will be able to read any data remaining on the stream head read queue. When all the data has been read, read returns 0 indicating that the stream can no longer be used. On the last close of the slave device, a 0-length message is sent to the master device. When the application on the master side issues a **read( )** or **getmsg( )** and 0 is returned, the user of the master device decides whether to issue a **close( )** that dismantles the pseudo-terminal subsystem. If the master device is not closed, the pseudo-tty subsystem will be available to another user to open the slave device. Since 0-length messages are used to indicate that the process on the slave side has closed and should be interpreted that way by the process on the master side, applications on the slave side should not write 0-length messages. If that occurs, the write returns 0, and the 0-length message is discarded by the **ptem** module.

The standard STREAMS system calls can access the pseudo-tty devices. The slave devices support the **O_NDELAY** and **O_NONBLOCK** flags.

EXAMPLES

```
int fdm fds;
char *slavename;
extern char *ptsname();

fdm = open("/dev/ptmx", O_RDWR);  /* open master */
grantpt(fdm);                     /* change permission ofslave */
unlockpt(fdm);                    /* unlock slave */
```

```
slavename = ptsname(fdm);           /* get name of slave */
fds = open(slavename, O_RDWR);      /* open slave */
ioctl(fds, I_PUSH, "ptem");         /* push ptem */
ioctl(fds, I_PUSH, "ldterm");       /* push ldterm */
```

FILES

/dev/ptmx	master clone device
/dev/pts/M	slave devices (M = 0 -> N-1)

SEE ALSO

grantpt(3C), **ptsname**(3C), **unlockpt**(3C), **ldterm**(7M), **ptm**(7D), **ptem**(7M)

STREAMS Programming Guide