Reimplementing FreeBSD's TTY layer

Ed Schouten, ed@80386.nl

May 8, 2008

Thanks...

Contents

TTY's?

- In-kernel TTY layer provides programming interface for serial transmission.
- TTY's can be used as call-in devices (getty's, etc).
- TTY's can also be used as call-out devices (minicom).
- Somewhat integrated in the process code (signal delivery, accounting, etc).

Problems with the TTY layer

- It still uses the Giant lock and cannot easily be pushed down.
- Hotplugging/garbage collecting is broken. Example: PTY driver.
- Buffer mechanism (clist) design is fragile and inefficient.

My assignment

- Research on FreeBSD's TTY implementation, to improve locking, hotplugging and performance.
- Dissertation internship for BASc at Fontys University of Applied Sciences in Eindhoven, Netherlands.
- Employer/sponsor is Snow B.V.

Patching or rewriting?

- Driver model would likely break anyway.
- Breaking the TTY layer makes it hard to debug.
- Earlier attempts to refactor the old TTY layer have stalled.

- Temporary addition of a second TTY layer to the kernel.
- Port the existing drivers one by one.

Locking

- Simple locking approach: per-TTY mutex.
- Mutex can be set to Giant to ease migration, used by sc(4).
- Most drivers can just use the per-TTY mutex to lock their internals.
- Old clist buffers had a global free list. Hard to lock down.

Hotplugging

- No more trailing PTY's. They are destroyed when unused.
- Driver abandons device by calling tty_rel_gone().
- When all threads, descriptors and references to the TTY are gone, ttydevsw_free() is called to inform the driver.

Buffering

- Two input queues merged to one.
- Input queue supports all existing semantics, without excessive copying.
- No more cfreelist - TTY holds free blocks. Eases locking.
- When storage is large enough (about 95% of the time), read() calls are unbuffered.

Things that already work

- Most tools seem to work fine.
- Giantless uart(4) and pts(4) drivers.
- Giantless kern_exit.c and kern_proc.c.
- ptycompat(4) driver implements classic BSD PTY naming.
- sc(4) works, but uses the Giant. Tested on i386, amd64 and sparc64.
- ABI should be compatible, except sgtty. FreeBSD 5.2.1 still works inside a Jail.

Things that are missing

- Line disciplines: PPP, SLIP, Netgraph.
- ISA drivers: cx, digi, rc, rp.
- USB drivers: ubser, ucycom, ufoma.
- Misc drivers: dcons, nmdm.
- Input flow control and parity marking is not finished yet.

How to get this integrated

- Patchset should not be committed at once.
- Step 1: commit all self-supporting patches first.
- Step 2: split off the console and clist routines.
- Step 3: the big commit, replace the TTY layer, without changing the binary interface.
- Step 4: commit any trailing patches, C library changes.

Questions

Questions, anyone?