Hurd’s PCI arbiter

Samuel Thibault
& Joan Lledo

2018 February 3rd
The Hurd is all about freedom #0

“The freedom to run the program, for any purpose”

I.e.:

- Freedom from sysadmin!
  - WTH is fdisk/mke2fs/... hidden in /sbin?
  - I should be able to just work with my disk/network access

- Freedom to innovate
  - Experimental filesystem, personal work-flow, new kind of process combination,…
  - Give a PCI card function to a process

- Freedom from misbehaving programs and drivers
Micro-kernel layering

Kernel

Tasks, memory, IPC

- ext2fs
- pfinet
- auth
- proc
- root

user

- sh
- cp
Micro-kernel layering

ext2fs
auth
pfinet
proc
root

Kernel

user

Tasks, memory, IPC
Micro-kernel layering

- Server crash? Not a problem
  - “Computer bought the farm” is just an error, not something-of-the-death
- Easier to debug/tune
  - Just run gdb, gprof, …
- Can dare crazy things
  - The Hurd console has dynamic font support
    - See chinese support in pseudo-graphical mode (actually pure VGA textmode!) of Debian installer.
    - And Emojis!
- Kernel only handles Tasks, memory, IPC
Hurd possibilities

Kernel

ext2fs
pfinet
auth
proc
ftpfs
user
root
isofs
sh
cp
Hurd possibilities

€ settrans -c ~/ftp: /hurd/hostmux /hurd/ftpfs /

(just once for good)

€ settrans -a ~/mnt /hurd/iso9660fs

€ ls ~/mnt

README-or-FAIL

...

• Only downloads what is needed.
• Can be permanently stored in ext2fs

€ settrans ~/.signature /hurd/run /usr/games/fortune
But also

€ remap /bin/sh $HOME/bin/sh
€ remap /bin $HOME/unionbin

...

• Check out Stow/Nix/Guix!
How does it work?

Kernel

ext2fs
pfinet
root
auth
proc
ftpfs
user
isofs
libc
sh
libc
cp
- **Everything** is an (interposable) RPC
- Translators exposed in the FS
  - The user gets to decide what/how to interpose
    - Without need for costly ptrace or fragile libc symbols interposition.
    - **Native** fakeroot/chroot
    - Fully virtualized and fine-grained interface
  - Just need to use what's provided by the admin, e.g.
    - $HOME/
    - TCP/IP stack
    and pile over it
Hurd possibilities (cont'ed)

Kernel
Hurd possibilities (cont'ed)

i.e. ISO image inside a partitioned disk image on ftp over a VPN
Hurd userland network support

/servers/socket/2

root

Kernel

eth0

pfinet

w3m

user
Hurd userland network support

DDE layer
Linux drivers
/dev/eth0

root

Kernel

pfinet

eth0

w3m

user
Hurd userland network support

Kernel

root

eth0

eth-filter

w3m

user

pfinet
Hurd userland network support

/servers/socket/2

root

Kernel

eth0

eth-filter

pfinet

openvpn

~/servers/socket/2

~/servers/tun0

w3m

pfinet

user
Accessing PCI cards

Eth0 network drivers (netdnde) access PCI config space at bootup

- Xorg too
- Rump sound drivers too
- ...

But that’s not concurrent-access-safe :)

Accessing PCI cards

Translator for concurrent access to PCI config

- Sitting on /servers/pci/<dom>/<bus>/<dev>/<fn>
- Provides pci_conf_read/write, get_dev_regions, get_dev_rom
- Enough to implement libpciaccess & pciutils backends
Accessing PCI cards

- pfinet
- eth0
- Xorg
- pcm
- w3m
- firefox

PCI arbiter

Kernel

root

user
Accessing PCI cards as user

While at it, let’s allow Unix user permission configuration

• Give PCI card access on the fly with
  • fsysopts /servers/pci --uid 1234 --p 00:1f.3
  • (or configure it permanently with settrans)

• User app can then
  • Read/write config
  • TODO: map resources / ROM
  • TODO: get i/o port access token
Accessing PCI cards as user

Kernel

PCI arbiter

root

nobody

user

eth0 -> pfinet -> w3m

Xorg

pcm

firefox
Accessing PCI cards as user

Kernel

root

nobody

PCI arbiter

Xorg

eth0

eth0

pfinet

w3m

User
Accessing PCI cards as user

Woah, give a PCI card to a user?!
- DMA access to the whole memory!

But IO-MMU can control that
- Just like PCI passthrough with hypervisors

And some cards provide various functions
- Give only one access to the card
Accessing PCI cards as user

- Kernel
- PCI arbirter
  - nobody
  - user
  - subuser
- PC arbirter
  - firefox
  - pcm

- Xorg
- eth0
- pfnet
- w3m
Current State

Hardware support

• i686
• start of 64bit support
  • Kernel boots completely, now missing RPC 32/64bit translation
• DDE Linux 2.6.32 drivers layer for network boards
  • In userland netdde translator!
• IDE, Xorg, …
• AHCI driver for SATA
• Xen PV domU
  • Required GNU Mach changes only
• Preliminary sound support through userland Rump
• No USB yet
Current State

Software support

• Quite stable
  • Have not reinstalled boxes for a decade.
  • Debian buildd keep building packages, no hang after weeks!

• ~80% of Debian archive builds out of tree
  • XFCE, almost gnome, almost KDE
  • Firefox (aka iceweasel), gnumeric, …

• Standard native Debian Installer
Recent work

• GNU Guix and GuixSD
  • A pure GNUish GNU/Hurd distro!
  • Proper bootstrap of the Hurd chain
    - Used by Debian GNU/Hurd rebootstrap effort
  • A bit more work to be bootable
• Using xattr for storing translator records
• Various optimizations and stabilization
  • Protected payloads
  • Paging management
  • Message dispatch
• Gsync ~= futex, used in glibc & libpthread
Recent work (2)

- Highmem support (>4GB mem)
- Unprivileged subhurds
  - Think of containers, but safe to run by users, by construction
- LwIP TCP/IP stack
- Distributed system (netmsg)
Releases

- Nice 0.401 release on April 2011.
- Arch Hurd LiveCD release on August 2011.
- Hurd 0.9, Mach 1.8, MIG 1.8
- Released Debian-unofficial
  - wheezy/sid snapshot CDs on May 2013 \o/
  - jessie/sid snapshot CDs on May 2015 \o/
  - stretch/sid snapshot CDs on June 2017 \o/
Future work

- {sound, usb, disk, net} Rump drivers
- x86_64 support
- Read-ahead
- Startup in scheme?
- Your own pet project?
Thanks!

- For listening
- And to the people working on all this
- http://hurd.gnu.org/
- http://www.debian.org/ports/hurd/
- The increasing irrelevance of IPC performance for microkernel-based Operating Systems