A roadmap for the Hurd?

Samuel Thibault

2019 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
auth
proc
root

user

sh
cp
Micro-kernel layering

ext2fs → auth
pfinet → proc
root
user
Kernel
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
- Can virtualize at a very fine grain
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
Current state

- Quite stable
  - Have not reinstalled boxes for a decade
  - Debian buildds keep building packages, no hang after weeks!
- ~75% of Debian archive builds out of tree
  - XFCE, gnome, KDE, …
- Support merged upstream: gcc, glibc, llvm
- Ongoing: go, rust
- Debian distribution
- GuixSD/Arch ongoing
A roadmap?

- Many many existing bits, just needs polishing
  - The tail of “90% support done in 10% time, 10% support done in 90% time”
  - But would be so cool to have really working!
- Many ongoing project ideas
  - Would welcome help
- Many crazy ideas
  - Would welcome experimenting
Bits to polish

- httpfs/ftpfs
  - dpkg -iO *20181030-2*.deb

- find . -name emacs/*
Bits to polish

- Namespace-based translators
  - cd software.tar.gz,/
  - cat README
- cd disk.img,,part1,,ext2fs/
Bits to polish

- mboxfs
  - cd mbox,,/
  - mkdir that_man
  - mv $(grep -l that@mail *) that_man
Bits to polish

- Xmlfs
  - cd index.html,,/
  - cd html/body/
  - for i in $(find . -name h1); do cat $i/.text ; done
Bits to polish

- High-level language bindings for translators
  - Perl (pith)
  - Lisp
  - Java
  - Python
Bits to polish

- Some more in hurdextras/
  - jfs
  - notice
  - run
  - unionfs
Ongoing work, welcomes help

- PCI arbiter
  - Safe concurrent access to PCI config space
  - Seed FOSDEM’18 talk *Hurd’s PCI arbiter*
  - Could use an IO-MMU to make it safe
- ACPI translator
  - Provide access to ACPI operations
    - System shutdown…
- Rump translator (and libguestfs?)
  - Hardware support, filesystems support
  - See FOSDEM’16 talk *Hurd, Rump kernel, sound, and USB*
- 64bit kernel support
  - Then on 64bit userland support
- SMP support
No driver in the kernel?

- Minimal kernel support
  - Tasks, memory, IPC
- But also, ATM
  - Disk drivers
- We moved network drivers to userspace thanks to netdde
  - See FOSDEM’14 talk GNU/Hurd DDE userland device drivers
- Move disk drivers?
Moving disk drivers to userspace?

- What’s the issue?
  - Booting the whole stuff :)
  - One more grub-loaded module
Hurd boot

Kernel

Tasks, memory, IPC, disk
Hurd boot

Kernel

Tasks, memory, IPC, disk

root

startup

proc

auth

ext2fs

exec
Hurd boot

Kernel

Tasks, memory, IPC, disk

startup
proc
auth
ext2fs
exec
init
Hurd boot

Kernel: Tasks, memory, IPC, disk

root

ext2fs
proc
auth
startup
pfinet
netdde

init
Hurd boot, userland disk

Kernel

Tasks, memory, IPC
Hurd boot, userland disk

Kernel

Tasks, memory, IPC

startup
proc
auth
ext2fs
exec
root
disk
Hurd boot, userland disk

Kernel

Tasks, memory, IPC
Hurd boot, userland disk

Kernel  Tasks, memory, IPC

- netdde
- pfinet
- startup
- proc
- auth
- ext2fs
- root
- disk
- init
- exec
Conclusion

• A lot of nice things to achieve in GNU/Hurd
  • Really finish them
    - It’s fun to hack something
    - It’s better to have it finished :)

• Making a microkernel-based OS a real thing is hard
  • GNU/Hurd is almost there with Debian/Guix/Arch
    - Just needs your help :)

Thanks!

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