GNU/Hurd
AKA
Extensibility from the Ground

Samuel Thibault

2011 August 26th
<marcus> Jeroen: you are a Hurd developer. Being insane is part of the public image.
It's all about freedom #0

Extensibility for the user

- Mount one's own files
  - Access archives content
  - Access remote files
  - Experiment with filesystems
- Access one's own network
  - Access remote networks / VPN
  - Access virtual machine network
- Redirect one's sound
  - Through network
  - Sound effects
  - Recording
- ...
Outline

• The big hammer
• Traditional monolithic layering
• Gvfs layering
• FUSE layering
• Micro-kernel layering
Outline

- The big hammer
- Traditional monolithic layering
- Gvfs layering
- FUSE layering
- Micro-kernel layering
- Hurd possibilities
Outline

- The big hammer
- Traditional monolithic layering
- Gvfs layering
- FUSE layering
- Micro-kernel layering
- Hurd possibilities
- So what?
Outline

- The big hammer
- Traditional monolithic layering
- Gvfs layering
- FUSE layering
- Micro-kernel layering
- Hurd possibilities
- So what?
- Yet more fun
The big hammer

Just run KVM/qemu/virtualbox/whatever!

- Quite slower
  - and kvm not available to users by default
- Communication between guest and host is a burden

Goal: having extensibility simply at the shell prompt
Traditional monolithic layering

VFS

EXTxFS  ISOFS  NFS

Kernel

IP

ifconfig
mount

root

sh

user

cp
Traditional monolithic layering

- User mounts through “users” option
  - Need to ask root
    - and frowned upon
  - Only kernel-provided filesystems
- User network through tap
  - Need to ask root
  - No firewall tuning support
gvfs layering

VFS
EXTxFS  ISOFS  NFS
Kernel

IP

ifconfig
mount
sh
cp
gedit
gvfs

root
user
gvfs layering

- Supports a lot of nice features
  - Transparent ftp, webdav, smb, ...
- Only works for gnome applications
  - Not even in gnome-terminal shells
  - Not easily extensible
- i.e., does not compose well.
FUSE layering, user

VFS
FUSE
EXTxFS
ISOFS
NFS

Kernel

root

ifconfig
mount

user

fuse

sh

ip

sh

cp
FUSE layering

- Provides a lot of nice features, but
- Does not combine well by default
  - `cd ~/.avfs/#ftp:ftp.gnu.org/.../coreutils-6.9.tar.bz2#`
  - does not work
- Does not optimize well by default
  - `fuseiso9660 ~/.avfs/#ftp:ftp.gnu.org/.../foo.iso ~/mnt`
  - downloads it all!
- Does not provide all root features by default
  - How to deal with partitioned disk image?
    - `e2fsck what?`
→ Users are still second-class citizens
Micro-kernel layering

ext2fs  auth  pfinet  proc  root  user  sh  cp

Kernel  Tasks, memory, IPC
Micro-kernel layering

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 of Debian installer.
- Kernel only handles Tasks, memory, IPC
Hurd possibilities

Kernel

ext2fs
pfinet
auth
proc
ftpfs
root
user
sh
isoofs

cp
Hurd possibilities

$ settrans ~/ftp: /hurd/hostmux /hurd/ftpfs /
(just once for good)

$ settrans ~/mnt /hurd/iso9660fs

$ ls ~/mnt

README-or-FAIL

...  

- Only downloads what is needed.
- Can be permanent

settrans ~/.signature /hurd/run /usr/games/fortune
How does it work?

- ext2fs
- proc
- pfinet
- auth
- ftpfs
- isofs
- sh
- libc
- cp
- libc

Kernel
Hurd possibilities (cont'ed)

Kernel

- `ext2fs`
- `pfinet`
- `proc`
- `auth`
- `ftpfs`
- `pfinet`
- `part`
- `ext2fs`
- `sh`
- `cp`
- `isofs`
- `user`
Hurd possibilities (cont'ed)

i.e. ISO image inside a disk image on ftp over a VPN
Hurd possibilities (cont'ed)

- VPN running as user
- Parted running as user
- Chroot as user
- ...
- No less power than root
  - Since root uses the same mechanism anyway!
  - Except hardware access, of course
    - And still, could be interfaced safely thanks to I/O MMU
- More power for everybody (root and non-root)
  - Combine translators, invent new ones without kernel programming, ...
So, FUSE vs Hurd?

• FUSE is here
  • At last!
  • Not available by default on Linux installations
  • Does not permit everything to users

• Hurd is here too
  • At last!
  • Provides everything by default
    – Including ioctlS, arbitrary RPCs actually, could have a node which is a file, AND a directory, AND a CD drive,...
  • Permits everything to users
    – Hardware access is controlled
Neighbour Hurds

Kernel
Neighbour Hurds

Kernel

user sh cp ext2fs auth pfinet

user sh cp ext2fs auth

root proc

root proc
Neighbour/Sub-Hurd

Looks like Linux containers

- Except they can be combined in many ways, including recursive
  - Since it is simply the standard features in the Hurd
  - Since it is safer, because ext2fs, pfinet, etc. are not shared
- And complete
  - Since that's how a normal Hurd system is structured already.
    - Linux containers have a hard time being completely contained, e.g. sound?
Current State

- **Hardware support**
  - DDE Linux 2.6 drivers layer for network boards
  - IDE, SCSI, PCMCIA, Xorg, ...
  - Xen domU

- **Software support**
  - ~68% of Debian archive
    - XFCE, almost gnome, almost KDE
    - Firefox (aka iceweasel), gnumeric, ...
  - Standard Debian Installation CD
  - Plan to release in Debian Wheezy
  - Arch Hurd LiveCD and GHAMP
People at work nowadays

- Emilio Pozuelo Monfort: gnome
- Jeremie Koenig: glibc, openjdk
- Olaf Buddenhagen: community, mentor
- Pino Toscano: KDE
- Samuel Thibault: debian installer, autobuilders
- Thomas Schwinge: GNU gdb, gcc
- And various porters: Gabriele Giacone, Svante Signell, ...
- You're welcome!
Thanks!

- http://hurd.gnu.org/
- http://www.debian.org/ports/hurd/
- The increasing irrelevance of IPC performance for microkernel-based Operating Systems