The GNU/Hurd architecture, nifty features, and latest news

Samuel Thibault

2013 February 2nd
It's 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,...

- Also provide freedom from misbehaving programs
It's all about freedom #0

From: xxx <xxx@yyy.fr>
Subject: Network expertise
Date: Thu, 31 Jan 2013 12:37:34 +0100

[...] Would it be possible to route to my VPN the traffic of only one application?

Actually, also well-known classical issue of full-VPN: traffic of the VPN itself shouldn't go through the VPN!

And yet, here root capabilities!!

Spoiler: Yes, GNU/Hurd can already do it. Without asking root.
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

- Monolithic/Gvfs/FUSE/micro-kernel layering
- Hurd layering
- Hurd nifty features
- Present and future
Traditional monolithic layering

Kernel

VFS

EXTxFS  ISOFS  NFS

IP

root

user

ifconfig

mount

sh

cp

sh
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 personal firewall tuning support
gvfs layering

ifconfig
mount

sh
cp
gedit
gvfs

root

user

VFS
EXTxFS
ISOFS
NFS
Kernel

IP
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

Kernel

VFS

FUSE

EXTxFS

ISOFS

NFS

IP

root

user

ifconfig

mount

sh

cp

fuse

sh

cp

10
FUSE layering, user

```
ifconfig
mount
root

VFS
FUSE
EXTxFS
ISOFS
NFS

Kernel

user

fuse
sh
cp

IP
```
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

Kernel

Tasks, memory, IPC

ext2fs
auth
pfinet
proc
root

user

sh
cp
Micro-kernel layering

Kernel

Tasks, memory, IPC

ext2fs
auth
root
user
pfinet
proc

sh

cp

sh

cp
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.
- Kernel only handles Tasks, memory, IPC
Hurd possibilities

Kernel

- ext2fs
- pfinet
- auth
- proc

ftpfs

sh
cp

user

root
Hurd possibilities

€ settrans ~/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
How does it work?
**Rationale**

- **Everything** is a (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
Example: interpose TCP/IP stack

\[ \text{\texttt{\$ settrans -ca \$HOME/servers/socket/2}} \]
\[ \text{\texttt{/hurd/pfinet -i \$HOME/servers/tun0}} \]
\[ \text{\texttt{\$ hexdump \$HOME/servers/tun0 \&}} \]
\[ \text{\texttt{\$ ~/remap/remap.sh /servers/socket/2 \}} \]
\[ \text{\texttt{\$HOME/servers/socket/2}} \]
\[ \text{\texttt{\$ wget \texttt{www.gnu.org}} \]

- My own translator
- Can now plug my own VPN software
- Only wget accesses it (well, the shell too :) )
But also

\[ ~/\text{remap/\text{remap.sh}} \ /	ext{bin/sh} \ \$\text{HOME/bin/sh} \]

\[ ~/\text{remap/\text{remap.sh}} \ /	ext{bin} \ \$\text{HOME/unionbin} \]

...

- Check out Guix!
Hurd possibilities (cont'ed)
Hurd possibilities (cont'ed)

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

- VPN running as user
- Parted running as user
- Chroot running as user
- Tarfs, Xmlfs, cvsfs, httpfs, gopherfs, ...
- ...
- No less power than root
  - Since root uses the same mechanism anyway!
  - Except direct hardware access, of course
    - And still, can chmod o+rw /dev/eth0
    - 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, ...
Neighbour Hurds

Kernel
Neighbour Hurds

Kernel

user → sh → cp → auth → ext2fs → proc
user → sh → cp → auth → ext2fs → proc
Sub-Hurd

Kernel

ext2fs
auth
pfinet
root

ext2fs
auth
pfinet
user/root

ext2fs
auth
pfinet
proc

sh
user

cp
Neighbour/Sub-Hurd

Looks like Linux containers

• Except they can be combined in many ways, including recursive
  • Simply the standard features in the Hurd
  • 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

- i686
- DDE Linux 2.6.32 drivers layer for network boards
  - In userland netdde translator!
- IDE, SCSI, PCMCIA, Xorg, ...
- Xen PV domU
- No USB, no sound, no SATA.
Current State

Software support

• Quite stable
  • I don't remember when I last reinstalled by system, several years ago at least. Used only for development, though.
  • Debian buildds keep building packages, usually hang after some weeks, out of some remaining memory leak.
• ~78% of Debian archive builds out of tree
  • XFCE, almost gnome, almost KDE
  • Firefox (aka iceweasel), gnumeric, …
• Standard Debian Installation CD
• Will release some unofficial Debian Wheezy CDs
• Nix-based distribution
Future work

- Make an unofficial Debian GNU/Hurd Wheezy release!!
- Xen PVH support
- SATA driver
- X86_64 support
- Language bindings for translators
- Read-ahead
- {hdd,sound,usb}dde?
- Official Debian GNU/Hurd Jessie?
- Your own pet project?
Hardware support

Æternam issue of all our nice micro-kernel projects

- KVM / Xen support
  - Leverage existing system
  - Not satisfactory, even if very good performance: users want to run on real hardware of course!

- DDE layer
  - Leverage Linux drivers
  - Still has to be maintained
  - Shared maintenance?
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
  