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 and drivers
Micro-kernel layering

- ext2fs
- auth
- pfinet
- proc
- root
- user
- sh
- cp

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.

• Kernel only handles Tasks, memory, IPC
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
Example: interpose TCP/IP stack

€ settrans -ca ${HOME}/servers/socket/2
   /hurd/pfinet -i ${HOME}/servers/tun0

€ openvpn ... ${HOME}/servers/tun0 &

€ remap /servers/socket/2
   ${HOME}/servers/socket/2

€€€ wget www.gnu.org

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

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

\[ \text{remap } /bin \text{ } \$\text{HOME/unionbin} \]

\[ \ldots \]

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

- ext2fs
- pfinet
- auth
- proc
- ftpfs
- sh
- libc
- cp
- libc
- isoofs
- user
- root

Kernel
Rationale

- **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)

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

```
/servers/socket/2  pfinet

root

Kernel

eth0

w3m

user
```
Hurd userland network support

DDE layer
Linux 2.6.32 drivers
/dev/eth0

root

Kernel

pfinet

eth0

w3m

user
DDE stack

Based on TU-Dresden's DDE stack

- Zheng Da's GSOC
- Ported to Mach kernel
- Ported to Mach device interface
- Updated libdde_linux26 for long-term-supported linux 2.6.32
  - Most drivers (and mostly the really useful ones) just work without patches
- Used by default by Debian GNU/Hurd
DDE stack

Linux drivers

Linux API

ddekit API

libddekkit

pkg_xmit

rx_callback

netdde

intr

vm_allocate_contiguous

kernel
DDE stack

- Only two additions to the kernel
  - Interrupt delivery and masking
  - Physically-contiguous memory allocation
  - (Direct I/O access was already available)
- Performance similar to in-kernel driver
- Driver in a separate process
  - Can just crash and be happy with it...
  - Can easily debug and profile them
  - Stack smashing protection ;)
  - Could benefit from I/O MMU for better isolation.
    - For now drivers can just access all RAM...
Rump sound support, v0
Rump sound support, v1

- BSD
- src-gnu
- librumpuser
- libpciaccess
- /dev/pcm
- intr
- vm_allocate_contiguous
- kernel
- mplayer
Rump sound support, v2

/dev/pcm

rumpuser

/home/samy/dev/pcm

rumpuser

PCI arbiter

IOMMU

intr vm_allocate_contiguous

kernel
Rump USB support

```
 ext2fs
 /dev/usbd0
 rump-usb-storage
     libusb
 /dev/usb
 rumpuser
 PCI arbiter
     IOMMU
 intr     vm_allocate_contiguous
 kernel
```
Rump USB support

```
ext2fs
/dev/usbd0
rump-usb-storage
/libusb
/dev/usb
rumpuser

intr
vm_allocate_contiguous
kernel
```
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 buildds keep building packages, no hang after weeks!
- ~80% of Debian archive builds out of tree
  - XFCE, almost gnome, almost KDE
  - Firefox (aka iceweasal), gnumeric, ...
- Standard *native* Debian Installer
Recent work

- GNU Guix
- Fixed native fakeroot
- SCM_CRED$
- Various optimizations
  - Node cache
  - Lockless reference counting
  - IPC table→radix tree
  - Kernel memory management
- New rpcscans tool
Releases

- Nice 0.401 release on April 2011.
- Arch Hurd LiveCD release on August 2011.
- Hurd 0.6 then 0.7, Mach 1.5 then 1.6, MIG 1.5 then 1.6
- Released Debian-unofficial
  - wheezy/sid snapshot CDs on May 2013 \
  - jessie/sid snapshot CDs on May 2015 \

Future work

- X86_64 support
- Read-ahead
- \{sound,usb\} Rump drivers
- GNU system: Guix/Hurd?
- Startup in scheme?
- Your own pet project?
Thanks!

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