

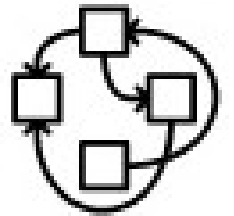
Indian GNU/Linux Users' Group, Chennai (ILUGC)

presents

GNU Hurd Workshop

conducted by

Shakthi Kannan, MS



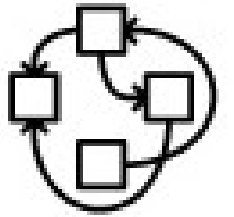
HURD

Venue

Madras Institute of Technology, Chennai



Introduction



HURD

Free software

- ✓ **Free software licenses**
- ✓ **Freedom from software patents**
- ✓ **User freedom**

Richard M Stallman
“The Danger of Software Patents”
(IIT-Madras)

<http://www.chennaiug.org/meeting/info/TheDangerOfSWPatent1of2.ogg>

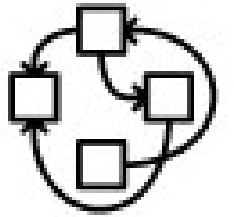
<http://www.chennaiug.org/meeting/info/TheDangerOfSWPatent2of2.ogg>

<http://www.cs.iitm.ernet.in/~ramk/stallman.wav>



GNU Hurd

Overview

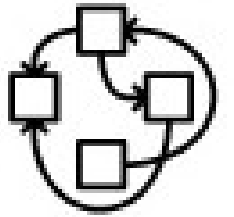


HURD

- **Multi-server**
- **POSIX compliant**
- **User extensible system framework**
- **Top of GNU Mach microkernel**



GNU Hurd



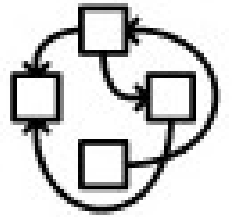
HURD

History

- 1983** GNU Project started by Richard Stallman
- 1988** Decision made to use Mach 3.0 as the kernel
- 1991** Mach 3.0 is released under compatible license
- 1991** Thomas Bushnell, BSG, starts the Hurd project
- 1994** The Hurd boots the first time
- 1997** Version 0.2 of the Hurd is released
- 1998** Debian hurd-i386 archive is created
- 2001** Debian GNU/Hurd snapshot three CD images



GNU Hurd



HURD

Kernel Architectures

Monolithic

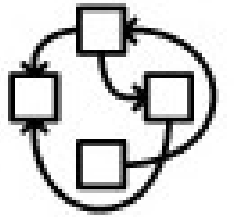
- **Device drivers**
- **Network protocols**
- **Process management**
- **Authentication**
- **File systems**
- **Scheduling**

Microkernels

- **Resource management**
- **Task management**
- **IPC**
- **Basic hardware support**



GNU Hurd



HURD

Single/Multi-server models

Single-server

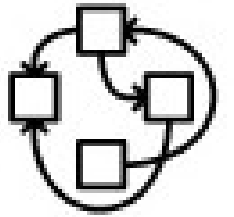
- **Single task**
- **Comparable to monolithic kernel**

Multi-server

- **Multiple tasks**
- **Cooperative**
- **Responsibilities distributed logically**
- **Stability**
- **Scalability**



GNU Hurd



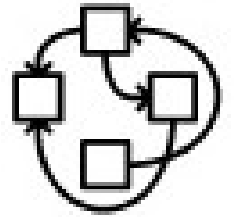
HURD

GNU Hurd allows users to:

- **write and run their own servers**
- **replace system servers dynamically with their own implementations**
- **decide what parts of the remainder of the system they want to use.**
- **extend the functionality of the system**

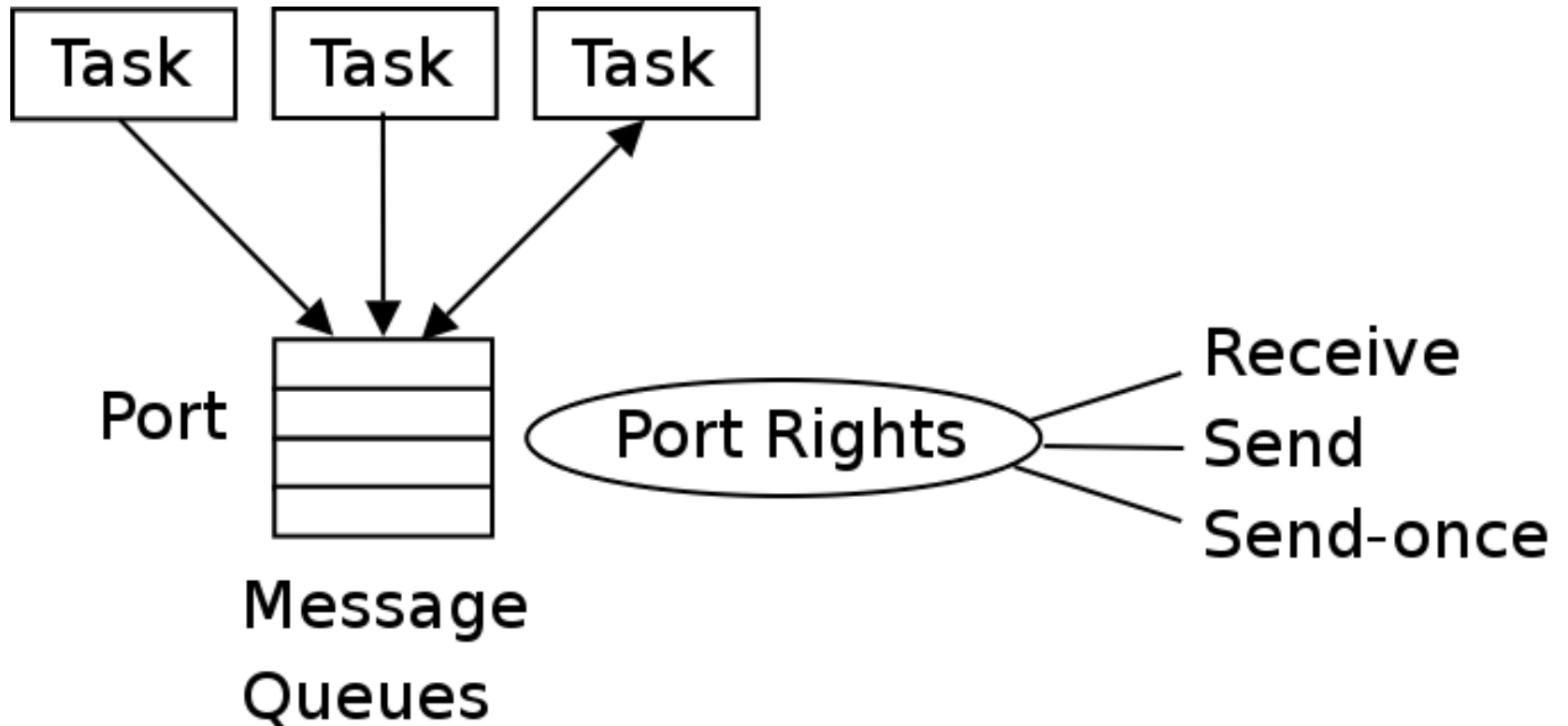


GNU Hurd



GNU Mach IPC

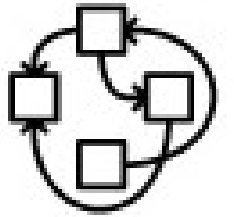
HURD



MIG – Mach Interface Generator



GNU Hurd



HURD

Obtaining a port

Traditional Mach

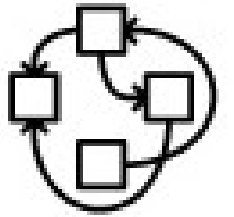
- **Nameserver provides ports to servers**
- **Nameserver port is itself provided by Mach**

Hurd

- **Filesystem provides nameserver space**
- **Root directory port is inserted into each task**
- **The C library finds other ports with**
hurd_file_name_lookup (pathname resolution)



GNU Hurd



HURD

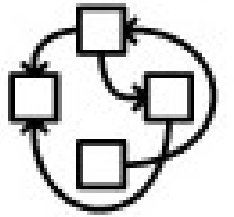
Example: *hurd_file_name_lookup*

```
mach_port_t identity;  
mach_port_t pwserver;  
kern_return_t err;  
pwserver = hurd_file_name_lookup  
    ("/servers/password");  
err = password_check_user (pwserver,  
    0 /* root */, "supass",  
    &identity);
```



GNU Hurd

POSIX Interface



HURD

File descriptor

fd = open(name,...)

read(fd, ...)

write(fd, ...)

fstat(fd, ...)

Port to server providing the file

dir_lookup(..,name,..,&port)

io_read(port, ...)

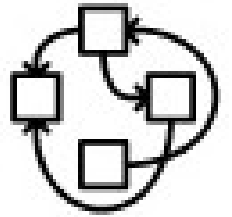
io_write(port, ...)

io_stat(port, ...)



GNU Hurd

File System Servers



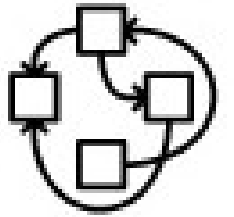
HURD

- **Provide file and directory services for ports**
- **Ports are returned by a directory lookup**
- **Translate filesystem accesses through their root path (hence the name translator)**
- **The C library maps the POSIX file and directory interface to RPCs to the filesystem servers ports**
- **Users can install file system servers on inodes they own**



GNU Hurd

Active Translators



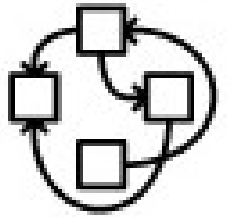
HURD

- **settrans -a /cdrom /hurd/isofs /dev/hd2**
- **Are running filesystem servers**
- **Are attached to the root node they translate**
- **Run as a normal process**
- **Go away with every reboot, or even time out**



GNU Hurd

Passive Translators

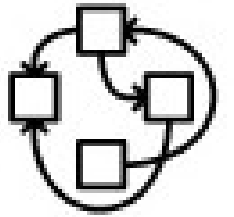


HURD

- `settrans /mnt /hurd/ext2fs /dev/hd1s1`
- Are stored as command strings into an inode
- Are used to start a new active translator if there isn't one
- Startup is transparent to the user.
- Startup happens the first time the server is needed
- Are permanent across reboots



GNU Hurd



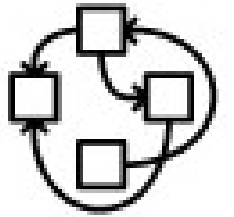
Authentication (auth) Server HURD

- **A user identity is just a port to an authserver**
- **Four set of ids for it:**
 - effective user ids**
 - effective group ids**
 - available user ids**
 - available group ids**
- **A 0 among the user ids identifies the superuser**



GNU Hurd

Password Server



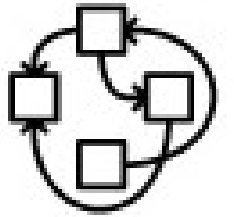
HURD

- `/servers/password`
- **It runs as root**
- **It returns a new authentication port in exchange for a password**
- **The ids corresponding to the authentication port match the unix user and group ids**
- **Support for shadow passwords**



GNU Hurd

Filesystems



HURD

- **Store based filesystems**

 - Ext2fs, ufs, fatfs (development)**

 - isofs (iso9660, RockRidge, GNU extensions)**

- **Network file systems**

 - Nfs, ftpfs**

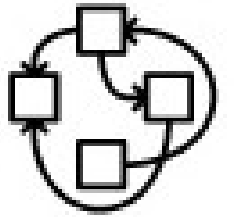
- **Miscellaneous**

 - Hostmux, usermux, tmpfs (development)**



GNU Hurd

Development Libraries



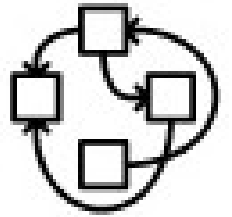
HURD

- **libdiskfs** **store based filesystems**
- **libnetfs** **network filesystems,**
virtual filesystems
- **libtrivfs** **simple filesystems**
- **libstore** **store media abstractions**



GNU Hurd

References



HURD

[1] Marcus Brinkmann. 2001. *Talk about the Hurd.*

<http://www.gnu.org/software/hurd/hurd-talk.html>

[2] Wolfgang Jährling. 2002. *Hurd Hacking Guide.*

<http://www.gnu.org/software/hurd/hacking-guide/hhg.html>

[3] GNU Hurd User's Guide.

http://www.gnu.org/software/hurd/users-guide/using_gnuhurd.html



Thank You