

GNU Hurd

From Wikipedia, the free encyclopedia

GNU Hurd (usually referred to as **the Hurd**) is the multiserer microkernel written as part of GNU. It has been under development since 1990 by the GNU Project of the Free Software Foundation, designed as a replacement for the Unix kernel,^[2] and released as free software under the GNU General Public License.

GNU Hurd consists of a set of protocols and server processes (or daemons, in Unix terminology) that run on the GNU Mach microkernel.^[2] The Hurd aims to surpass the Unix kernel in functionality, security, and stability, while remaining largely compatible with it. The GNU Project chose the multiserer microkernel (http://www.gnu.org/software/hurd/faq/multiserer_microkernel.html) for the operating system, due to perceived advantages over the traditional Unix monolithic kernel architecture.^[3]

Contents

- 1 Name and logo
- 2 Development history
- 3 Architecture
 - 3.1 Other microkernels
 - 3.2 Unix extensions
 - 3.3 Architecture of the servers
 - 3.3.1 Core servers
 - 3.3.2 Filesystem servers
- 4 GNU distributions running Hurd
- 5 See also
- 6 References
- 7 External links

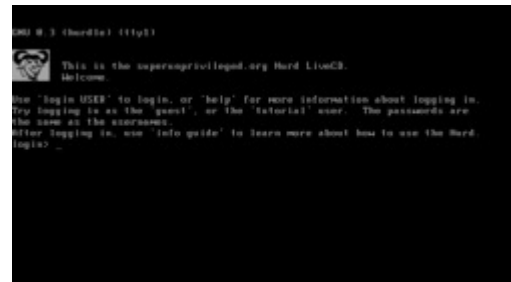
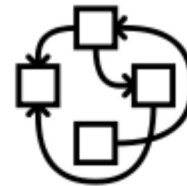
Name and logo

In December 1991 the primary architect of the Hurd described the name as a mutually recursive acronym:^[4]

It's time [to] explain the meaning of "Hurd". "Hurd" stands for "Hird of Unix-Replacing Daemons". And, then, "Hird" stands for "Hurd of Interfaces

Representing Depth". We have here, to my knowledge, the first software to be named by a pair of

GNU Hurd



Company / developer	GNU Project Thomas Bushnell Roland McGrath Marcus Brinkmann Neal Walfield
Programmed in	Assembly, C
OS family	Unix-like
Working state	Current
Source model	Free and open source software
Latest stable release	0.5 / 27 September 2013 ^[1]
Supported platforms	i386 architecture
Kernel type	multiserer microkernel (http://www.gnu.org/software/hurd/faq/multiserer_microkernel.html)
Userland	GNU and others
Default user interface	Bash
License	GNU General Public License
Official website	www.gnu.org/software/hurd/ (http://www.gnu.org/software/hurd/)

mutually recursive acronyms.

—Thomas (then Michael) Bushnell

As both *hurd* and *hird* are homophones of the English word *herd*, the full name *GNU Hurd* is also a play on the words *herd of gnus*, reflecting how the kernel works.^[5] The logo is called the *Hurd boxes* and it also reflects on architecture. The logo is a graph where nodes represent the Hurd kernel's servers and directed edges are IPC messages.^[4]

Due to the state of microkernel technology, the doubly recursive acronym is sometimes referred to by experts as the most interesting feature of GNU Hurd.^[6]

Development history

Richard Stallman founded the GNU project in September 1983 with an aim to create a free GNU operating system. Initially the components required for kernel and development were written: editors, shell, compiler and all the others. By 1989, GPL came into being and the only major component missing was the kernel.^{[7][8]}

Development on the Hurd began in 1990 after an abandoned kernel attempt in 1986, based on the research TRIX operating system developed by Professor Steve Ward and his group at MIT's Laboratory for Computer Science (LCS).^[9] According to Thomas Bushnell, the initial Hurd architect, their early plan was to adapt the 4.4BSD-Lite kernel and, in hindsight, "It is now perfectly obvious to me that this would have succeeded splendidly and the world would be a very different place today".^[10] In 1987 Richard Stallman proposed using the Mach microkernel developed at Carnegie Mellon University. Work on this was delayed for three years due to uncertainty over whether CMU would release the Mach code under a suitable license.^[9]

With the release of the Linux kernel in 1991, the primary user of GNU's userland components soon became operating systems based on the Linux kernel (Linux distributions), prompting the coining of the controversial term *GNU/Linux*.

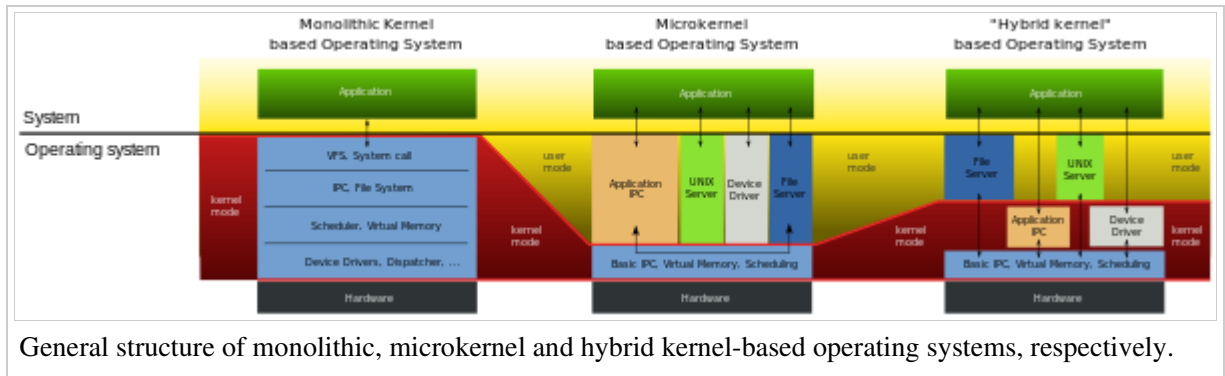
Development of the Hurd has proceeded slowly. Despite an optimistic announcement by Stallman in 2002 predicting a release of GNU/Hurd later that year,^[11] the Hurd is still not considered suitable for production environments. Development in general has not met expectations, and there are still bugs and missing features.^[12] This has resulted in a poorer product than many (including Stallman) had expected.^[13] In 2010, after twenty years under development, Stallman said that he was "not very optimistic about the GNU Hurd. It makes some progress, but to be really superior it would require solving a lot of deep problems", but added that "finishing it is not crucial" for the GNU system because a free kernel already existed (Linux), and completing Hurd would not address the main remaining problem for a free operating system: device support.^[14]

The Debian project, among others, have worked on the Hurd project to produce binary distributions of Hurd-based GNU operating systems for IBM PC compatible systems.

Architecture

Unlike most Unix-like kernels, the Hurd uses a server–client architecture, built on a microkernel that is responsible for providing the most basic kernel services – coordinating access to the hardware: the CPU (through process management and scheduling), RAM (via memory management), and other various input/output devices (via I/O scheduling) for sound, graphics, mass storage, etc. In theory the microkernel

design would allow for all device drivers to be built as servers working in user space, but today most drivers of this kind are still contained in the GNU Mach kernel space.^[15]



According to Hurd developers the main advantage of microkernel-based design is the ability to extend the system: developing a new module would not require in depth knowledge of the rest of the kernel and a bug in one module would not crash the entire system. Hurd provides a concept of *translators*, a framework of modules used to extend a file system functionality.^[16]

From early on, the Hurd was developed to use GNU Mach as the microkernel. This was a technical decision made by Richard Stallman, who thought it would speed up the work by saving a large part of it. He has admitted that he was wrong about that.^[17] Other Unix-like systems working on the Mach microkernel include OSF/1, Lites, and MkLinux. Mac OS X and NeXTSTEP use hybrid kernels based on Mach.

Other microkernels

From 2004 onward, various efforts were launched to port the Hurd to more modern microkernels. The L4 microkernel was the original choice in 2004, but progress slowed to a halt. Nevertheless, during 2005, Hurd developer Neal Walfield finished the initial memory management framework for the L4/Hurd port, and Marcus Brinkmann ported essential parts of glibc; namely, getting the process startup code working, allowing programs to run, thus allowing the first user programs (trivial ones such as the hello world program in C) to run.

Since 2005 Brinkmann and Walfield started researching Coyotos as a new kernel for HURD.^{[18][19]} In 2006, Brinkmann met with Jonathan Shapiro (a primary architect of the Coyotos Operating System) to aid in and discuss the use of the Coyotos kernel for GNU/Hurd. In further discussion HURD developers realised that Coyotos (as well as other similar kernels) are not suitable for HURD^[citation needed].

In 2007, Hurd developers Neal Walfield and Marcus Brinkmann gave a critique of the Hurd architecture, known as "the critique",^[20] and a proposal for how a future system may be designed, known as "the position paper".^[21] In 2008, Neal Walfield began working on the Viengoos microkernel as a modern native kernel for HURD. As of 2009, development on Viengoos is paused due to Walfield lacking time to work on it.^[22]

In the meantime, others have continued working on the Mach variant of Hurd.^[23]

Unix extensions

A number of traditional Unix concepts are replaced or extended in the Hurd.

Under Unix, every running program has an associated user id, which normally corresponds to the user that started the process. This id largely dictates the actions permitted to the program. No outside process can change

the user id of a running program. A Hurd process, on the other hand, runs under a *set* of user ids, which can contain multiple ids, one, or none. A sufficiently privileged process can add and remove ids to another process. For example there is a password server that will hand out ids in return for a correct login password.

Regarding the file system, a suitable program can be designated as a *translator* for a single file or a whole directory hierarchy. Every access to the translated file, or files below a hierarchy in the second case, is in fact handled by the program. For example a file translator may simply redirect read and write operations to another file, not unlike a Unix symbolic link. The effect of Unix *mounting* is achieved by setting up a filesystem translator (using the "settrans" command). Translators can also be used to provide services to the user. For example, the ftpfs translator allows a user to encapsulate remote FTP sites within a directory. Then, standard tools such as ls, cp, and rm can be used to manipulate files on the remote system. Even more powerful translators are ones such as UnionFS, which allows a user to unify multiple directories into one; thus listing the unified directory reveals the contents of all the directories.

The Hurd requires a multiboot-compliant boot loader, such as GRUB.

Architecture of the servers

According to the Debian documentation there are 24 servers (18 core servers and 6 file system servers) named as follows:^[24]

Core servers

- **auth** (authentication server): Receives requests and passwords from programs and gives them an ID, which changes the privileges of the program.
- **crash** (crash server): Handles all fatal errors.
- **exec** (execution server): Translates an executable image (currently ELF and a.out are supported) to a runnable image in memory.
- **fifo** (FIFO translator): Implements named pipes.
- **new-fifo** (new FIFO server): An alternate server for named pipes.
- **firmlink** (the firmlink translator): Implements firmlinks – "half-way between a symbolic link and a hard link".^[25]
- **fwd** (forward server): Forwards requests to other servers, used by fifo and symlink servers.
- **hostmux** (host multiplexer server)
- **ifsock** (server for sockets interface): Helps with UNIX domain socket addresses.
- **init** (init server): Basic system booting and configuration.
- **magic** (magic server)
- **null** (null server): Implements /dev/null and /dev/zero.
- **pfinet** (pfinet server): Implements the PF_INET protocol family.
- **pflocal** (pflocal server): Implements UNIX domain sockets.
- **proc** (process server): Assigns PIDs and manages process-level actions.
- **symlink** (symbolic link translator): Implements symbolic links for filesystems that don't support them.
- **term** (terminal server): A POSIX terminal.
- **usermux** (user multiplexer server): Invokes user-specific translators.

Filesystem servers

ext2fs

The ext2 filesystem translator. It receives disk blocks from the microkernel and gives files and directories to the applications.

isofs

The translator for the ISO 9660 filesystem. Translates blocks of a CD or DVD to files and directories for the applications.

nfs

See Network File System.

ufs

Translator for the BSD filesystem of the same name, UFS.

ftpfs

File transfer protocol filesystem translator.

storeio

The storage translator.

The servers collectively implement the POSIX API, with each server implementing a part of the interface. For instance, the various filesystem servers each implement the filesystem calls. The storage server will work as a wrapping layer, similar to the block layer of Linux. The equivalent of VFS of Linux is achieved by libdiskfs and libpager libraries.

GNU distributions running Hurd

Hurd-based GNU distributions include:

- Arch Hurd
- Debian GNU/Hurd
- NixOS^{[26][27]}
- Bee GNU/Hurd (discontinued)
- Gentoo GNU Hurd (discontinued)
- GNU/Hurd Live CD^[28] (discontinued)

See also

- Linux-libre
- Minix 3

References

- ↑ "Happy 30th birthday, GNU! GNU Hurd 0.5, GNU Mach 1.4, GNU MIG 1.4 released." (<http://www.gnu.org/software/hurd/news/2013-09-27.html>). 27 September 2013.
- ↑ ^{*a b*} "What Is the GNU Hurd?" (http://www.gnu.org/software/hurd/hurd/what_is_the_gnu_hurd.html). *GNU*. Retrieved 2010-03-04.
- ↑ "advantages" (<http://www.gnu.org/software/hurd/advantages.html>). *GNU*. Retrieved 2011-12-07.
- ↑ ^{*a b*} Vervloesem, Koen (July 7, 2010). "The Hurd: GNU's quest for the perfect kernel" (<http://lwn.net/Articles/395150/>). LWN.net. Retrieved October 5, 2012.
- ↑ "GNU Hurd: Origin of the Name" (http://www.gnu.org/software/hurd/hurd/what_is_the_gnu_hurd/origin_of_the_name.html). *GNU*. Retrieved 2010-03-04.
- ↑ Doeppner, Thomas W. "Operating System Design." Computer Science 167. Brown University. Providence, RI, Oct 1, 2011.

7. ↑ Hillesley, Richard (June 30, 2010). "GNU HURD: Altered visions and lost promise" (<http://www.h-online.com/open/features/GNU-HURD-Altered-visions-and-lost-promise-1030942.html>). Retrieved October 1, 2012.
8. ↑ "Linux and the GNU Project" (<http://www.gnu.org/gnu/linux-and-gnu.html>). *GNU*. 2010-01-26. Retrieved 2010-03-04.
9. ↑ ^{*a*} ^{*b*} "The GNU Hurd History, "How it Started"" (<http://www.gnu.org/software/hurd/history.html>). *GNU*. Retrieved 2006-08-27.
10. ↑ Salus, Peter H.. "The Hurd and BSDI" (<http://www.groklaw.net/article.php?story=20050727225542530>). *The Daemon, the GNU and the Penguin*. Retrieved 2006-08-08.
11. ↑ Ribeiro, John (2002-03-11). "Free Software Sees GNU Loose of Linux" (<http://www.pcworld.com/article/88464/article.html>). *PC World*. Retrieved 2012-10-05.
12. ↑ "Status" (<http://www.gnu.org/software/hurd/hurd/status.html>). *GNU*. Retrieved 2010-03-04.
13. ↑ Stallman, Richard (2006-03-09). "The Free Software Movement and the Future of Freedom" (http://mjsec.ffzg.hr/~dpavlin/stallman2006/free_software_movement_and_the_future_of_freedom_zagreb_09_march_2006.ogg) (ogg). *Free Software Foundation*. Zagreb. Lay summary (<http://fsfeurope.org/documents/rms-fs-2006-03-09.en.html>). "This is the way, also, that people thought was the cleanest possible way to design kernels back in 1990. Well, it took many many many years to get this kernel to run at all, and it still doesn't run well, and it looks like there may be fundamental problems with this design, which nobody knew about back in 1990."
14. ↑ Stallman, Richard (2010-07-29). "RMS AMA" (<http://blog.redd.it.com/2010/07/rms-ama.html>). Reddit. Retrieved 2011-12-07.
15. ↑ Kousoulos, Constantine (2007-03-21). "Re: Device drivers in Mach?" (<http://lists.gnu.org/archive/html/bug-hurd/2007-03/msg00089.html>). *bug-hurd mailing list*. <http://lists.gnu.org/archive/html/bug-hurd/2007-03/msg00089.html>.
16. ↑ Doeppner, Thomas W. (20 December 2010). *Operating Systems In Depth: Design and Programming* (<http://books.google.com/books?id=xX5tfrAQQ8cC>). John Wiley & Sons. p. 160. ISBN 978-0-471-68723-8. Retrieved 29 November 2012.
17. ↑ Stallman, Richard (2000-10-12). "In Defense of Red Hat" (<http://www.linuxtoday.com/infrastructure/2000101200421OPCYRH>). *Linux Today*. Retrieved 2011-12-07. "I take full responsibility for the technical decision to develop the GNU kernel based on Mach, a decision which seems to have been responsible for the slowness of the development. I thought using Mach would speed the work by saving us a large part of the job, but I was wrong."
18. ↑ Shapiro, Jonathan S. (2005-10-27). "Re: A comment about changing kernels" (<http://lists.gnu.org/archive/html/l4-hurd/2005-10/msg00755.html>). *l4-hurd mailing list*. <http://lists.gnu.org/archive/html/l4-hurd/2005-10/msg00755.html>.
19. ↑ Bachmann, Tom (2006-07-07). "Re: seL4, L4.sec and coyotos mess" (<http://lists.gnu.org/archive/html/l4-hurd/2006-07/msg00004.html>). *l4-hurd mailing list*. <http://lists.gnu.org/archive/html/l4-hurd/2006-07/msg00004.html>.
20. ↑ Walfield, Neal H.; Brinkmann, Marcus (July 2007). "A Critique of the GNU Hurd Multi-server Operating System" (<http://walfield.org/papers/200707-walfield-critique-of-the-GNU-Hurd.pdf>) (PDF). *GNU*. Retrieved 2011-12-07.
21. ↑ Walfield, Neal H.; Brinkmann, Marcus (2007-01-04). "Improving Usability via Access Decomposition and Policy Refinement" (<http://walfield.org/papers/20070104-walfield-access-decomposition-policy-refinement.pdf>) (PDF). *GNU*. Retrieved 2011-12-07.
22. ↑ "viengoos" (<http://www.gnu.org/software/hurd/microkernel/viengoos.html>). GNU. Retrieved 2010-03-04.
23. ↑ "What happened to the L4/Coyotos/viengoos micro-kernels?" (http://www.gnu.org/software/hurd/faq/which_microkernel.html). GNU. Retrieved 2011-01-07.
24. ↑ "Preliminary GNU/Hurd User Interface Description" (<http://www.debian.org/ports/hurd/hurd-doc-server>). *Debian*. 1996-10-10. Retrieved 2010-03-04.
25. ↑ "GNU/Hurd - Documentation" (<http://www.debian.org/ports/hurd/hurd-doc-server#firmlink>). Debian. 1996-10-10. Retrieved 2012-07-12.
26. ↑ nixos.org (<http://nixos.org/>)
27. ↑ "Nix-based GNU/Hurd System" (<http://www.gnu.org/software/hurd/hurd/running/nix.html>). Gnu.org. 2012-03-21. Retrieved 2012-07-12.
28. ↑ "GNU Hurd/ hurd/ running/ live cd" (http://www.gnu.org/software/hurd/hurd/running/live_cd.html). Retrieved November 19, 2012.

External links

- Official website (<http://www.gnu.org/software/hurd/>)

Retrieved from "http://en.wikipedia.org/w/index.php?title=GNU_Hurd&oldid=593257045"

Categories: [Microkernels](#) | [GNU Project software](#)

| [Operating system distributions bootable from read-only media](#) | [Unix variants](#)

- This page was last modified on 31 January 2014 at 10:08.
- Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#).
Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.