Installing the Hurd

Neal H. Walfield

1 Overview

The Debian GNU Hurd distribution, unlike distributions of other operating systems, does not have a nice installation program. One day it will and maybe you will help design and implement it; however, until that day, installing the GNU Hurd requires another operating system, specifically, another Unix-like system[1]. Users have indicated successful installations using different flavors of GNU/Linux as well as the BSDs. The minimum requirements of the bootstrap operating system are the ability: to create an ext2 file system; to extract a tar archive on to it; and to install GNU Grub.

The GNU Hurd is similar in nature to any Unix-like system: after logging in, the user is presented with a shell and the familiar Unix VFS, virtual filesystem. Although GNU tries to be POSIX compliant, it is not Unix. The Hurd builds upon many of the Unix concepts and extends them to either add new functionality or to fix what has been perceived as flaws in the original design. The most noticeable difference is translators, user space programs which interact with the VFS. These filesystems do not live living in the kernel nor do they need to be run as root; they only need access to the backing store and the `mount point`. Another difference is that processes, rather having a single user identity fixed at creation time, have identity tokens which are disjoint from the process, i.e. they may be added with the appropriate permission from an authority or destroyed.

Being familiar with the Unix environment is an imperative for feeling at ease in GNU. Having experience with the Debian tools will also prove invaluable to the configuration and maintenance of a GNU/Hurd box.

This guide endeavors to make installing the Hurd as painless a process as possible. If there are errors, they are most certainly the author's. Please report them, along with any other suggestions or criticisms, to him; all are gladly accepted.

2 Real Estate or Finding A Home

If you do not have an available partition or an extra hard drive, this can be the longest step. In this case, you will need to repartition the hard drive. One solution is to use GNU's partition editor, Parted. It features not only basic partition editing but also partition resizing and moving functionality. It can be found at http://www.gnu.org/software/parted. The manual is quite complete and includes several tutorials.

The Hurd can only support partition sizes of up to approximately two gigabytes; anything larger than this will not work. This limitation is due to a design decision that was made several years ago in which the filesystem server maps the entire filesystem into virtual memory. As the amount of virtual memory available on an ia32 is only four gigabytes of which Mach allocates three gigabytes to the application and, of that, a significant portion is reserved for the code, the stack and the heap, the final, maximum contiguous virtual memory area that remains is generally about two gigabytes. This limitation is scheduled to be removed.

[1] Philip Charles has created a set of CDs (available at http://www.debian.org/ports/hurd/hurd-cd) which contains a live Debian GNU/Linux system thereby arguably eliding this requirement, however, we maintain that this is functionally equivalent.

Having said that, a single two gigabyte filesystem is more than enough for a working system. Many, however, prefer at least two filesystems: a root filesystem and a second for '`/home`'. This latter scheme is highly advised for developers: compiling the Hurd can take up quite a bit of space.

The Hurd supports several extensions to the ext2fs filesystem format. Foremost among these are passive translators and a fourth set of permission bits for unknown users (users without an identity–not the other user). To use these extensions, the owner of the partition must be set to `hurd`. `mke2fs`, unless specifically overridden on the command line, will set the owner to whatever operating system it is running on. As the Hurd will diligently respect this setting, care must be taken to set this appropriately or the Hurd will fail in subtle ways. Be aware that even if a file system is owned by a particular operating system, others may still use it; they just may not be able to use certain extensions.

To create a filesystem, use `mke2fs` and pass it '`-o hurd`' to designate the Hurd as the owner of the new file system. For instance, assuming the parition is '`/dev/hda2`':

```
# mke2fs -o hurd /dev/hda2
```

3 The Boot Loader

Unlike GNU/Linux and the BSDs, the Hurd does not have its own boot loader; any boot loader that supports the multiboot standard can be used to load the Hurd. At the moment, there is only one project which that satisfies these requirements: Grub, the GRand Unified Boot loader.

A word about Grub. Unlike traditional boot loaders on the x86, such as LILO, Grub is very powerful. It has a command line interface, bootp, dummy terminal support and a plethora of other features. In addition, it can boot most any operating system. If you have ever booted an alpha or sparc, you will understand what Grub can do. Therefore, do not be scared: Grub is better. You will like it. You will not go back.

To find Grub, visit http://www.gnu.org/software/grub/. Here, there is a source tarball and a floppy image. If you choose to download the tarball, it is a normal `configure`, `make` and `make install`. Included is a wonderfully complete manual on how Grub works. Read it. If, on the other hand, you choose to download the floppy image, it is sufficient to dump it to a floppy disk to get a working Grub, for example:

```
# dd if=grub-boot-image of=/dev/fd0
```

You can always install Grub onto your hard drive at a later date.

4 Cross Install

The next step is to download the base system at: ftp://alpha.gnu.org/gnu/hurd/contrib/marcus/gn

The tarball is setup to extract everything into the current directory. After the filesystem is mounted, the archive can be extracted. Assuming that the filesystem is on '`/dev/hda2`', the mount point is '`/gnu`' and archive is in current user's home directory, the following is required:

```
# mount -t ext2 /dev/hda2 /gnu
# cd /gnu
# tar --same-owner -xvzpf ~/gnu-latest.tar.gz
```

5 Booting the Hurd

All is now in readiness to boot the Hurd for the first time. After verifying that the Grub boot disk is in the drive, reboot. If all goes well, either a Grub menu or command line will be displayed. If presented with a menu, press `c` to go to the command line.

First, GNU Mach needs to be loaded. This requires knowing the filesystem and the path to GNU Mach. Grub uses a partition nomenclature that is a bit different from both Linux and the Hurd: both IDE and SCSI drives are named '`(hdN,M)`'. `N` is the drive number (zero based) as enumerated by the BIOS. That is, Grub makes no distinction between IDE and SCSI disks. `M` identifies the partition on the drive. It is also zero based index. If this sounds confusing, relax: Grub is also helpful.

To determine on which filesystem a particular file resides, Grub provides the `find` command. When this command is issued along with a filename, Grub searches on each filesystem for the specified file and prints where it was found. For example, to search for the kernel, '`/boot/gnumach.gz`':

```
grub> find /boot/gnumach.gz
 (hd0,0)
```

Here, Grub is indicates that '`/boot/gnumach.gz`' is on '`(hd0,0)`'.

Before loading the kernel, at least one option, the root partition, must be specified on the command line. This will be used by the Hurd itself (i.e. not Grub). As such, it must be in terms that the Hurd can understand.

GNU Mach enumerates disks starting at zero. IDE drives are prefixed with `hd`, while SCSI disks are prefixed with `sd`. Like Linux, drives are number by their position on the controller. For instance, the primary master is `hd0` and the secondary slave is `hd3`. Partitions use the BSD slice naming convention and append `sM` to the drive name to indicate a given partition. Note that `M` is a one, not zero, based index. The slice number is simple to calculate: just increment what was used for Grub by one.

Since the Hurd has not yet been configured, it must be started in single user mode. Adding a '`-s`' to the kernel command line is all that is required.

Assuming that the first drive (i.e. '`(hd0)`') is the master on the secondary controller, we would have:

```
grub> kernel (hd0,0)/boot/gnumach.gz root=device:hd2s1 -s
 [Multiboot-elf, ...]
```

Next, the root filesystem server and the `exec` server must be loaded. This is done using Grub's boot module capability. The ${var} are filled in by GNU Mach. The arguments are used by the Hurd to indicate what type of information is being provided. Since the ext2fs command line is very long, it can be broken up by escaping the newline character in the normal Unix fashion. Be sure that there is not space after the slash at the end of each line. Also be sure to differentiate `{` and `}` from `(` and `)`.

```
grub> module (hd0,0)/hurd/ext2fs.static \
 --multiboot-command-line=${kernel-command-line} \
 --host-priv-port=${host-port} \
 --device-master-port=${device-port} \
 --exec-server-task=${exec-task} -T typed ${root} \
 $(task-create) $(task-resume)
```

```
  [Multiboot-module  0x1c4000, 0x2cfe6a bytes]
grub> module (hd0,0)/lib/ld.so.1 /hurd/exec $(exec-task=task-create)
  [Multiboot-module  0x494000, 0x27afe bytes]
```

Once the GNU Hurd is running, process can be automated by adding the appropriate commands to Grub's '`/boot/grub/menu.lst`' configuration file.

The GNU Hurd can be now booted:

```
grub> boot
```

Sit back and watch the messages. This is actually more important than most people believe: there is a bug in GNU Mach whereby hitting a key during the boot process causes the kernel to panic.

If the Hurd fails to boot, it could be due to shared IRQs: GNU Mach does not play well with these. You can verify your situation by looking at, for instance, the '`/proc/interrupts`' file under GNU/Linux. Also, as GNU Mach does not support loadable kernel modules, many of the drivers are compiled into the default kernel. If there are old peripherals, this can be a problem: a device may incorrectly respond to a probe intended for a completely unrelated device and thereby cause a crash. Building a new kernel with only the required device drivers will usually solve this problem. GNU Mach is easily cross compiled. If you are running Debian, try installing the '`gcc-i386-gnu`' package.

If this does not help, explore the resources listed at the end of this document. Finally, ask on the appropriate mailing list.

6 Native Install

Once you are presented with a shell prompt, and any time that the the Hurd is in single user mode, it is necessary to set the terminal type:

```
# export TERM=mach
```

Be warned that ⟨CONTROL-C⟩ and family will not work in single user mode.

We can now run the `native-install` script. This will configure the packages and set up several important translators:

```
# ./native-install
```

Before the script terminates, it will indicate that it needs to be run a second time. Follow its instructions and reboot using the `reboot` command. Again, go into single user mode and run `./native-install`.

7 Configuration

7.1 The Network

To configure the network, the pfinet translator must be configured. This is done using the `settrans` command to attach a translator to a given filesystem node. When programs access the node by, for example sending an RPC, the operating system will transparently start the server to handle the request.

```
# settrans -fgap /servers/socket/2 /hurd/pfinet -i eth0 \
  -a a.b.c.d -g e.f.g.h -m i.j.k.l
```

Here, `settrans` is passed several options. The first two, '`fg`', force any existing translator to go away. The next two, '`ap`', make both active and passive translators. By making the translator active, we will immediately see any error messages on '`stderr`'. The latter, saves the translator and arguments in the node so it can be transparently restarted later (i.e. making the setting persistent across reboots). The options are followed by the node to which the translator is to be attached, then the program (i.e. translator) to run and any arguments to give it. The '`-i`' option is the interface `pfinet` will listen on, '`-a`' is the ip address, '`-g`' is the gateway and '`-m`' is the network mask.

Be sure to add name servers to your '`/etc/resolv.conf`' file:

```
nameserver 192.168.1.1
```

To test the configuration, `ping -c2 gateway`. The '`-c`' is important to limit the number of pings; recall, ⟨CONTROL-C⟩ does not work in single user mode.

DHCP does not yet work on the Hurd. This is due to limitations of pfinet: it is based on the Linux' TCP/IP code and unable to listen on '`0.0.0.0`'.

Help on `settrans` can be obtained by passing it the '`--help`' option. Help on a specific translator can be gotten by invoking it from the command line with the same argument, e.g.:

```
# /hurd/pfinet --help
```

As there can be a lot of output, consider piping this through a pager such as `less`.

7.2 Other File Systems

Next, edit '`/etc/fstab`' to add any additional filesystems as well as swap space. It is *very important* that swap space be used; the Hurd will be an order of magnitude more stable. Note that the Hurd can transparently share a swap partition with Linux but will happily page to any device including a raw partition such as your home partition. By default, `nano` is the only editor installed by the the base distribution.

Here is an example '`/etc/fstab`' file:

```
# <file system> <mount point>   <type>  <options>  <dump>  <pass>
/dev/hd2s1      /               ext2    rw         0       1
/dev/hd2s2      /home           ext2    rw         0       2
/dev/hd2s3      none            swap    sw         0       0
```

Remember to create any devices using the `MAKEDEV` command:

```
# cd /dev
# ./MAKEDEV hd2s1 hd2s2 hd2s3
```

To mount an nfs filesystem, `/hurd/nfs` translator is used. When run as non-root, the translator will connect to the server using a port above 1023. By default, GNU/Linux will reject this. To tell GNU/Linux to accept connections originating from a non-reserved port, add the '`insecure`' option to the export line. Here is an example '`/etc/exports`' file assuming the client's ip address is '`192.168.1.2`':

```
/home  192.168.1.2(rw,insecure)
```

To mount this from a Hurd box and assuming that nfs server's ip address is '`192.168.1.1`':

```
# settrans -cgap /mount/point /hurd/nfs 192.168.1.1:/home
```

7.3 Rebooting

Finally, reboot into multiuser mode, i.e. in the same way single user mode was brought up minus the '`-s`' option when loading the kernel. For details, refer to See Chapter 5 [Booting the Hurd], page 3.

Happy Hacking!

8 Final Words

8.1 Documentation

To understand the Hurd, start with the home page on Debian's site: `http://www.debian.org/ports/hurd/` and GNU's site: `http://hurd.gnu.org`.

Also consider reading the source code and writing more documentation.

8.2 The Grub Menu

Having to always load the kernel by hand can be very tedious. Edit the '`/boot/grub/menu.lst`' and tailor it appropriately; booting will become much quicker and easier.

8.3 Adding Devices

By default, only a few devices are created in the '`/dev`' directory. Use the `MAKEDEV` script to create any needed device nodes.

8.4 Installing More Packages

There are several ways to add packages. Downloading and using `dpkg -i` works but is very inconvenient. The easiest method is to use `apt-get`. Edit '`/etc/apt/sources.list`' and add the following two entries:

```
deb ftp://alpha.gnu.org/gnu/hurd/debian unstable main
deb ftp://ftp.debian.org/debian unstable main
```

`ftp://alpha.gnu.org` contains packages that have hacks or patches that have not yet been integrated upstream. There are no mirror sites. To use a local Debian mirror, visit `http://www.debian.org/distrib/ftplist`.

If GNU Mach does not recognize your network card or you use a modem, the only way to upgrade will be to download the packages and then transfer them to the GNU Hurd system. The easiest way to do this is to use apt off-line. Refer to '/usr/share/doc/apt/offline' for detailed instructions.

8.5 XFree86

XFree86 has been ported and all video cards, which it supports that do not require a kernel module should work.

First, set up the keyboard translator:

```
# cd /dev
# ./MAKEDEV kbd
```

And then the mouse translator. For a serial port mouse, run the following replace 'com0' with the appropriate communication port:

```
# settrans /dev/mouse /hurd/mouse --device=com0 --protocol=microsoft
```

Make sure that '/dev/com0' actually exists. If it does not, create it using MAKEDEV in the usual fashion.

PS/2 so not require a device node. It is simple a matter of:

```
# settrans /dev/mouse /hurd/mouse --protocol=ps/2
```

Other mice can be used; run '/hurd/mouse' with the '--help' option for details.

You will need several X packages. x-window-system-core, rxvt and twm or fvwm are a good start.

Debconf can be used to configure XFree86, however, it is not Hurd aware and the configuration file will need to be tweaked. Change the pointer section to read:

```
Section "Pointer"
  Protocol "osmouse"
  Device "/dev/mouse"
EndSection
```

'Emulate3Buttons' may be optionally added. Nothing else will work.

The GNU Hurd does not use ld.so.conf. Since '/X11R6/lib' is not in the default library search path, it is necessary to add the following to either '/etc/profile' or each user's '.profile':

```
export LD_LIBRARY_PATH=/X11R6/lib:$LD_LIBRARY_PATH
```

Finally, run startx.

There are several caveats to be aware of. xterm does not work correctly as it is SETGID (and thus ignores LD_LIBRARY_PATH and fails to load the appropriate shared libraries); try rxvt. update-menu has not yet been ported. As such, there are no fine Debian menus. Although a pthreads implementation exists not all pthread packages have been ported: do not expect Gnome and KDE to work.

8.6 Virtual Terminals

The Hurd does not have virtual terminals although it is in development. Use the screen package in the interim.

8.7 Mailing Lists

1. `debian-hurd@debian.org` Discussion on the Hurd as it relates to Debian.
    1. Archive: `http://lists.debian.org/#debian-hurd`
2. `web-hurd@gnu.org` Development of the Hurd web pages at `http://hurd.gnu.org`.
    1. Archive: `http://mail.gnu.org/pipermail/web-hurd/`
3. `help-hurd@gnu.org` Help on the Hurd in general.
    1. Archive: `http://mail.gnu.org/pipermail/help-hurd/`
4. `bug-hurd@gnu.org` Bug reports and general development. Send patches here.
    1. Archive: `http://mail.gnu.org/pipermail/bug-hurd/`

8.8 Other Resources

The Hurd Wiki, `http://hurd.gnufans.org/bin/view/Hurd/WebHome`, answers commonly addressed problems that new users face.

9 Works Cited

"The Easy Guide to Installing Hurd on a Linux Box" Copyright © 1999 Matthew Vernon `matthew@debian.org`. `http://www.pick.ucam.org/~mcv21/hurd.html`

Table of Contents