

The Utah PA-RISC Code Snapshot

Contents

- Important:** Last Minute Changes
- What it is (and is not)
- Supported Hardware
- Known Bugs
- FTP Information
- About the Pieces
- Future Plans
- Acknowledgments

More Details

- Installation
- FAQ
- PA Code Tour
- List of Installed Binaries

Last Minute Changes

Complete source for the Lites server is not included in this snapshot. We had planned to release this snapshot at the same time as Johannes' initial Lites release, but it will not be released until later this month. We do include a binary of a pre-release of the Lites server and emulator in the disk image as well as source for the PA specific parts (for reference). The initial release of Lites will have integrated PA support.

We are using a pre-release of Lites. Thus, many or all of the "Lites bugs" listed below may be fixed in the official Lites release. You *should not* infer anything about the quality of Lites based on this version. You *should* pick up and use the official version of Lites when it is released. We thank Johannes for letting us release his "snapshot" code.

This current snapshot won't work on a 712. There appear to be a host of annoying problems. The bootloader doesn't work on an external SCSI-1 drive, only an internal disk. Either the boot ROM doesn't load it or loads it someplace we are not expecting. In any case, the bootloader never gets control. This may be hard to find/fix and is not high priority. If you boot from an internal disk, the keyboard doesn't work. This is no doubt a simple configuration problem in our kernel and will be fixed soon. Finally, even if you use the serial port as console and boot single user, the machine hangs doing fsck. This is something else we broke late in the game as it used to work fine. This will be fixed soon.

The snapshot was tested on only a small subset of the supported machines. We successfully booted

to multi-user on a snake cpu (730), a 7100 (715/75), and a 7150 (735/125).

What it is (and is not)

Integrated support for the PA-RISC 1.1 (aka HP 700) platform, in these components: freely distributable source, binaries, and boot image for a complete Mach kernel that includes some of our recent research (though not exploited), the “Lites” BSD-based single-server, include directories and libraries, and a complete GNU tool chain for the ELF object format. In addition, there are assorted other PA-related device drivers, kernel components and utilities which could be useful. A fairly complete 4.4-lite user environment is also provided. The entire system was self-built on mach4/Lites.

We refer to this as a *snapshot* and not a *release* as the provided code is not robust, does not provide a complete application environment, contains minimal bootstrap capability, and has practically no documentation. Hence, **we recommend that you do not run this, unless you are a hard-core O.S. hacker who has an interest in, and knowledge of, the PA architecture.** Later true releases, with help from you O.S. hackers, will allow the more faint-at-heart to run an alternative O.S. on their HP workstations.

This snapshot was facilitated by HP when they recently “freed-up” a number of essential files. Thanks HP!

Supported Hardware

CPUs

original: 705, 710, 720, 730, 750

PA7100: 715/33, 715/50, 715/75, 725/50, 725/75, 735/99, 755/99

PA7100LC: 712/60, 712/80, 715/64, 715/80, 715/100

PA7150: 735/125

Displays

GRX, CRX

Peripherals

internal SE and FW SCSI (disks and tapes)

serial ports, HIL, PS/2 keyboard and mouse, builtin 802.3 LAN

Notably unsupported are: builtin FDDI, EISA, parallel port, audio, teleshare port, and PS/2 floppy.

Known Bugs

These are bugs which we have encountered while preparing this release. A number of these may be related to our peculiar environment (a large number of machines running non-vendor supplied software) and you may or may not see them. In no particular order:

- The PA boot loader is not well integrated with Mach right now. In particular, you can only pass flags to the kernel, you cannot pass flags on to the server.
- The kernel will run out of activations under a heavy load. Might be a leak.

- RPC (i.e. two-way IPC) is even slower than in Mach 3.0. This kernel is optimized for migrating RPC and the old, hand-crafted fast IPC paths in `mach_msg` are gone. Unfortunately, nothing yet uses migrating RPC so everything uses the slow, general IPC path.
- The included migrating threads code will not work for Lites anyway as it does not maintain sequence numbers in messages.
- There appears to be a paging space leak in Lites. You will likely eventually run out of paging space and panic. This is believed to be fixed in the current version of Lites.
- Writes across NFS fail (always return an error). Curiously, they fail as either a client (Lites access to another machine) or server (other machine access to Lites). Possibly a by-product of our NFS environment.
- We have encountered a variety of problems with 4.4's COMPAT_43 code when running our old BSD binaries. These include the console not working after `getty` gets an interrupted system call return, raw tty handling not working correctly in `emacs`, and X11 clients getting X protocol errors. None of these affect the native 4.4 ELF binaries.
- There is at least one bug in the native build environment which may require that you hand build pieces: there is a parsing problem in the assembler having to do with use of “!” as a line separator in hand-written assembly. You must use:

```
"AS=tr \! '\012' | as"
```

to build these. The makefile for `libc` takes care of this, but you may encounter it elsewhere. You may also see “ignoring subspace decl...” warnings from the assembler, ignore these. The ELF linker is known to core dump when linking code with undefined symbols. Also, there is no ELF `libc nlist` routine,

FTP Information

These files are all in the anonymous ftp “flexmach” directory on `jaguar.cs.utah.edu`. If you are using Mosaic to transfer them, you will want to give the “Load to Local Disk” option before clicking on them, or they will expand mightily. The installation instructions describe how to use these files.

Kernel source (3 MB, compressed *tar* file)

Sources for the Mach kernel, Lites server, and standalone bootstrap.

User source (36 MB, compressed *tar* file)

Sources for a scaled-down 4.4-lite environment (i.e. everything else). Should include source for everything that is in the disk image and `/usr` binary *tar* file.

User binaries (23 MB, compressed *tar* file)

A scaled down 4.4-lite `/usr` filesystem.

Disk image (8 MB, compressed *dd* file)

A raw disk image of a bootblock, root and swap partitions.

WEB docs in postscript (40K, compressed *tar* file)

These WEB PA-related documents in postscript form.

WEB docs in ASCII (20K, compressed *tar* file)

These WEB PA-related documents in plain ASCII form.

About the Pieces

The Bootloader

The bootloader is an implementation of the standard (if there is such a thing) BSD “boot program” traditionally found in the `/sys/stand` directories. The PA version was originally written by Mt. Xinu and heavily modified at Utah since. The original AT&T encumbered routines have been replaced by those from the Mach distribution. It is a single-level bootstrap, i.e. it is loaded by the ROM and is directly responsible for loading the kernel, and uses PDC (aka “the boot ROM”) routines for all console, disk and tape operations. It can load either SOM or ELF format kernels. By default it attempts to load `/mach` and, failing that, `/vmunix`.

It is completely incompatible with the HP-UX bootstrap, and totally unrelated to the x86 bootstrap.

See the bootstrap section of the code tour for more details.

The Kernel

The Mach kernel source is derived from CMU’s MK83 release and is what we loosely refer to as Mach 4. It contains some initial work we have done as part of the ARPA-funded Fast and Flexible Mach Systems work. In particular it contains a prototype implementation of migrating threads and a basic framework for signature-based RPC, a fundamental component of the presentation/interface work. Currently, none of these features is used either by the Lites server or within the kernel itself.

The PA specific portion includes all the necessary interrupt, exception, and system call handling code (“locore”), a pmap module, and device drivers for the VSC and GSC bus based workstations. The only kernel or server component which is provided only as a binary library and not in source form is the so-called “floating point emulation” code which handles operations and exceptional conditions not done in hardware. It is similar in function to Motorola’s FPSP package for the 68040. The source is HP-proprietary, and the distributed “spmath” library cannot be used for commercial purposes.

The libmach and libcthreads libraries, which are conventionally considered part of the kernel, are also included with the necessary changes for PA support.

We also include some additional code which has not yet been integrated into Mach4. The code is either part of our earlier HP 800 series (PA-RISC 1.0) Mach or HP 700 OSF/1 ports or it is part of our 4.3/4.4 hybrid BSD system. The former includes bus configuration and rudimentary device drivers for the CIO bus based workstations as well as a remote kernel/task debugging facility developed by Convex. The latter includes some basic EISA support and alternative LAN drivers.

Finally, the kernel source tree has been reorganized in a new build environment which uses the FSF’s configure and make utilities.

The Server

The operating system “personality” is provided by the Lites server/emulator. Lites is a user-mode, single-task implementation of BSD Unix which runs on top of a Mach micro-kernel. It is being

developed by Johannes Helander at Helsinki University of Technology (HUT) in Finland. Lites is based on the 4.4-lite release from Berkeley, single-server code from CMU, machine dependent ports from various people, and a lot of code and work from Johannes. It is freely available and runs on the i386, pc532, Mips and PA-RISC. The PA port was done at Utah. See the Lites home page for more details.

On the PA, Lites can exist on either a Utah BSD binary base (SOM format files) or a true 4.4bsd binary base (ELF format). This means that it can not only execute binaries in those formats, but it also understands their filesystem formats and layouts. The ability to execute HP-UX binaries is not implemented in this snapshot, but will be in the future.

Compiler and Other GNU Tools

We include a GNU compiler toolchain that generates PA ELF binaries, not the usual HP SOM object format. The ELF linker is currently very slow; the tradeoffs are explained in the FAQ. GNU Tools provided include:

- binutils: version 2.5.2.u2 (includes ar, gas, ld, nm, ranlib, size, strings, and strip)
- gcc: version 2.6.3.u4
- make: version 3.69

where a “.u?” suffix reflects Utah enhanced/fixed versions. For details on Utah modifications, see the file “UtahChangeLog” included with the tools. **The language tools we distribute separately on jaguar are not suitable for use with this snapshot. We expect to remedy this situation within the next month.** Other useful GNU tools included in this snapshot:

- grep: version 2.0
- shellutils: version 1.9.4
- tar: version 1.11.2
- textutils: version 1.11

Note that there is currently no emacs.

Other Code

The application base included for this snapshot is largely taken from the 4.4-lite release. We include complete sources for /sbin, /bin, a large number of libraries, and a smattering of other applications required for a “self-hosting” environment.

PA specific code is confined to some include files and routines in the C and math libraries. The sys/parisc/include directory includes some 4.4-ish kernel header files that are required by the various 4.4 applications. The lib/libc/parisc subdirectory contains assembly code for syscall traps, fast string and memory routines, and compiler runtime support routines (“millicode”). The lib/libm/parisc subdirectory contains assembly code for a few basic operations which are efficiently supported by the hardware co-processor.

Future Plans

Short term plans for the PA code are to continue our research efforts, and to make some progress on robustness and performance. It would be nice to have something suitable for desktop usage by the spring. Beyond that, the future of the PA port is unclear. HP is no longer funding us and ARPA does not have a great deal of interest in the PA as a research platform. As long as the PA machines continue to be our main computing resource we will likely develop on and support them, but unless the funding situation changes, the PA will likely be dropped at some point.

For more on our long term Mach plans, refer to the Fast and Flexible Mach-based Systems page.

Acknowledgments

Utah has been involved with the PA for the last 5 years. During that time a number of people have been involved with BSD and Mach. Here are a few of those: Bryan Ford, Jeff Forys, Mike Hibler, Jeff Law, Jay Lepreau, Doug Orr, Donn Seeley, Leigh Stoller, and Bob Wheeler.

A number of universities, research labs, and companies have used our BSD and Mach over the years and given us valuable feedback. Among those are: U. of Arizona, U. of California Berkeley, U. of Wisconsin, U. of Colorado, Convex, LBL, Mt. Xinu, OSF, and Stratus.

Without the Free Software Foundation and their many software packages, we would not have nearly as complete an environment as we do. Thanks especially for GCC and the other compiler related tools.

And of course, most importantly, thanks to Hewlett-Packard not only for freeing up a lot of essential code but for supporting our PA work over the years.

More Details

[UP to “flexmach” project overview]

Mike Hibler <mike@cs.utah.edu>