Porting Linux to PA-RISC, The Final Architecture

by Alex deVries

The Puffin Group/Linuxcare
41 1/2 William Street
Ottawa, Ontario, Canada
K1N 6Z9
Phone: (613) 562-2759
Fax: (613) 562-9304
alex@thepuffingroup.com

1. Introduction

It's rare that individuals and different corporations can work together to have just one goal, but the PA-RISC Linux project appears to have achieved this.

The goal is simple: to create an entire Linux distribution with kernel running on all HP PA-RISC hardware, and now.

This paper intends to explain the challenges of this project by describing the various PA-RISC hardware in existance, and the complicated situation of PA-RISC build tools.

2. An Overview of HP Hardware Architecture

Understanding how Linux on PA-RISC is to be implemented requires some knowledge of the components and architecture of the systems. The following is an overview of these components.

2.1 Sources of Information

All information in this paper must be available from publically available sources. In order to prove that the information herein is data that could be collected by anyone, special effort has been made to document that this information is available from various publically accessible sources.

The first and most significant sources of information are those provided publically by HP for the purposes of Linux development. These can be found at http://www.thepuffingroup.com/parisc/documentation.html.

Firmware and userland tools used to query firmware can offer quite a bit of information. The HP-UX tool *ioscan* queries firmware, then displays version information for all the onboard components that exist.

HP released the Appendix B of the document titled PA-RISC 1.1 I/O Firmware Architecture Reference Specification publically. From this PDF, a data file was used which is used to correlate the hardware version numbers to actual textual descriptions.

In order to have an understanding of a large number of machines in existance, the PA-RISC develoment group had to rely on the contributions of many different PA-RISC computer owners. They were asked to provide their ioscan output, which is then catalogued at http://www.thepuffingroup.com/parisc/hw.html.

2.2 Groups of PA-RISC Machines

There are about 130 different PA-RISC models in use today, being able to group them allows for easier analysis of the tracking of the development of these machines. These groupings are useful only for the purposes of calculating support for PA-RISC Linux.

2.2.1 Ancient Machines

Some older machines are simply too old to consider porting to: early models which contain PA1.0 CPUs, which are too difficult, and offer no significantly large install base to consider a priority.

As well, many industrial machines have HP-PB (NIO) based IO subsystems; without documentation on these components, it really becomes quite difficult to offer support. It isn't unrealistic to expect that support for older systems will be included one day.

The HP3000 line, intended to run MPE/iX are VME based. There is actually VME support in other Linux architectures, but it's unlikely that support for these will happen soon due to limited availability to developers.

All of these systems have the Gecko System Connection (GSC) bus. GSC isn't implemented in any other architectures but PA-RISC, so it follows that there will be no Linux drivers written yet.

One of these is the CIO controller chips, for which there is no publically available documentation. Also, developers have access to more modern hardware, so this set of machines just isn't receiving their attention. It's unlikely that there will be support for these machines soon.

2.2.2 Modern PA RISC 1.1 Machines

This is the category for which there is the most support for PA-RISC Linux.

There are many 32 bit PA 1.1 based machines around. In particular, the 712 Gecko desktop systems have made ideal machines for getting started. There are plenty available, are older enough to be low cost, and have a very simple architecture.

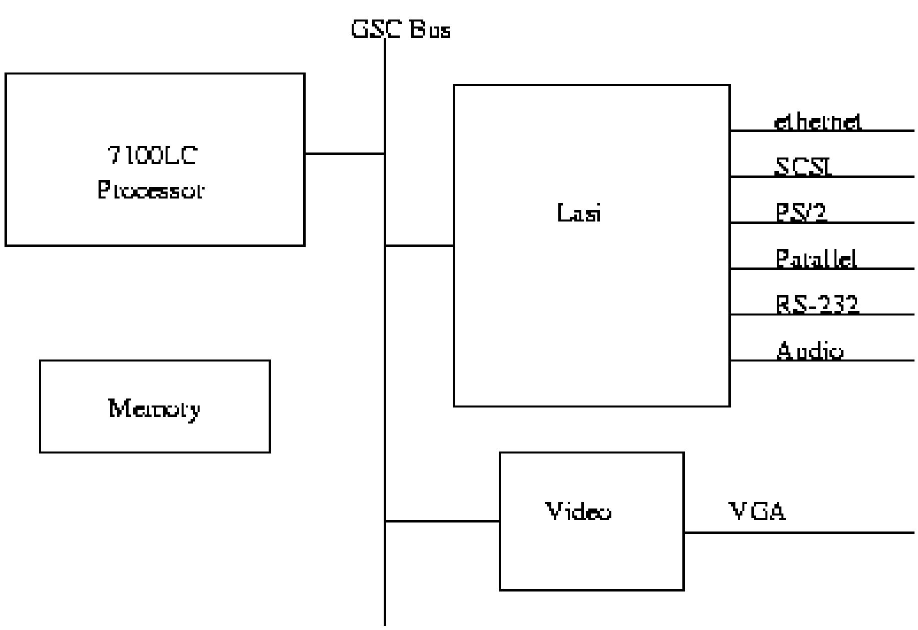

Figure 1: 712 Architecture

As shown in Figure 1, a 712 system is not much more than a PA 1.1 CPU with GSC bus, a Lasi IO controller for SCSI, floppy, serial, audio and ethernet. This machine is intended to be a desktop machine, so also contains a VGA graphics controller.

The 715 is similiar in design, but is slightly older and has the IO controller functionality spread among multiple chips in older implementations. Newer 715s are virtually identical in I/O architecture to the 712.

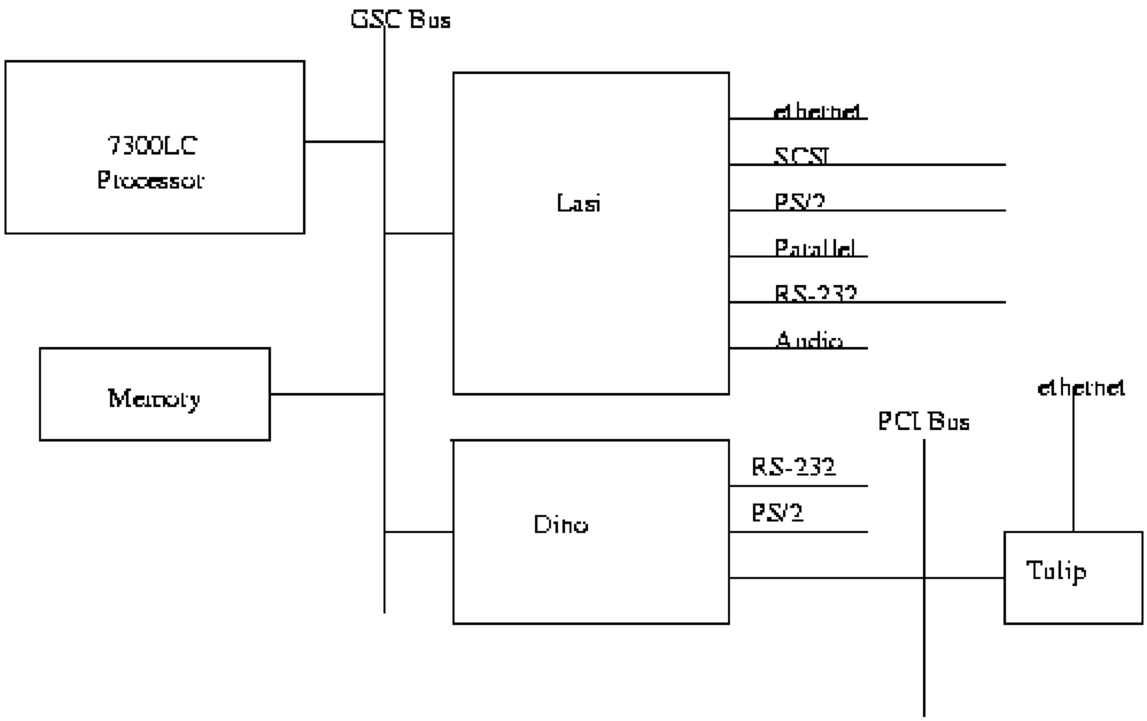

Figure 2: A180 architecture.

The first systems that HP have loaned to developers are A180s. It has clear lineage from the 712, although this is a server based system. It contains Lasi, although it only implements portions of it such as serial and SCSI. It offers PCI bus access, and uses a Dino GSC to PCI bridge. There are two PCI slots available for external cards, and there is an onboard 10/100 ethernet PCI controller, the infamous Intel 21143 forwhich there exists a solid device driver. This is shown in Figure 2.

All of these systems implement legacy firmware.

2.2.3. Early PA-RISC 2.0 Systems

This generation of systems combines early PA2.0 processors and 32-bit GSC I/O components (including users' add-in cards, GSC token ring card, anyone?). PA-RISC 2.0 CPUs have many performance advantages, although they implement a 40-bit Runway bus instead of GSC. This explains the architecture as shown in Figure 3. Here, either a U2 chip implement a Runway to GSC bridge.

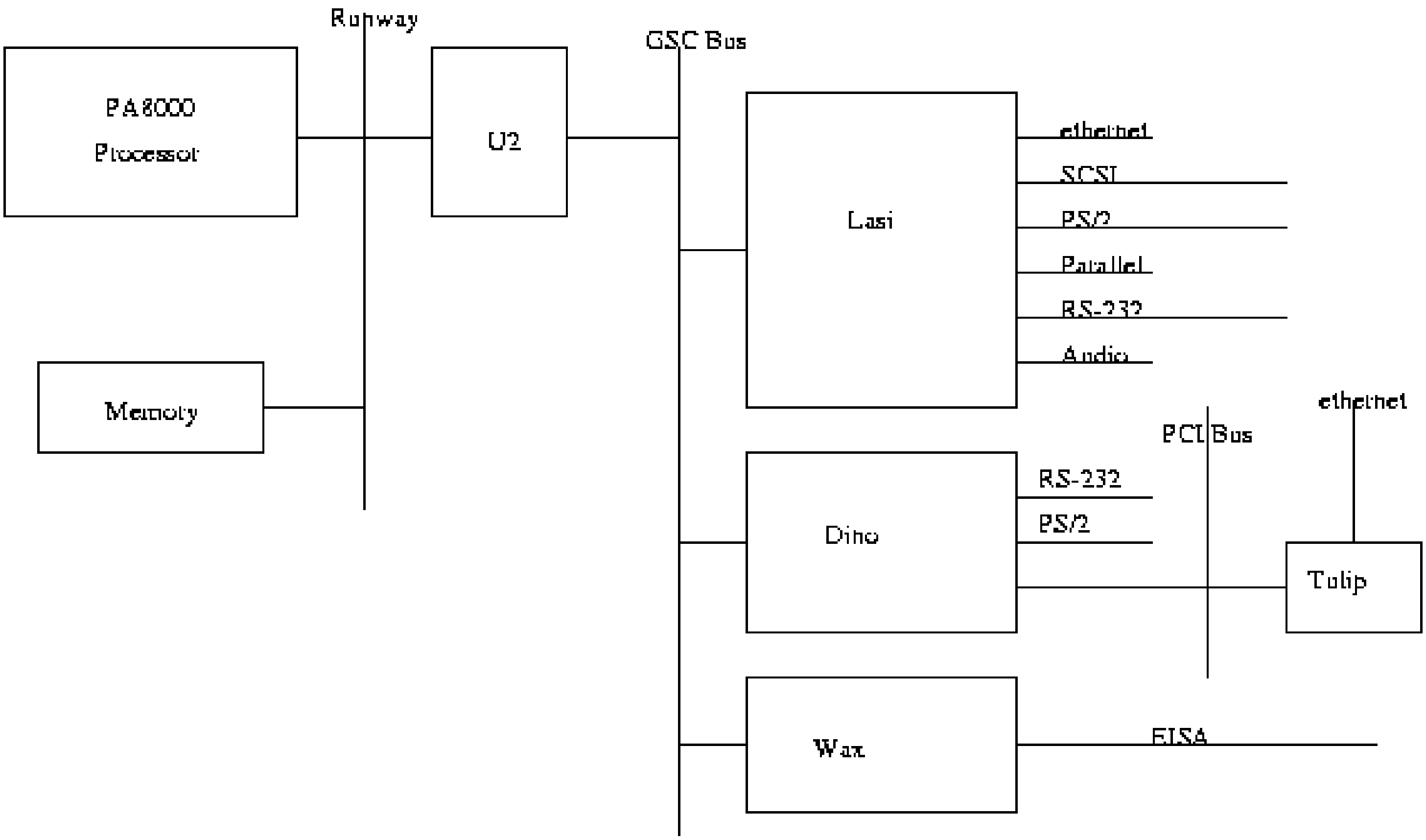

Figure 3: The architecture of the D270.

Some of these systems contain GSC chips such as Lasi and Dino to implement their IO interfaces, such as the D270 or the C360.

HP has not yet released any technical documentation for these chips, so one would think that it would be impossible to offer support for machines with U2 and Uturn in them.

These chips appear to be passive components. Once they're setup to remap certain portions of IO space and handle interrupts, they simply work. This is actually done by firmware on these machines, so there is virtually no development work required by the Linux kernel.

All of these machines implement legacy PDC.

Although the first PA 2.0 machine to boot to a shell prompt was a C360, no work has been done yet on the IO architecture.

2.2.4 Modern PA-RISC 2.0 Systems

New PA2.0 systems themselves have several commonalities amongst each other, ranging from the N class, C3000, J5000 to L2000. Although different perhaps in a mechanical and marketting perspective, these machines all contain common IO architecture.

These machines are all PCI based, but the actual IO architecture is not yet public. However, it is known to contain an LBA to connect the runway bus to PCI. More detail may be surmised by reading the hardware pages, which summarize ioscan output.

All IO components are PCI based on these systems, and common IO components include Symbios 53cXXX SCSI controller chips and Tulip 10/100 ethernet controllers. The good news for Linux on PA-RISC is that all of these components exist in other architectures (particularly i386), so it's possible to leverage most of this work.

One thing that divides these machines is their implementation of firmware. In order to better control a 64-bit system, HP implemented 64-bit firmware. Support for this is still being worked on, which is stopping support for the L class and theoretically the N-class.

2.2.5 NUMA Architecture

The HP V-class implements the NUMA (Non-Uniform Memory Access) memory architecture, which Linux has never run on. Technically, this is a difficult task, and it doesn't make much sense yet from a practical perspective, since this is a market where HP-UX performs well.

As well, the current power requirements and cost of a V-class makes it a challenge to have one sitting in one's basement.

2.3 Processors

There are really only three categories of processors.

PA 1.0
These CPUs are only found in very early machines. They are 32-bit only, and have a GSC bus as their interface.

PA 1.1 (7000, 7100, 7200 and 7300)
This is the most common group of CPUs found, and are 32-bit only and all have GSC busses.

PA 2.0 (8000, 8200, 8500, 8600 and more)
These are 64-bit capable processors, although initial support of Linux will be 32-bit based. The bus on these processors are 40-bit Runway busses.

2.4 I/O Components

2.4.1 Lasi

Lasi is a GSC IO controller that contains:

- an 82596 10Mb/s ethernet controller
- a 53c710 SCSI controller
- a standard parallel port
- two PS/2 ports
- a standard 16550A serial controller
- a custom sound controller (Crystal)

Lasi appears to have had it's roots with a set of chips called ASP (the expansion of this acronym is unclear).

It's used in just about all of the machines in the Modern PA-RISC 1.1 Machines group. It also makes it's furry face known in Early PA-RISC 2.0 Machines in conjunction with the U2 Runway to GSC bus adapter.

Looking at the motherboard of a 712, this one-chip solution makes it very simple to create a very small computer. All a 712 really is is a 7100LC CPU, a Lasi and some connectors.

Some machines include a Lasi, but don't actually include the connectors for all the components. The A180c, for instance, is a machine targetted towards the server market. It wouldn't make sense to include sound support. So that functionality exists on the A180's Lasi, but it isn't connected. Similiarly, the A180 needed 100Mb/s ethernet support, so the 10Mb/s ethernet controller isn't used.

Lasis are used also in add-on cards for roles such as second SCSI controllers or ethernet controllers. Sparc developers will be reminded of the Sun Happy Meal combo add-on card.

Linux support for Lasi is reasonably complete, since most of the devices in this macrocell already have support from different Linux architecture ports. HP has released full documentation for this chip.

2.4.2 Dino

Dino is the only GSC to PCI bus bridge that HP uses, which simplifies drivers for Early PA-RISC 1.1 Machines considerably. It also makes token appearances in Early PA-RISC 2.0 Machines in conjuntion with U2.

This is a relatively simple chip found on motherboards of systems with PCI busses. As well, it's used in add-on cards where it made sense to leverage an existing PCI chipset. As an example, HP ships GSC add-on gigabit ethernet cards. These are on a card with a GSC connector, which lead directly into a Dino controller. This then creates an on-card PCI bus, which contains an Alteon PCI controller chip. With the plethora of PCI chips available, this makes it very simple to create just about any GSC add-on card.

Linux support for Dino PCI bus controllers is complete. HP should be credited with supplying documentation to further this effort.

2.4.3 Wax

Wax is GSC to EISA controller chip that's used to support EISA chipsets and add-on cards. It has generally only found it's way in Modern PA-RISC 1.1 Machines (such as 715s), Early PA-RISC 2.0 Machines (with U2 to provide a GSC) (such as the D270) and some add-on cards.

There's no Linux support for these chips because there is no access to documentation. As well, EISA simply isn't yet a priority in terms of device driver support.

2.4.4 U2

U2 is bridge or bus that connects to a PA2.0 processor's Runway bus and controlls a GSC bus.

This allows the machine in question to use IO controller chips such as Lasi, Dino and Wax which require a GSC bus.

The existance of U2 can only be concluded because of the discovery using ioscan tools. It exists only in machines that have a PA2.0 processor and GSC components such as Lasi and Dino.

HP hasn't released documentation on this component.

2.4.5 Lower Bus Adapter

The LBA is found in Modern PA-RISC 2.0 Machines such as the C3000, N class, J5000 and others. It offers a PCI bus on PA2.0 systems. It is often used in multiples, in order to handle more than one PCI bus.

HP has not released documentation on this component.

3. Development Tools

Porting Linux to a new architecture always starts with creation or discovery of development tools.

In the spring of 1999, the team investigated the existing open source tools that had been used by the OpenBSD group. They created 32 bit ELF files, which weren't quite complete. The author of those tools, Michael Shalayeff, suggested we abandon those tools. The only reasonable alternative were the build tools included with HP-UX. Although these proved to be functional, they were only available to be hosted on HP-UX of course.

Hosting a Linux build environment on a commercial OS is not the way that the Linux development community is used to working, and requires that to be functional, all developers must have two machines: one to run HP-UX on to host the build environment, and the other to boot (and reboot and reboot) test kernels on. While not a large problem for developers within corporations such as HP and The Puffin Group who have access to many different machines, this isn't the case with the lone basement hacker, an important part of this development project.

The final product must include open source development tools.

3.1 SOM vs. ELF

In the beginning, there was SOM, the Spectrum Object Model format. This execution format was created in the early days of PA-RISC, and is the only format for 32 bit HP-UX binaries. It is unlikely any other object model supported by Linux on other architectures.

However, using SOM for Linux on PA-RISC holds two key advantages: binary compatibility and accelerated development. Binary compatibility will be discussed in Section 3.5.

3.2 SOM

The tools being used to create 32 bit SOM kernels and userland tools are:

- the GNU compiler (gcc) and assembler (gas)
- HP's linker
- other GNU tools (such as make, awk, sed, etc)

In December of 1999, HP released the source of the HP SOM linker in order to allow for hosting other platforms. Work is being done by the open source development community to fix the linker to work on other platforms.

It is clear that SOM is really just going to be used for binary compatibility and as an intermediary file format until ELF formats can be generated.

3.3 64-Bit ELF

PA-RISC 2.0 processors are 64 bit capable, so some new binary format was required. HP chose 64 bit ELF for the 64 bit versions of HP-UX (starting with version 11.0), which is perfectly in line with other ports of Linux to various architectures.

Until January, 2000, there was no publically available 64-bit PA-RISC compiler. The only way to create a 64-bit kernel would have been to retrofit the Linux kernel to build with HP-UX's commercial compiler, which would have been at considerable cost to the Linux development community.

However, HP sponsored work to be done on this aspect of GNU build system which enables there to be Linux on 64-bit PA-RISC architectures. Work is being done to create 64-bit PA-RISC kernels using these development tools.

3.4 32-Bit ELF

Ultimately, in order to make a successful Linux distribution, there will have to be a 32 bit ELF file format. This can be achieved by retrofitting the open source development tools used in 64 bit ELF for use with 32. First, one would modify the GNU binutils' BFD (Binary File Descriptor), make some minory changes to GCC and then to the GNU C library, glibc. It's likely that some kernel modifications and bootloader changes are required. This is left as an exercise to the reader.

3.5 Binary Compatibility

The current plan is for there to exist binary compatibility for both ELF64 and SOM HP-UX. In order to take advantage of binary compatibility, the first goal of Linux on PA-RISC was to be able to use this format. This would allow

By being able to run HP-UX executables, the PA-RISC developers are able to create userland programs quicker. They no longer need to be concerned with having access to a C library to link against: they can use the C library from HP-UX. The Puffin Group is working with HP to distribute HP's libc in order to accelerate this portion of development.

4. Current Status

As of mid January, 2000, considerable progress has been made towards the goal of having Linux run on different PA-RISC hardware.

HP had open sourced the SOM linker, and released a proper version of 64-bit GNU build tools. Some development has started to get a proper set of cross compiling tools working.

Various developments were achieved in the kernel itself: a 32-bit kernel now boots on both PA2.0 and PA1.1 powered computers. Memory management and core kernel functions have been implemented. On each of these, various I/O components are detected, including Dino and Lasi. Some work has started on PA2.0 I/O components as well. Preliminary work has begun on network devices built into these systems. The kernel now boots on most machines tested, and runs a small shell that's in a ramdisk.

5. Contributors to Linux on PA-RISC

The significant contributions to the Linux on PA-RISC have come from different sources.

Linux development community contributions include:

- Thomas Bogendoerfer
- Alan Cox
- Helge Deller
- Philipp Rumpf

Hewlett-Packard has contributed in at least the following ways:

- open sourcing certain build tools: 64-bit GCC, SOM linker
- hardware for both corporate developers and the Linux development community
- releasing documentation on hardware
- network infrastructure for CVS, web, e-mail and FTP servers
- internal development contributions (Grant Grundler, Paul Bame, Scott Holbrooke and John Marvin)
- access to various engineering staff who understand the hardware
- marketing and public relations (Mike Balma)

The Puffin Group contributed:

- open source project management (Alex deVries)
- various development components
- internal development contributions (Michael Ang, Christopher Beard, Alex deVries, David Kennedy, Philip I. Schwan, Matthew Wilcox)