PA-RISC/Linux Boot HOWTO

Deb Richardson
The Puffin Group

deb@thepuffingroup.com

This document outlines the procedure for getting the current PA-RISC/Linux development kernel to boot on your PA-RISC system.

This HOWTO does not currently include the information needed for booting your PA-RISC/Linux system from disk. The booting from disk procedures will be added as soon as they are available. For more information about the PA-RISC/Linux porting project, please see http://www.thepuffingroup.com/parisc/.

1. Copyright and Licensing

2. Supported Hardware

The following PA-RISC machines can be booted to a sash prompt using the latest PA-RISC/Linux development kernel.

- any 712, including 712/60, 712/64, 712/80, 712/100
- newer 715s, including 715/75, 715/100
- A180C and similar

The following hardware won't work:

- anything PA1.0 or PA2.0 based
- older 715s have I/O architecture problems that are still being worked on

3. Preparing to boot

3.1. Consoles

In order to boot your PA-RISC system with the PA-RISC/Linux kernel, you must first get a console on that system. You can use either a graphics console, which requires that a monitor be attached to the system, or a serial console, which allows communication with the system from a remote Linux machine.

If you have a monitor for your PA-RISC system, the easiest method is to use a graphics console.

3.1.1. Using a graphic console

Using a graphic console is just a fancy way of saying "use the monitor attached to the machine". If you have a monitor for your PA-RISC system, you should be able to bring up a console using the monitor, without having to mess with changing to and configuring a serial console.

Obviously, if you can use a graphic console, this is the easiest way to proceed.

3.1.2. Using a serial console

The only time that you should have to use a serial console is if you either don't have a monitor for your PA-RISC machine, or your machine doesn't support graphics. If you need to use a serial console, use the following procedures to set up serial console support.

3.1.2.1. Serial Cable

To connect your PA-RISC machine to your PC so you can boot the system using a console, you need a 9 pin-to-9 pin female null-modem cable. You should be able to obtain such a cable at your local computer hardware supplier.

3.1.2.2. Configuring minicom on Linux

In order to communicate with your PA-RISC machine during this process, you have to set it up in "serial console" mode and configure a serial communication program. We recommend "minicom", which is installed in most default Linux distributions. If you don't have minicom on your system, you can find the latest package at any of the major Linux software sites.

Most of the minicom configuration is machine-dependent. You must, however, ensure that:

a. the baud rate is set to 9600
b. parity is set to 8-N-1

3.1.2.3. Switching consoles

Before you can set up a serial connection between your Linux machine and your PA-RISC machine, you have to change the console mode on the PA-RISC system.

The default console mode is graphic_1. To change this, use the following procedure:

1. Turn the PA-RISC machine on.
2. During the boot process, the following message will appear:

   Searching for Potential Boot Devices.
   To terminate search, press and hold the ESCAPE key.

   When this message appears, press and hold the Esc key until an options menu appears.

3. The options menu looks like this:

   b) Boot from specified device
   s) Search for bootable devices
   a) Enter Boot Administration mode
   x) Exit and continue boot sequence
   ?) Help

   Select from menu:

   Select "a) Enter Boot Administration mode". This will bring up a "BOOT_ADMIN>" prompt. Everything else you do will be in BOOT_ADMIN mode.

3.1.2.3.1. Checking current console mode

Type: path console to see to which mode the console is currently set.

If it's in graphic console mode, it will return "Console path = graphic_1".

If it's set to serial console, it will return "Console path = rs232_a.9600.8.none" or something similar.

3.1.2.3.2. Changing to serial console mode

To change to serial console mode, type the following command at the BOOT_ADMIN command prompt:

path console rs232_a.9600.8.none

To verify that the console path has been properly set, type path console. This should return Console path = rs232_a.9600.8.none, indicating that the system is now set to boot in serial console mode.

3.1.2.3.3. How can I change the boot console to serial on a 712?

Unfortunately, you can't. Although 712s are configured for in-house HP development to use serial console, this configuration isn't supported in the field. You have to use a graphics console for 712s.

3.2. Preparing to boot from the network

3.2.1. rbootd or bootp?

All new machines can boot using bootp, including the 715/100, /120, and /75, as well as all 712s.

3.2.2. Using rbootd

3.2.2.1. Obtaining rbootd

If you have an older machine that requires rbootd to boot off a network, use the following procedure to set up, configure, and boot using the PA-RISC/Linux kernel.

Older machines, including the 715/33 and /50, require rbootd. You can obtain rbootd:

for RedHat, from ftp://puffin.external.hp.com/pub/parisc/binaries/RPMS/

for Debian, from http://www.debian.org/Packages/stable/net/rbootd.html

3.2.2.2. Configuring rbootd

To boot a PA-RISC 715 system, you have to have rbootd installed on the Linux system (a "boot server") where you will be storing the PA-RISC/Linux kernel image that you want to use to boot your PA-RISC system.

Once installed, use the following procedure to configure rbootd to work with your PA-RISC system:

1. In /etc/rbootd.conf add the following line:

   ethernet addr bootfile

   Replace bootfile with the name of your PA-RISC/Linux kernel image, usually "vmLinux".

2. Now get the ethernet address of your PA-RISC system by typing lan_addr at the BOOT_ADMIN prompt on your PA-RISC system.

   It will return a number like 080009-7004b6. Make note of the number returned.

3. In /etc/rbootd.conf on your boot server, the ethernet address has to be colon-delimited. So, you'll have to modify the number you just obtained so every two characters (after removing the "-") is separated by a colon. For example:

   080009-7004b6

   becomes

   08:00:09:70:04:b6

Add the colon delimited ethernet address to /etc/rbootd.conf on your boot server. The resulting file will look something like this:

```
# ethernet addr      boot file(s)      comments

08:00:09:87:e4:8f    vmlinux           # Linux
08:00:09:70:04:b6    stanley           # Linux
```

This rbootd.conf example contains the ethernet addresses and boot file names for two different machines.

Once you have changed the configuration file, restart rbootd.

> Note: By default, rbootd assumes that bootfiles are located in /export/hp/rbootd/. Therefore, you will have to put your bootable kernel image in that directory, or, if you really hate that directory for some reason, you can recompile rbootd to use a different directory.
>
> The easiest thing, of course, is to just drop your kernel images in /export/hp/rbootd/.

3.2.3. Using bootp/tftp

3.2.3.1. Obtaining bootp/tftp

TBD

3.2.3.2. Configuring bootp/tftp

Use the following procedure to use bootp on your boot server:

1. Configure /etc/inetd.conf on your boot server by adding the following lines:

   ```
   tftp   dgram   udp   wait   root   /usr/sbin/tcpd in.tftpd/tftpboot
   ```

bootps dgram udp wait root /usr/sbin/bootpd /etc/bootptab

Here, /tftpboot is being used for all file placement. You can choose another directory if you want.

When this is done, restart inetd with: /etc/rc.d/init.d/inetd restart.

2. Set up the /etc/bootptab file to contain:

```
[hostname]:hd=/tftpboot/Image:\
                    :rp=/usr/src/parisc/:\
                    :ht=ethernet:\
                    :ha=[mac address]:\
                    :ip=[ip address]:\
                    :bf=[boot file
name]:\
                    :sm=255.255.255.0:\
                    :to=7200:
```

You have to fill in the [hostname], [mac address], and [ip address] with the appropriate information, of course, where:

[hostname] is the name of the PA-RISC host

[mac address] is the ethernet address of the PA-RISC box, which you obtain by typing lan_address at the BOOT_ADMIN> prompt

[ip address] is the IP address of the PA-RISC system

[boot file name] is the name of the bootable kernel image (usually Image)

You'll end up with something like this:

```
vodka:hd=/tftpboot:\
                    :rp=/usr/src/parisc/:\
                    :ht=ethernet:\
                    :ha=080069088717:\
                    :ip=140.244.9.208:\
                    :bf=Image:\
                    :sm=255.255.255.0:\
                    :to=7200:
```

3.3. Preparing to boot from disk

TBD

3.4. Changing the boot path

3.4.1. When booting from the network

You now have to change the PA-RISC system's boot path to allow for booting from a Linux kernel image stored elsewhere on the network.

First, at the BOOT_ADMIN> prompt, type path alt lan. This sets the "alternet boot path" to "local area network".

Now type autoselect off to shut off the boot autoselect mechanism.

3.4.2. When booting from disk

TBD

3.5. Obtaining and using the latest kernel

You can obtain the most recent PA-RISC/Linux development kernel at the following ftp site. Note that the kernel images are named according to the date they were created:

ftp://puffin.external.hp.com/pub/parisc/binaries/kernels/

When you have downloaded the latest image, we recommend that you rename it to Image. You must then place it in one of the following directories, depending on the boot method you are using:

If you are booting using rbootd, put the Image file in x

If you are booting using bootp, put the Image file in x.

If you are booting from disk, put the Image file in ...

4. Booting your PA-RISC/Linux system

4.1. Booting from the network

Whether you're booting using rbootd or bootp, the procedure for booting is identical:

1. Get to a BOOT_ADMIN> prompt.
2. Type boot lan

This will retrieve the kernel from the network boot server and use it to boot the PA-RISC machine.

4.2. Booting from disk

TBD

5. HOWTO contributors

The following people contributed to this fine HOWTO in one way or another.

David Alexander deVries <adevries@thepuffingroup.com>

Philip Imperial Schwan <pschwan@thepuffingroup.com>