May 2001
DTN-21 Revision 0-0

Adding a Kernel to the Sun Cobalt™ RaQ 3, 4, and XTR Server and Qube 3 Appliance Families

1 Overview

The Sun Cobalt™ RaQ 3, 4, and XTR servers and the Qube 3 appliance ROM is designed to load a Linux kernel. Developers who build their own Linux kernels need to be aware of special considerations to work on a Cobalt server. This Technical Note tells you how to successfully build and boot your kernel on a Cobalt system.

Table of Contents

Technical notes are located in `ftp://ftp.cobaltnet.com/pub/developer/`

1.1 Audience

This Technical Note is intended only for software developers creating or modifying Linux kernels to be used with any of Sun Cobalt's server. Most software developers will use the Linux kernel that ships with the Cobalt server, and will not need to build one themselves. Also, since the existing kernel fully supports the Linux module mechanism, many desired new features, such as plug-in device drivers, can be added as modules to the existing kernel.

> **Note**
> Replacing the Linux kernel voids your Sun Cobalt software warranty. This means that Cobalt Technical Support cannot help you if you call them. Cobalt strongly recommends that you try building your kernel on a nonproduction system before installing it on a mission-critical system.

1.2 Applicable Products

This Technical Note applies to the Sun Cobalt™ RaQ 3, RaQ 4, and RaQ XTR servers and Qube 3 appliance. These are all Intel-compatible systems. These directions do not apply to any MIPS processor based systems.

This note applies only to 2.2 series Linux kernels with versions greater than or equal to 2.2.16C20. The latest kernel version always includes functionality and security enhancements. Use of old kernels is **not** recommended.

Note

This technical note applies only to kernel versions greater than 2.2.16C20, which may be included in the Sun Cobalt™ Qube 3 appliance and RaQ 3 and 4 servers. For kernel versions less than 2.2.16C20, see Technical Note 6.

2 Adding a Kernel on the Sun Cobalt™ Server Family using the ROM Interface

To debug any kernel issues, you must have a serial console. Attaching a serial console differs slightly on different servers.

2.1 For Sun Cobalt RaQ Server Families

1) Attach a null-modem cable to the first serial port on the back of the Sun Cobalt RaQ server, which is the console port.

2) Using a terminal emulator program of your choice, for example, `minicom` in Linux, `SmartModem` or `HyperTerm` in Windows, or `zterm` in Macintosh OS, set the port configuration to 115200 baud, 8 data bits, no stop-bits, 1 parity bit (commonly called 8-N-1).

2.2 For Sun Cobalt Qube 3 Appliance

1) Attach a null-modem cable to the serial port on the back of the Sun Cobalt Qube 3 appliance.

2) Using a terminal emulator program of your choice, for example, minicom in Linux, SmartModem or HyperTerm in Windows, or zterm in Mac OS, set the port configuration to 115200 baud, 8 data bits, no stop-bits, 1 parity bit (commonly called 8-N-1). If no console appears on the terminal software, the Qube's console is turned off: proceed to step 3. If a console appears, you're done.

3) Shut down the Sun Cobalt Qube 3 appliance, and turn it off.

4) Using a paper clip or pin, depress and hold the recessed `reset-password` button on the back of the Sun Cobalt Qube 3 appliance, and turn the Qube on. The LCD panel should display the message `console ON`. If it displays `console OFF`, turn off the Sun Cobalt Qube 3 appliance and repeat this step. After you see `console ON`, you can release the button.

3 Building your Kernel

If you are using the Sun Cobalt™ kernel source as a basis for your development, you should obtain the newest kernel packages available for your system. Official kernel updates are provided as Cobalt `.pkg` files. Experimental (unsupported) kernels are available at the Sun Cobalt FTP site:

```
ftp://ftp.cobaltnet.com/pub/experimental
```

Sun Cobalt updates the kernel as necessary. The latest experimental kernel releases can always be found at the above URL. The most recent `bwmgmt` package, if needed, for each kernel revision is also located at the above URL.

To install RPM files, use the `-U` upgrade option to the RPM command:

```
% rpm -Uvh kernel*2.2.12C5-1*
```

Before building your kernel, you must decide which of the two generations of Sun Cobalt servers you plan to use. Sun Cobalt servers are divided into two generations:

- Generation `III`, which includes RaQ 3 and RaQ 4 servers, and Qube 3 appliance
- Generation `V`, which includes RaQ XTR server.

Important

If you need to cover all Sun Cobalt servers, you should build a kernel for each generation.

To build a kernel, follow these directions:

1) Change directories to the kernel source directory. Type:

```
cd /usr/src/linux
```

2) Set your generation variable. Type:

```
export COBALT_GENERATION="_III""
or
export COBALT_GENERATION="_V"
```

3) Build a kernel configuration file. Type:

```
make oldconfig
```

4) Prepare a your build environment. Type:

```
make dep clean
```

5) Build the kernel image. On the Cobalt-supplied kernel source, type:

```
make cobalt
```

6) On the non-Cobalt-supplied source, type:

```
make vmlinux
gzip -f -9 vmlinux
```

This build takes several minutes to complete. This build compiles a kernel that has the same features as the default Linux kernel. Checksums do not match, because of embedded date and time strings in the kernel image.

Note

If you are familiar with Linux on standard personal computers, you might be familiar with the `make zImage` or make `bzImage` commands. Do **not** use these commands. Cobalt systems do not need the extra information in these formats; these commands do **not** work.

---

7) Make the changes you are planning to the kernel source tree.

8) Repeat the commands in steps 4 and 5. You should now have a kernel image that reflects your changes.

9) Copy the new file to a new file in `/boot`, for example:

```
cp vmlinux.gz /boot/mykern.gz
```

After you have built your kernel, follow these steps:

1) Reboot.
2) Press the space key from the terminal emulator window during the boot process after you see the ROM banner. You should see a prompt:

```
Cobalt:Main Menu>
```

3) To boot your kernel, type:

```
bfd /boot/mykern.gz
```

4) When the system has finished booting, verify that the kernel that is running is the correct version by looking at the date in the output of the following command:

```
uname -a
```

5) If you want, you can now make your kernel the default. First, save the old kernel. Type:

```
cp /boot/vmlinux.gz /boot/cobalt-vmlinux.gz
cp /boot/System.map /boot/cobalt-System.map
mv /boot/mykern.gz /boot/vmlinux.gz
cp /usr/src/linux/System.map /boot/System.map
```

Note
If you ever need to boot the original Sun Cobalt kernel, repeat steps 1 through 5 above, substituting the name `/boot/cobalt-vmlinux.gz` for `/boot/mykern.gz.`

The `/usr/games/.doug` backup kernel from Sun Cobalt's MIPS-based server does not work on newer systems.

---

4 Booting From ROM

If needed, you can also boot the server using the `bfr` (boot from ROM) command in the boot menu; see steps 1 through 5 above. This brings the system up running the kernel that is in ROM.

You can also boot the ROM kernel from the front panel:

1) While booting, when the Cobalt logo is scrolling, press and hold the **Select** button until the LCD displays `Select option`:

Note
Press the S button for Select. Do **not** press the four directional buttons.

---

2) Use **Select** to scroll to **Boot from ROM**.
3) Press **Enter**.

You need the hard disk in the Sun Cobalt server for the file system, but the system will boot from a known good kernel, even if the kernel you compiled fails to boot.