

Kernel 2.6.18 compilation for COPPER CPU

Sep. 5, 2013

Satoru Yamada

Since we found that the default kernel (2.6.18) is unstable on COPPER CPU unfortunately, we recommend you to configure and compile the kernel to obtain more stable OS.

1, Get kernel 2.6.18 source and build modules

```
[boot server] % wget https://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.18.tar.gz
```

```
[boot server] # mv linux-2.6.18 /usr/src/kernels/
```

Before compiling, check if the system time is correct, otherwise the compilation never ends.

```
% date
```

If it is not, correct by some commands.

```
# date --set="2013/09:05 8:46" or ntpdate
```

Check gcc version. (I failed to boot Copper CPU when compiling kernel w/ gcc 4.7.3)

```
% which gcc
```

```
% /usr/bin/gcc
```

```
% gcc -dumpversion
```

```
4.1.2
```

2, Get network adapter driver for PCI-E Gigabit network connection

It is not available in the kernel tar ball.

2-1, Copy a driver directory

To compile e1000e driver, it seems that /lib/module/BUILD-KERNEL is needed. So compile modules w/o e1000e first.

Get a configuration file from svn, (use svn **rev. 6535** or later)

<https://belle2.cc.kek.jp/svn/trunk/software/daq/copper/driver/kernel/config-130903>

```
[boot server] # cp config-130903 /usr/src/kernels/linux-2.6.18/.config
```

```
[boot server] # cd /usr/src/kernels/linux-2.6.18
```

```
[boot server] # make clean
```

```
[boot server] # make modules -j10
```

“-j10” means use 10 threads for compiling. This option is not necessary.

```
[boot server] # make modules_install -j10
```

2-2, download and compile driver source

You can download it from

https://downloadcenter.intel.com/Detail_Desc.aspx?lang=jpn&DwnldID=15817

```
[boot server] % gtar xvf e1000e-2.4.14.tar.gz
```

```
[boot server] # cp -r e1000e-2.4.14/src/* /usr/src/linux-2.6.18/drivers/net/e1000e
```

```
[boot server] # vi /usr/src/linux-2.6.18/drivers/net/e1000e/Makefile
```

Read Makefile and add a line that “BUILD_KERNEL=2.6.18” at a proper position so that it would not be overwritten afterwards.

2-3, Modify Kconfig and Makefile of kernel directory

- Add the following lines to /usr/src/kernels/linux-2.6.18/drivers/net/Kconfig

config E1000E

tristate "Intel(R) PRO/1000 PCI-Express Gigabit Ethernet support"

depends on PCI

---help---

This driver supports the PCI-Express Intel(R) PRO/1000 gigabit ethernet family of adapters. For PCI or PCI-X e1000 adapters, use the regular e1000 driver For more information on how to identify your adapter, go to the Adapter & Driver ID Guide at:

<<http://support.intel.com/support/network/adapter/pro100/21397.htm>>

For general information and support, go to the Intel support website at:

<<http://support.intel.com>>

More specific information on configuring the driver is in <file:Documentation/networking/e1000e.txt>.

To compile this driver as a module, choose M here and read

<file:Documentation/networking/net-modules.txt>. The module will be called e1000e.

- Add an e1000e related line to /usr/src/kernels/linux-2.6.18/drivers/net/Makefile

```
obj-$(CONFIG_E1000) += e1000/  
+ obj-$(CONFIG_E1000E) += e1000e/  
obj-$(CONFIG_IBM_EMAC) += ibm_emac/
```

3, compile and install kernel

```
[boot server] # cd /usr/src/kernels/linux-2.6.18
```

```
[boot server] # sudo make bzImage -j10    or    make bzImage -j10
```

Then you might be asked:

“Intel(R) PRO/1000 PCI-Express Gigabit Ethernet support (E1000E) [N/m/y/?] (NEW) ”

Then type m.

After building, bzImage should be made.

```
[boot server] # make modules -j10
```

```
[boot server] # make modules_install -j10
```

```
[boot server] # make install
```

Check /etc/grub.conf if this kernel is not the default kernel for booting a PXE boot server, otherwise you may fail to boot the server.

4, copy the kernel to tftpboot directory

```
[boot server] # cp -r /lib/modules/2.6.18 /tftpboot/copper/root/lib/modules/
```

```
[boot server] # cp -r /usr/src/kernels/linux-2.6.18 /tftpboot/copper/root/usr/src/kernels
```

```
[boot server] # cp /boot/initrd-2.6.18.img /tftpboot/copper/root/boot
```

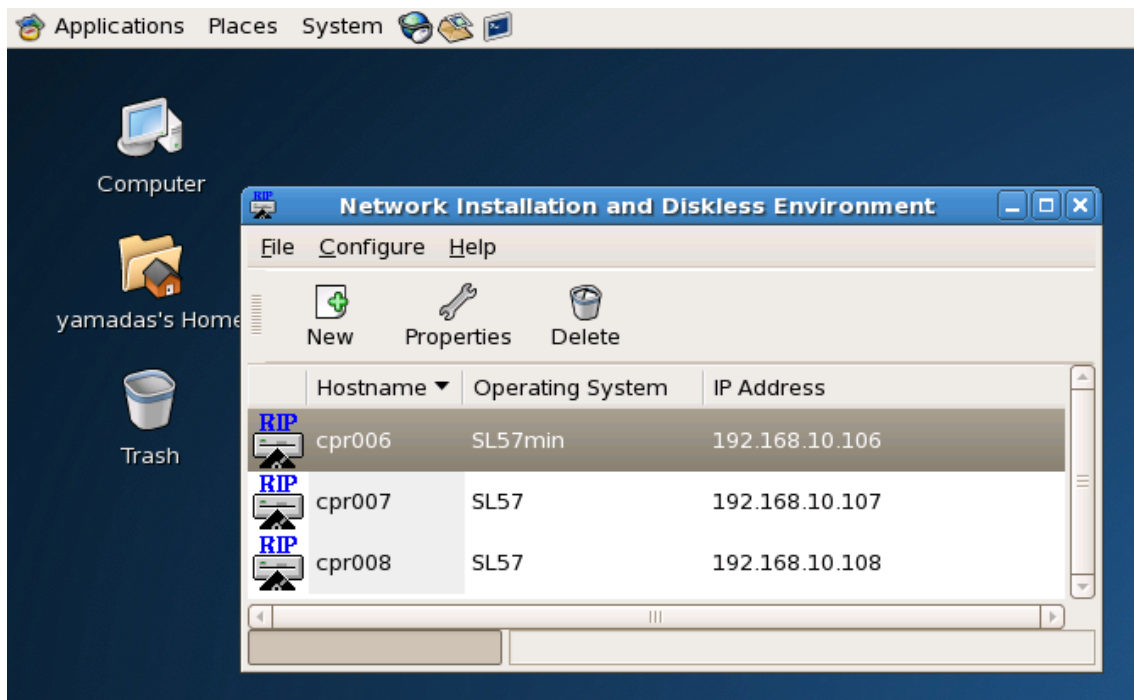
```
[boot server] # cp /boot/System.map-2.6.18 /tftpboot/copper/root/boot
```

```
[boot server] # cp /boot/vmlinuz-2.6.18 /tftpboot/copper/root/boot
```

5, Modify diskless client setup on PXE boot server

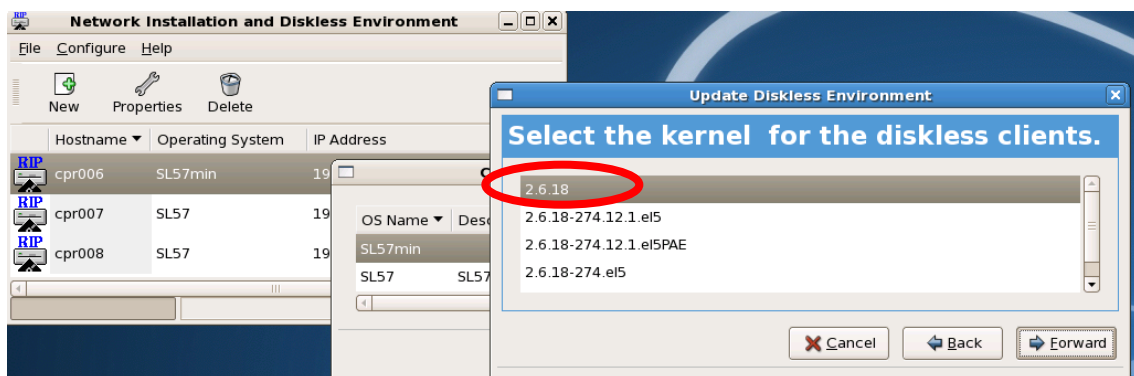
System -> administration -> server setting

Configure->Diskless-> Add



Proceed in the same way as shown in https://belle2.cc.kek.jp/svn/trunk/software/daq/copper/doc/SetupPocketDAQ_1_PXE_boots/tserver.pdf

A new kernel entry should appear on a kernel list.

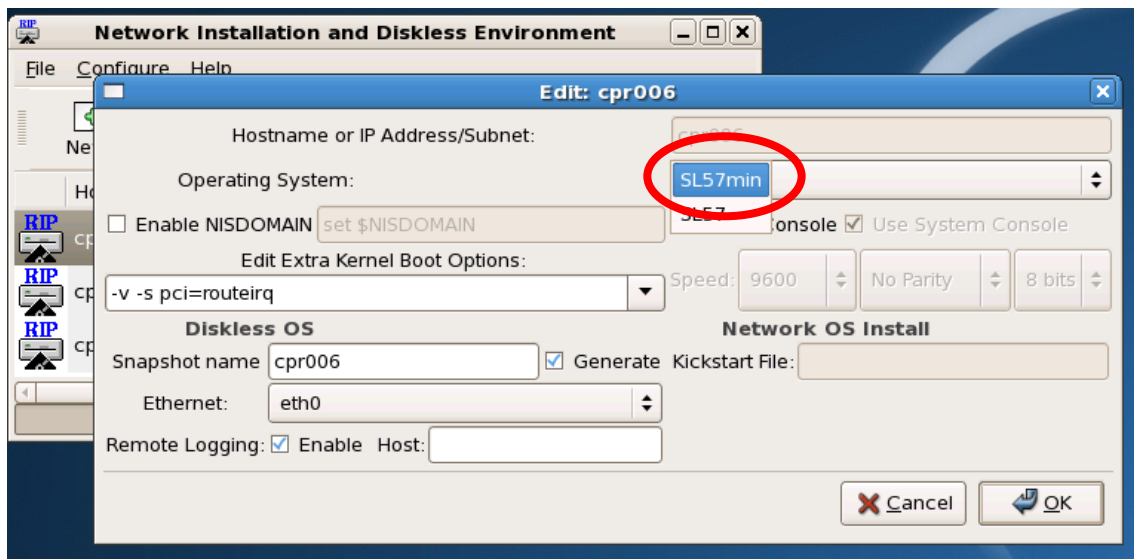


You need to check kernel image is copied to /tftpboot/linux-install/SL57min.

```
% ls -l /tftpboot/linux-install/SL57min
```

Notice : When you update /tftpboot/copper/boot/*, do not forget to make (or replace) a new entry of "Operating System" using this GUI.

After updating diskless environment, each COPPER CPU's setting also should be modified.



Please check /tftpboot/linux-install/pxelinux.cfg/C0A80A** files so that the modification is correctly applied.

6, Boot COPPER CPU

7, Update copper and cpr_fngeneric drivers

You need to recompile COPPER related drivers when you update the kernel, otherwise you may receive an error message below during installation of modules.

"insmod: error inserting 'daq/cprdist-0.1.0/drv/copper.ko': -1 Invalid module format"

Before compiling drivers, please read the following notices;

NOTICE : The version of compiler should be same as the one used for the kernel compilation.

NOTICE : Maybe you will compile COPPER related drivers on a PXE server. In that case, please do not forget to modify KERNEL_VERSION in drv/Makefile and cprfn_fngeneric/Makefile like;

```
#KERNEL_VERSION = $(shell uname -r)
```

```
KERNEL_VERSION = 2.6.18
```