KERNEL COMPILATION

`/usr/src/kernel-source-2.4.27` directory will be named as the linux directory.

- Change your directory to linux directory.

- Open **`Makefile`** file. You'll see:

```
VERSION = 2
PATCHLEVEL = 4
SUBLEVEL = 27
EXTRAVERSION = -10
```

as the first 4 lines. The first three one indicates your current kernel version 2.4.27. Now for your new compiled kernel you'll define a new version and you'll do it by EXTRAVERSION. You will name it by yourself but don't forget this, because you'll use this 4 together later on. We'll be calling these four as the version from now on. (e.g, our version may be 2.4.27-10) . Our EXTRAVERSION is -10.

- We begin the configuration by wiping out all previous configurations and resetting the source directory to a pristine state. The main reason for doing this is that some files do not automatically get rebuilt, which can lead to failed builds, or at worst, a buggy kernel.

```
# make mrproper
```

Use one of the config options. (config, menuconfig, xconfig). **You may need to enable Socket Filtering in Network Options, and Intel Gigabit Ethernet Adaptor Driver in Network Devices. (Use the BBB Course recording on March 20th to see it more clearly)**:

```
# make menuconfig
```

The next step is to create the necessary include files and generate dependency information:

```
# make dep
```

Once the dependency information is created we can clean up some miscellaneous object files:

```
# make clean
```

We are now (finally) ready to start the actual kernel build. At the prompt type:

```
# make bzImage
```

Check whether our bzImage is created, type in the command line:

```
# ls -l arch/i386/boot
```

To build the modules we run:

```
# make modules
```

Once the modules are built they can be installed. If you were building as a non-privileged user you will now need to switch to root to complete this next step:

```
# make modules_install
```

Once your kernel is created, you can prepare it for use. From the linux directory, run these commands.

```
# cp arch/i386/boot/bzImage /boot/bzImage-2.4.27-10
```

```
# cp System.map /boot/System.map-2.4.27-10
```

If you have built your main boot drivers as modules (e.g., SCSI host adapter, filesystem, RAID drivers) then you will need to create an initial RAMdisk image. The initrd is a way of sidestepping the chicken and egg problem of booting -- drivers are needed to load the root filesystem but the filesystem cannot be loaded because the drivers are on the filesystem. As to create the initrd, do the following:

> mkinitrd creates filesystem images which are suitable for use as Linux inital ramdisk (initrd) images. Such images are often used for preloading the block device modules (such as IDE, SCSI or RAID) which are needed to access the root filesystem. mkinitrd automatically loads file system modules (such as ext3 and jdb), IDE modules, all scsi_hostadapter entries in /etc/modules.conf, and raid modules if the systems root partition is on raid, which makes it simple to build and use kernels using modular device drivers.

To create the initrd, do the following:

```
# mkinitrd -o /boot/initrd-2.4.27-10 /lib/modules/2.4.27-10
```

In this case please be careful for the version. You should write exactly the same version to this command as you write in the Makefile file as we explained above.

After that, open **/boot/grub/menu.lst** with a text editor and add the following **bold red** lines to the end of the file without modifying the rest. **(Do this only once)**

```
## ## End Default Options ##

title           Debian GNU/Linux, kernel 2.4.27-3-386
root            (hd0,0)
kernel          /boot/vmlinuz-2.4.27-3-386 root=/dev/hda1 ro
initrd          /boot/initrd.img-2.4.27-3-386
savedefault
boot

title           Debian GNU/Linux, kernel 2.4.27-3-386 (recovery mode)
root            (hd0,0)
kernel          /boot/vmlinuz-2.4.27-3-386 root=/dev/hda1 ro single
initrd          /boot/initrd.img-2.4.27-3-386
savedefault
boot

title           Debian GNU/Linux, kernel 2.4.27-10
root            (hd0,0)
kernel          /boot/bzImage-2.4.27-10 root=/dev/hda1 ro
initrd          /boot/initrd-2.4.27-10
savedefault
boot

### END DEBIAN AUTOMAGIC KERNELS LIST
```

DO NOT FORGET!!!

(1) You should boot your computer with 2.4.27-10 kernel and select it in the boot grub menu when you want to see changes you will have done in your kernel.
(2) When you make a change in the 2.4.27-10 kernel code, you should recompile it.
(3) You should boot your computer with the 2.4.27-3-386 kernel when you recompile the 2.4.27-10 kernel.