Step # 1 Get Latest Linux kernel code

Visit http://kernel.org/ and download the latest source code. File name would be linux-x.y.z.tar.bz2, where x.y.z is actual version number. For example file inux-2.6.25.tar.bz2 represents 2.6.25 kernel version. Use wget command to download kernel source code:
$ cd /tmp
$ wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-x.y.z.tar.bz2

Note: Replace x.y.z with actual version number.

Step # 2 Extract tar (.tar.bz3) file

Type the following command:
tar -xjvf linux-2.6.25.tar.bz2 -C /usr/src
cd /usr/src

Step # 3 Configure kernel

Before you configure kernel make sure you have development tools (gcc compilers and related tools) are installed on your system. If gcc compiler and tools are not installed then use apt-get command under Debian Linux to install development tools.
apt-get install gcc

Now you can start kernel configuration by typing any one of the command:

- **$ make menuconfig** – Text based color menus, radiolists & dialogs. This option also useful on remote server if you wanna compile kernel remotely.
- **$ make xconfig** – X windows (Qt) based configuration tool, works best under KDE desktop
- **$ make gconfig** – X windows (Gtk) based configuration tool, works best under Gnome Dekstop.

For example make menuconfig command launches following screen:
$ make menuconfig

You have to select different options as per your need. Each configuration option has HELP button associated with it so select help button to get help.

Step # 4 Compile kernel

Start compiling to create a compressed kernel image, enter:
$ make
Start compiling to kernel modules:
$ make modules

Install kernel modules (become a root user, use su command):
$ su -

make modules_install

Step # 5 Install kernel

So far we have compiled kernel and installed kernel modules. It is time to install kernel itself.
make install

It will install three files into /boot directory as well as modification to your kernel grub configuration file:

- System.map-2.6.25
- config-2.6.25
- vmlinuz-2.6.25

Step # 6: Create an initrd image

Type the following command at a shell prompt:
cd /boot
mkinitrd -o initrd.img-2.6.25 2.6.25

initrd images contains device driver which needed to load rest of the operating system later on. Not all computer requires initrd, but it is safe to create one.

Step # 7 Modify Grub configuration file – /boot/grub/menu.lst

Open file using vi:
vi /boot/grub/menu.lst

```
title          Debian GNU/Linux, kernel 2.6.25 Default
root           (hd0,0)
kernel         /boot/vmlinuz root=/dev/hdb1 ro
initrd         /boot/initrd.img-2.6.25
savedefault
boot
```

Remember to setup correct root=/dev/hdXX device. Save and close the file. If you think editing and writing all lines by hand is too much for you, try out update-grub command to update the lines for each kernel in /boot/grub/menu.lst file. Just type the command:
update-grub
Neat. Huh?

Step # 8 : Reboot computer and boot into your new kernel

Just issue reboot command:
reboot

Step 3.1 : Generic Way

Step 3.1.1 Compile the Kernel and its modules

make
above code is take 1 to 2 hours depending on system performance. In this step it compile the kernel and store the kernel in binary form to **arch/x86/boot/bzImage** file
then based on kernel headers it will compile the kernel modules (device driver, file-system, network,...)
and generate .ko files. ko means kernel object. These modules also called Loadable Kernel Modules (LKM).

Step 3.1.2 Install Kernel modules

make modules_install
This step copy the all kernel modules
(*.ko) to **/lib/modules/<version>/kernel/** folder.

Step 3.1.3 Install Kernel

make install
This step copy the the kenel from **arch/x86/boot/bzImage** to **/boot** folder and copy the **.config** file to **/boot/config-<latest-version>** and generate the **System.map** file.

Step 3.1.4 Create Initramfs file

up to now kernel and its modules are compiled and installed. when next boot up time we need to choose latest kernel. so we need to prepare the boot-loader and its support files. When system turns on, after bios and boot loader load the kernel to main memory and mount initial dummy file system as a root file system of system. this initial file system have necessary drivers for disk hardware (SCSI or IDE) and mount the correct file system as a root file system.

so we need to create initramfs file using update-initramfs or mkinitfs tool

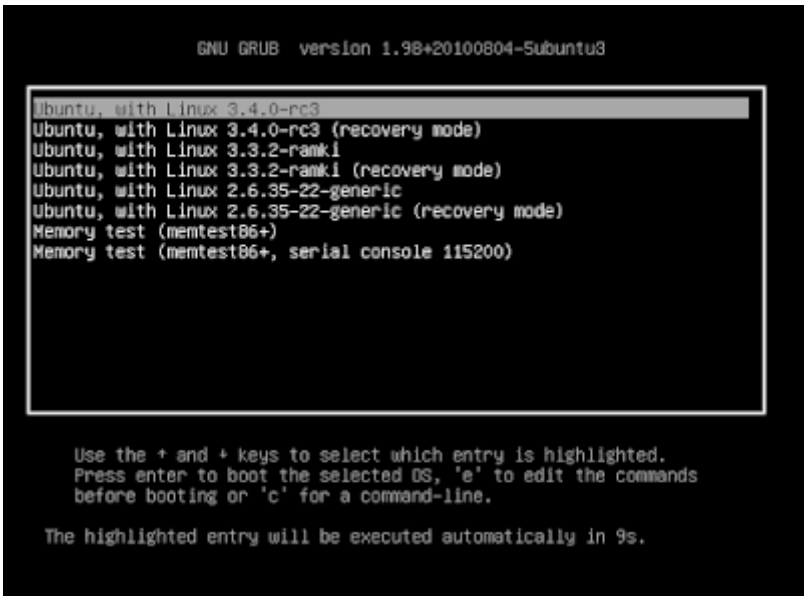
update-initramfs -c -k 3.3.3
here 3.3.3 is new kernel version.

Step 3.1.5 Update GRUB bootloader

the last step is update the boot loader here i m using GRUB boot-loader.

update-grub
this command automatically probe the kernels in **/boot** folder and add the entries in its configuration file, **grub.cfg**

now restart the system , we will see the new kernel is added in boot loader entries. then choose new kernel in boot loader.



now open the terminal and issue **uname -r** command, its shows the current kernel version.