Compiling the Linux Kernel (Ubuntu) – With explanations

This is a quicker and easier way to compile the kernel after the first time as opposed to using simple 'make'

It works for Debian packages.

Compiling for the first time

1. sudo bash

Makes you the root. You will be prompted for your password.

2. rm -f /bin/sh

3. ln -s /bin/bash /bin/sh

Just a cautionary step to make /bin/sh a symlink to /bin/bash

4. apt-get update

Updates the list of available packages

5. apt-get install kernel-package libncurses5-dev fakeroot wget bzip2

Installs the mentioned packages. The ncurses library is needed. So you should install it.

Henceforth, whenever a package seems to be missing (e.g. dpkg) just use the above command as *apt-get install package-name*

Also, *apt-cache --name-only search package-name*, will search the list of available packages for the package-name and tell you the exact name to use

6. cd /usr/src

7. wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-x.x.xx.x.tar.bz2

Downloads the kernel source to /usr/src. (Use apt-get for wget if necessary!)

8. tar xjf linux-x.x.xx.x.tar.bz2

9. ln -s linux-x.x.xx.x.1 linux

10. cd /usr/src/linux

Untars the tar file to the mentioned linux directory. Also creates a sym link to linux so henceforth, 'linux' will point to 'linux-x.x.xx.x'

11. cp /boot/config-`uname -r` ./.config

Use this command only if you want to use your existing kernel configuration (Recommended if you do not want to configure kernel again)

12. make menuconfig

Brings up the config menu. If above step was followed, you can select ‘Load Alternate Configuration File’
Else you can configure it to have only the things you need and make the kernel lean and fast.

13. make-kpkg clean
Removes all old linkings and object files to build your kernel from scratch

14. fakeroot make-kpkg --initrd --append-to-version=-your_kernel_name kernel_image kernel_headers
Starts building the kernel

15. cd /usr/src
16. ls –l
Change directories
You will find two new files here
linux-image-something.deb
linux-headers-something.deb

17. dpkg –i linux-image-something.deb
18. dpkg –I linux-headers-something.deb
This will install the kernel, modify your grub and make the kernel ready to boot. The beauty of the deb files is that you can now transport them to any computer and run the above two commands without compiling the kernel again.

19. shutdown –r now
Reboot

----------Kernel Loaded----------

Compiling Henceforth

Now you have at least two kernel versions:
.. Generic
.. Your_kernel_name

To modify your kernel, boot into Generic

Modify the source files under /usr/src/linux

Now do

1. sudo bash
2. fakeroot make-kpkg --initrd --append-to-version=-your_kernel_name kernel_image 3. kernel_headers

4. cd /usr/src
5. dpkg –i linux-image-something.deb
6. dpkg –I linux-headers-something.deb
7. shutdown –r now

.. Remember NOT to do any other steps from above.
.. By not making a new config file and by not doing make-kpkg clean, the system will just compile the modified files and their linked files.
.. Also remember to give the same your_kernel_name as before.
.. The system will ask you if you are sure, select ‘yes’ and select ‘no’ to abort. It should be fine!