

A SIMPLE GUIDE TO BUILDING YOUR OWN LINUX KERNEL

This article is for all the newbies who want to learn how to build their own Linux kernel. Following the steps mentioned will demonstrate how easy the task is!

Please note that the steps mentioned were done on Ubuntu 10.04, but would remain the same for any Linux distribution.

Pre-requisites

Git is the utility for version control on Linux. Install it with a simple `sudo apt-get install git-core`. You also need the curses library development files install the package with `sudo apt-get install libncurses5-dev`.

Next, verify the current Linux kernel version with the `uname -a` command so you can download the relevant kernel source to build using the `git clone` command. I have used Linux 2.6 in this example.

Configuring the kernel

Now that you have the source, you need to configure it. You can use the command `make menuconfig` if you are experienced on what configuration parameters to set. If you are new to this, I would suggest you use the default configuration copy the existing config file to the kernel source directory with, for example, `cp /boot/config/-2.6.32-38-generic .config`

Now, run `make oldconfig` to start the configure process. Please note that here you will be prompted to answer Yes or No. If you are unsure, just keep on hitting Enter (that is a default Yes).

Naming the kernel

Next, it might be good to give your kernel a name. For this, open the Makefile, and edit the lines below:

VERSION = 2

PATCHLEVEL = 6

SUBLEVEL = 32

EXTRAVERSION = -dips

NAME = Building My Kernel

Now, issue the make command. This should take a couple of hours to complete.

Installing the kernel

Now, issue the following command to install all the kernel modules:

```
make INSTALL_MOD_STRIP=1 modules_install
```

Next, run a make install.

Updating with initramfs

Run the command, `sudo update-initramfs -c -k 2.6.32-dips+`

For readers who are keen to know what initramfs is, it is a root filesystem that is loaded at an early stage of the boot process. It is the successor of the old initrd.

Verify the installation

Now, we are almost done! To verify, check your `/boot` directory. Are you able to find the new kernel image and the config file for your build? If yes, then congratulations! You have succeeded in building your kernel.

Modify the GRUB file and reboot

GRUB stands for Grand Unified Bootloader. It is a boot-loader package from GNU. GRUB provides a user the choice to boot any one of multiple operating systems installed on a computer, or to select a specific kernel configuration available on a particular operating system's partitions.

You can find the GRUB file at `/boot/grub/grub.cfg`.

Issue the command `sudo update-grub` and your GRUB file will be modified. You should be able to see an entry for your kernel, which looks like what's shown below:

```
### BEGIN /etc/grub.d/05_debian_theme ###

set menu_color_normal=white/black

set menu_color_highlight=black/light-gray

### END /etc/grub.d/05_debian_theme ###

### BEGIN /etc/grub.d/10_linux ###

menuentry Ubuntu, with Linux 2.6.32-dips+ --class ubuntu --class
gnu-linux --class gnu --class os {

    recordfail

    insmod ext2

    set root=(hd0,5)

    search --no-floppy --fs-uuid --set 0b0ebe63-24f6-45c0-a3df-
5d9c447b5ad9

    linux    /vmlinuz-2.6.32-dips+ root=UUID=325f1d1d-2070-45f3-
94ed-2027c5ffcbf3 ro    quiet splash

    initrd   /initrd.img-2.6.32-dips+

}
```

Now, just reboot the system. Your system should now boot up with your just-built kernel. You can verify it by using the `uname a` command again. That's all you need to build a kernel. Now wasn't that simple? So, have fun building your kernel!

Source : <http://www.opensourceforu.com/2013/05/a-simple-guide-to-building-your-own-linux-kernel/>