Raspberry Pi – Linux Kernel Cross Compilation

Raspberry Pi – Linux Kernel Cross Compilation
Mohamed Haneef M.A
MentorLinux
http://www.mentorlinux.wordpress.com

Contents

- Overview
- To do before kernel compilation
- Cross compilation
- Flashing the kernel image

Overview

I assume here the you have a raspberry pi with you and it is running on raspbian wheezy. To make things simpler on how to connect raspberry pi to your host machine please visit this.

My host machine is ubuntu 12.04 and my RPi is running on raspbian wheezy.

Kernel Compilation can be done by compiling the linux source code in the target machine (this may take days to get completed) or we could use a host machine to compile the source code (this is a more sensible idea), hence I am going to show you how it is to be compiled with a host machine.

To do (before compilation)

- Get the raspberry kernel source
  git://github.com/raspberrypi/linux.git
- Get the latest raspberry pi compiler
- git://github.com/raspberrypi/tools.git

To do (before compilation)

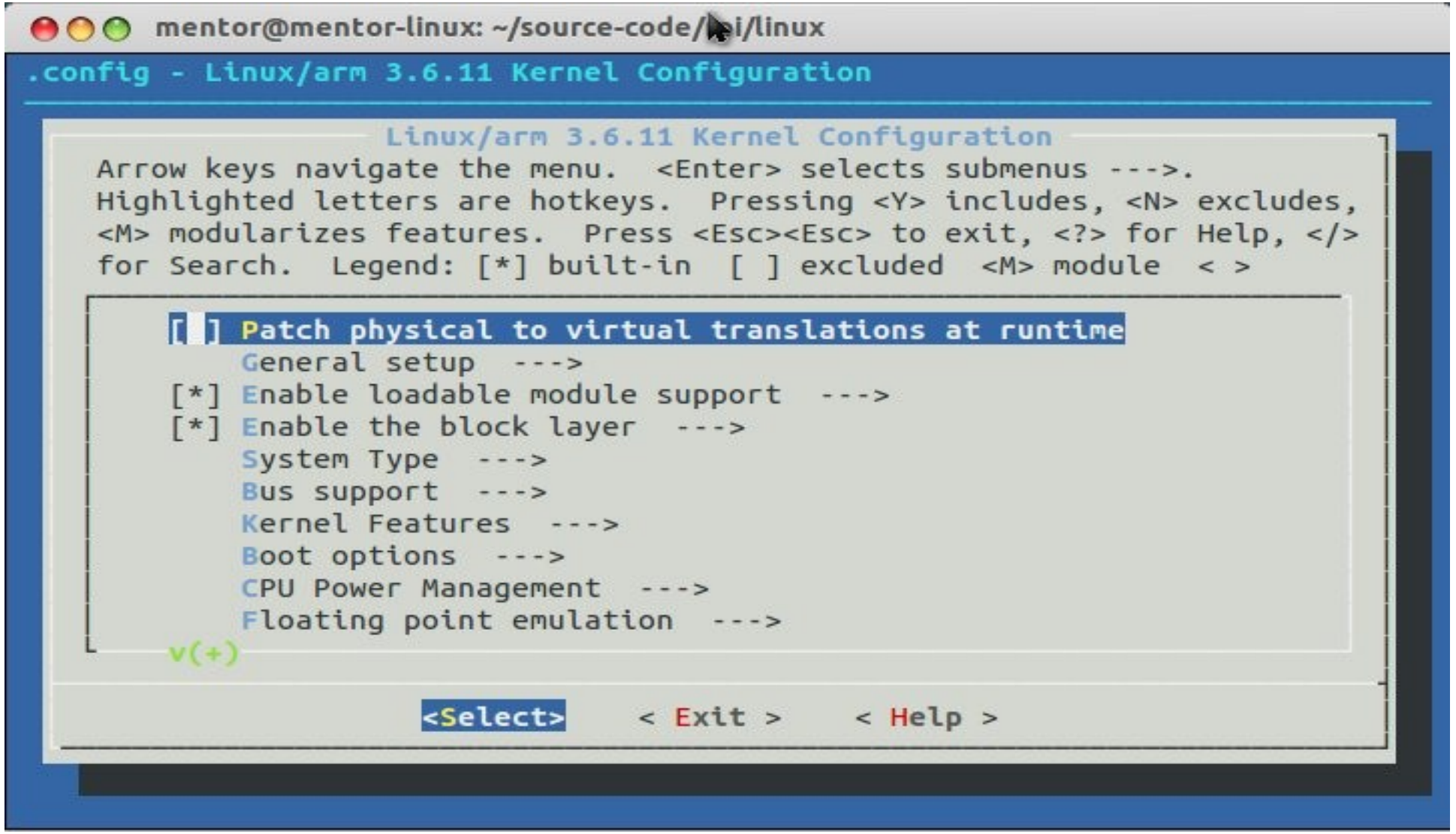
Open a terminal in your host machine and type
$ mkdir workdir
$ cd workdir
$ sudo git clone git://github.com/raspberrypi/linux.git
$ sudo git clone git://github.com/raspberrypi/tools.git

Raspberry Pi – Cross Compilation

- Once the linux source code has finished downloading
- Change to that directory by doing a “cd linux” from the terminal.
- Let us get the .config right.
- $cp arch/arm/configs/bcmrpi_defconfig .config
- The above copy command will create a .config file with default configurations
- If you want to make changes to the configurations then do a
- $ sudo make ARCH=arm CROSS_COMPILE={path to your cross compiler} menuconfig

Please select the loadable module support as shown below

Raspberry Pi – Cross Compilation



Raspberry Pi – Cross Compilation

Now we are ready to build the kernel – our configuration is set and we have also selected the lodable module support for dynamic modules.

So our next step is to buid the image – give the following command.

```
$sudo make ARCH=arm CROSS_COMPILE={path to your cross compiler}
```

So if every thing goes fine, you should be able to get a kernel image in arch/arm/boot/Image – Image is the kernel image which we will be flashing via a ssh connection into the /boot bolder by renaming it to kernel.img

Raspberry Pi – Flashing the kernel.img

So now we need to move to the arch/arm/boot directory, and from there give the following command.

$ sudo scp Image pi@192.168.10.2:/home/pi

This would send your Image to /home/pi.

- Now log into the raspberry pi via"ssh pi@192.168.10.2" and then go to the pi directory from there take a backup of kernel.img as

  $sudo cp /boot/kernel.img kernel_wrking_backup.img

- Then transfer the Image file as kernel.img to /boot directory as

  $sudo cp Image /boot/kernel.img

- Now we are ready to reboot Raspberry pi, if every thing was good enough then you should be able to ssh into RPi.