Linux Kernel Compilation & Installation

1. Raspberry PI kernel is stored in a Git repository and requires installing a Git client to fetch the files. To install Git client on your Local machine, run the following command:

**sudo apt-get install git**

1. Create a directory to store Raspberry-related files (e.g. **/home/<username>/Raspberrypi**) and run the following command in that directory to download the kernel source:

**git clone** [**https://github.com/raspberrypi/linux.git**](https://github.com/raspberrypi/linux.git)

1. Prepare the sources and tools
   1. Install packages used for cross compiling on the Ubuntu, run the following command

**sudo apt-get install gcc-arm-linux-gnueabi make git-core ncurses-dev gcc-arm-linux-gnueabihf**

* 1. Install toolchain in Raspberrypi directory you have created by running command

**git clone** [**https://github.com/raspberrypi/tools**](https://github.com/raspberrypi/tools)

* 1. Add the toolchain to your path, for adding enter:

**export CCPREFIX=/home/username/Raspberrypi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/bin/arm-linux-gnueabihf-**

* 1. Then enter:

**${CCPREFIX}gcc -v**

1. Before we can build the kernel we need to configure it. Follow the below steps to grab the Configuration file from existing Raspberry Pi.
   1. **ssh pi@raspberrypi**

**Example: ssh pi@192.168.0.X**

* 1. **sudo zcat /proc/config.gz > config**

If the config.gz file is missing, run the following command to load the module that provides it:

**sudo modprobe configs**

* 1. Exit the ssh session by typing **exit**
  2. Enter **cd /home/<username>/Raspberrypi/linux**
  3. Copy the .config from the Raspberry Pi to the Ubuntu using scp

**sudo scp pi@raspberrypi:config .**

**Example: scp pi@192.168.0.X:config .**

* 1. **mv config .config**

##### Configure Source and Compile Kernel. Run the following commands:

* 1. **cd /home/<username>/Raspberrypi/linux**
  2. **KERNEL=kernel7**
  3. **ARCH=arm CROSS\_COMPILE=${CCPREFIX} make bcm2709\_defconfig**
  4. To Compile the kernel, run

**ARCH=arm CROSS\_COMPILE=${CCPREFIX} make -j4**

After this you will get zImage in /linux/arch/arm/boot

1. To Create modules on the Ubuntu, run the following command:
   1. **ARCH=arm CROSS\_COMPILE=${CCPREFIX} INSTALL\_MOD\_PATH=../modules make modules\_install**
2. Create the new kernel image and copy it to your home folder ~
   1. Run the following command:

**scripts/mkknlimg arch/arm/boot/zImage ~/kernel7.img**

1. Copy the new kernel over to the Raspberry Pi and place it into the /boot directory. Run the following commands:
   1. **ssh pi@192.168.0.X**
   2. **sudo scp pi@ubuntu:kernel7.img /boot/kernel7.img**

Example: **sudo scp username@192.168.0.Y:kernel7.img /boot/kernel7.img**

1. Before copying the modules, first remove the unneeded directories:
   1. **cd /home/username/Raspberrypi/modules/lib/modules/4.9.73-v7+/**
   2. **sudo rm -r build**
   3. **sudo rm -r source**
2. To Copy modules to raspberry pi, run the following commands:
   1. **ssh pi@192.168.0.X**
   2. **sudo rm -f -r /lib/modules**
   3. **sudo rm -f -r /lib/firmware**
   4. **sudo scp username@192.168.0.Y:/home/username/Raspberrypi/modules/lib /**
   5. **sudo scp username@192.168.0.Y:/home/username/Raspberrypi/linux/arch/arm/boot/dts/\*.dtb /boot/**
   6. **sudo scp username@192.168.0.Y:/home/username/Raspberrypi/linux/arch/arm/boot/dts/overlays/\*.dtb\* /boot/overlays/**
3. To restart Raspberry pi, enter **reboot**
4. Once Raspberry PI boots, connect to it over SSH and run the **uname -r** command to see the new kernel