Hello Learners,

There is always fear for new people when they install a new operating system or while installing a new kernel. Today in this article I will try to cover every aspect of kernel Compilation and cross compilation and meaning of each and every step.

Article contains Below Parts

- Environment and tool chain setup for compilation
- New Kernel setup
- Building Kernel
- Installing Kernel
- Loading New Kernel

Purpose behind setting up a new kernel is purely for development and experiment purposes, so if something goes wrong with the experiment and the system doesn't respond, no worry still stable/generic kernel will be there to boot up the machine. There are different versions of Kernel tarball are available on www.kernel.org we need to download a stable version of kernel or whichever suits your requirement.

- Download the Kernel stable version (or any As per Requirement)
  https://www.kernel.org/linux-5.7.13.tax.bz

"Wget '' command helps to get the file from the link for more information you can check linux **man** pages. Download kernel tarball at home Location in any directory. After Downloading Complete extract the File.

- tar xvf linux-5.7.13.tar.xz

"tar" command will help you to extract the .xz compressed file linux-5.7.13 named Folder will be extracted with all source code inside it.

- New Kernel setup

Always check Release Note before downloading kernel or OS code there compiler version and it's tool chain will be mentioned. Mentioned tool chain is needed to build the kernel. For above mentioned kernel GNU gcc compiler tool chain is enough to compile complete kernel. (gcc tool chain version you might need to confirm with release note but latest tool chain always make your work done)

- sudo apt-get install gcc

this will install GNU tool chain on your machine if it is not there. There is no compulsions of GNU tool chain only you can use others too you just need verify with release note and forum.

- Kernel Build

Before building kernel we need to configure the kernel. There are many ways to configure kernel I will suggest below two simple ways

a. Copy Old kernel configuration into current Linux Directory
b. Run Make menuconfig

- make menuconfig

Above command will give you GUI interface to do kernel config. If you want to do default config simply save the configuration and exit so default config will be taken

After extracting kernel tar ball package linux* named directory must be created. We need to go into the directory. Make file will be there. For building complete kernel we need to run make file.

- make –j4

make command will start building the kernel –j is the parameter for creating a number of threads(Here 4) which simultaneously build the kernel. Only make will also work. This takes relatively longer time to build the kernel. All drivers and related files will be compiled and library generated here.

- Kernel Installation

- make modules_install

Above command will install compiled modules and files to your rootfs file system. normally "lib" directory will be created and all kernel needed libraries and dependencies get copied. if you want to install files into some different directory, you change the env variable as INSTALL_MOD_PATH = < Directory Path Here >.

- make install

  This is Final command which will make kernel final Grub entry and in initilise other init script.

upto now we Downloaded new kernel & compiled kernel, all dependent modules are builded. Kernel image and dependent files are copied in the root directory. Now we will be able to see kernel image as below

vmliunz{Kernel version} : This is compressed bootable kernel Image. this will be shown in Grub Menu

initrd.img-{Kernel version}: This image is executed by the kernel to mount the root file system on RAM/prime memory contains the root file system.

system.map-{Kernel version}: This Contains symbolic Map For kernel

If all above images are created in the boot directory or the directory you gave at the time of make module install. then kernel modules installation got successful

→ Follow Below Note For Cross Compilation

Note:

When you want to cross compile kernel image for some other platform please do follow below Modification in compilation steps

1. Make Sure GNU or another suitable tool chain is installed in your system.
2. While building kernel by make command modify as Below
   a. make CROSS_COMPILE={tool chain Path} ARCH={architecture} -j4 all (CROSS_COMPILE= specifies Tool chain Directory path which we are going to use ARCH specifies architecture you are building for example=arm, i386)
3. After Building complete kernel architecture specific Files and library will be generated Now need to get them together in directory so you will be able to copy them on target
   a. make INSTALL_MOD_PATH={directory PATH} modules_Install
4. Make install will be not useful in cross compilation that you need to do on target.
   a. copy kernel image and dependant directory to your boot. kernel image is present at