**Native and Cross Compilation**

1. **Steps for Native Compilation:**

1. Boot the board with default kernel version(5.10.17-v7l+).

[ If your board gets booted with zImage(cross compiled kernel image) i.e. 5.10.52-v7l+ then

On board do the following steps :

pi@raspberrypi ~ $ cd /boot

$ sudo vim config.txt

Comment the line which you have added i.e. #kernel=zImage

and save it.

$ cd ~

$ sudo reboot

Your board gets booted with default kernel version (5.10.17-v7l+) ]

1. pi@raspberrypi ~ $ uname -r
2. First install raspberrypi kernel headers using following command

pi@raspberrypi ~ $ sudo apt install raspberrypi-kernel-headers

1. Create directories as shown down and write hello module.

pi@raspberrypi ~ $ mkdir device-drivers

$ cd device-drivers

$ mkdir cross-compiler

$ mkdir native-compiler

$ cd native-compiler

$ mkdir hello\_module

$ cd hello\_module

$ vim hello.c

Write following code in hello.c file and save it

#include<linux/module.h>

#include<linux/init.h>

static int hello\_init(void)

{

printk("Hello World\n");

return 0;

}

static void hello\_cleanup(void)

{

printk("Good Bye\n");

}

module\_init(hello\_init);

module\_exit(hello\_cleanup);

MODULE\_LICENSE("GPL");

MODULE\_AUTHOR("CDAC");

MODULE\_DESCRIPTION("A simple hello\_world kernel module");

hello\_module $ vim Makefile

Write following instructions in Makefile and save it.

Obj-m += hello.o

KDIR=/lib/modules/$(shell uname -r)/build/

all:

make -C $(KDIR) M=$(PWD) modules

Clean:

Make -C $(KDIR) M=$(PWD) clean

1. Compile and build module :

hello\_module $ make

$ ls

hello.ko file is our module

1. Insert module :

hello\_module $ sudo insmod hello.ko

1. Check for your module :

$ lsmod

1. Check for module initializer function got executed :

$ dmesg

1. Remove module :

$ sudo rmmod hello

1. Check for module cleanup function got executed :

$ dmesg

This is the native compilation - on raspberrypi board.

1. **Steps for cross-compilation:**

On Host(ubuntu) :

1. Install cross-compiled kernel modules onto the rootfs of your host using below commands :

~ $ cd rpi

rpi $ cd linux

rpi/linux $ sudo make ARCH=arm CROSS\_COMPILE=arm-linux-gnueabihf- modules\_install

1. Come back to “rpi” directory and create directories as shown down :

rpi $ mkdir device-drivers

$ cd device-drivers

$ mkdir cross-compiler

$ cd cross-compiler

$ mkdir hello\_module

$ cd hello\_module

$ vim hello.c

Write same code of hello.c file here and save it

hello\_module $ vim Makefile

Write following instructions in Makefile and save it.

Obj-m += hello.o

KDIR=/lib/modules/5.10.52-v7l+/build/

all:

make -C $(KDIR) M=$(PWD) modules

Clean:

Make -C $(KDIR) M=$(PWD) clean

1. Cross-Compile and build module :

hello\_module $ make ARCH=arm CROSS\_COMPILE=arm-linux-gnueabihf-

$ ls

hello.ko file is our module

1. Boot your raspberrypi board with our cross-compiled kernel version(5.10.52-v7l+)

[ If your board gets booted with default kernel version i.e. 5.10.17-v7l+ then

On board do the following steps :

pi@raspberrypi ~ $ cd /boot

$ sudo vim config.txt

Un comment the line which you have added i.e. kernel=zImage

and save it.

$ cd ~

$ sudo reboot

Your board gets booted with our cross-compiled kernel version (5.10.52-v7l+)

Check this with uname -r command ]

1. You should be opening two terminals one is your pc(host ubuntu) terminal and another is of your raspberrypi board terminal.

In your host(ubuntu) pc terminal :

Copy cross-compiled hello.ko file from host PC(ubuntu) to raspberrypi using below command

hello\_module $ scp pi@<your-board-ip>:/home/pi/device-drivers/cross-compiler/

In your raspberrypi board terminal :

pi@raspberrypi ~ $ cd device-drivers

$ cd cross-compiler

$ ls

You will notice hello.ko only file is copied

1. Now Insert module :

pi@raspberrypi ~/device-drivers/Cross-compiler $ sudo insmod hello.ko

1. Check for your module :

$ lsmod

1. Check for module initializer function got executed :

$ dmesg

1. Remove module :

$ sudo rmmod hello

1. Check for module cleanup function got executed :

$ dmesg