Name :- Amit.Y.Zope
Roll No. :- 223067
Gr No. :- 21810714
Division :- C
Batch :- C3
Subject :- Operating System

Assignment No 7

Aim: Linux Kernel configuration,compilation , and rebooting from newly compiled kernel. Add your own system call to the kernel.

Objective : To Study the Linux kernel configuration and to add a system to Linux kernel.

Theory:

Linux Kernel

- The main purpose of a computer is to run a *predefined sequence of instructions*, known as a **program**. A program under execution is often referred to as a **process**.

- Now, most special purpose computers are meant to run a single process, but in a sophisticated system such a general purpose computer, are intended to run many processes simulteneously.

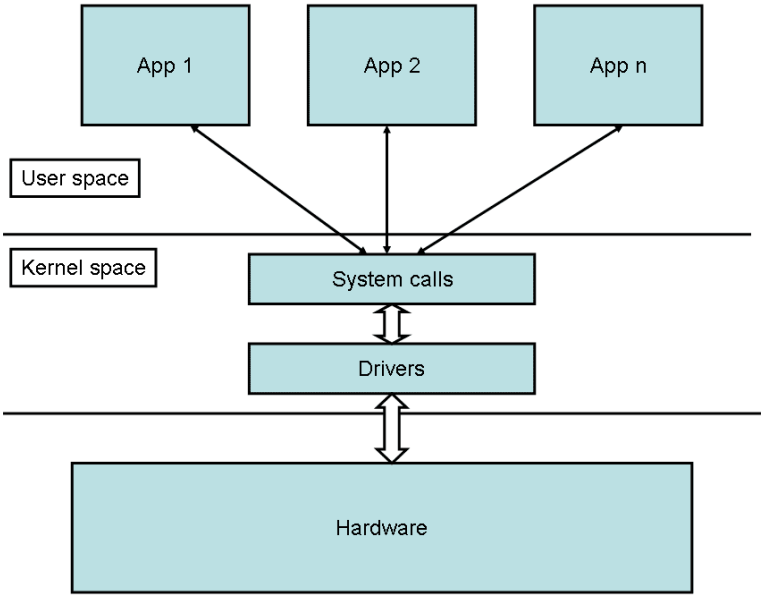
- Any kind of process requires hardware resources such are Memory, Processor time, Storage space, etc.

- In a General Purpose Computer running many processes simulteneously, we need a middle layer to manage the distribution of the hardware resources of the computer efficiently and fairly among all the various processes running on the computer. This middle layer is referred to as the **kernel**.

- Basically the kernel virtualizes the common hardware resources of the computer to provide each process with its own virtual resources. This makes the process seem as it is the sole process running on the machine.

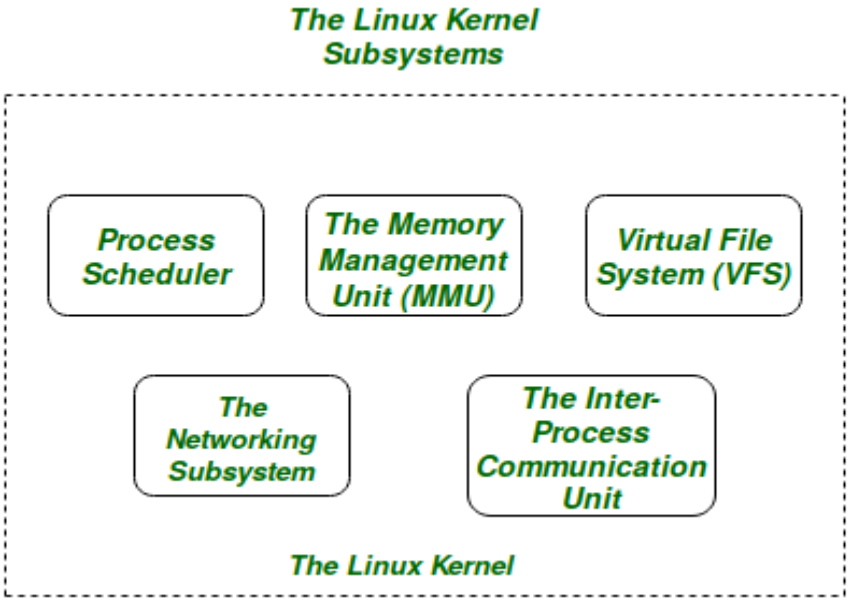
- The kernel is also responsible for preventing and mitigating conflicts between different processes.

The following schematic gives an idea abour Kernel Space :



The **Core Subsystems** of the **Linux Kernel** are as follows:

1. The Process Scheduler
2. The Memory Management Unit (MMU)
3. The Virtual File System (VFS)
4. The Networking Unit
5. Inter-Process Communication Unit



The basic functioning of each of the 1st three subsystems is elaborated below:

- **The Process Scheduler:**
  This kernel subsystem is responsible for fairly distributing the CPU time among all the processes running on the system simulteneously.

- **The Memory Management Unit:**
  This kernel sub-unit is responsible for proper distribution of the memory resources among the various processes running on the system. The MMU does more than just simply provide separate virtual address spaces for each of the processes.

- **The Virtual File System:**
  This subsystem is responsible for providing a unified interface to access stored data across different filesystems and physical storage media.

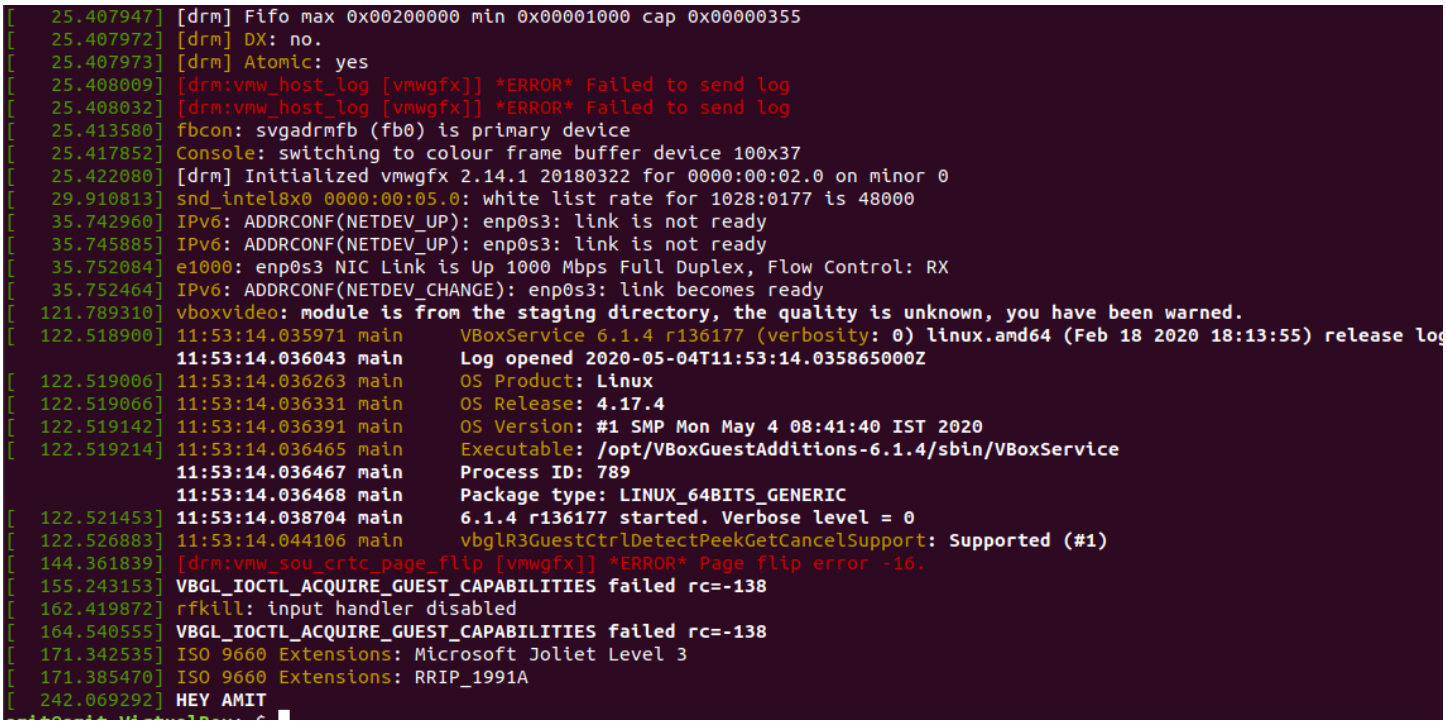
OUTPUTS FOR THIS ASSIGNMENT :

```
amit@amit-VirtualBox:~$ gedit test_sys_amit.c
amit@amit-VirtualBox:~$ gcc test_sys_amit.c
amit@amit-VirtualBox:~$ ./a.out


        System call sys_amit returned value  : 0


amit@amit-VirtualBox:~$
```

```
[  122.518900] 11:53:14.035971 main     VBoxService 6.1.4 r136177 (verbosity: 0)
 linux.amd64 (Feb 18 2020 18:13:55) release log
               11:53:14.036043 main     Log opened 2020-05-04T11:53:14.035865000
Z
[  122.519006] 11:53:14.036263 main     OS Product: Linux
[  122.519066] 11:53:14.036331 main     OS Release: 4.17.4
[  122.519142] 11:53:14.036391 main     OS Version: #1 SMP Mon May 4 08:41:40 IS
T 2020
[  122.519214] 11:53:14.036465 main     Executable: /opt/VBoxGuestAdditions-6.1.
4/sbin/VBoxService
               11:53:14.036467 main     Process ID: 789
               11:53:14.036468 main     Package type: LINUX_64BITS_GENERIC
[  122.521453] 11:53:14.038704 main     6.1.4 r136177 started. Verbose level = 0
[  122.526883] 11:53:14.044106 main     vbglR3GuestCtrlDetectPeekGetCancelSuppor
t: Supported (#1)
[  144.361839] [drm:vmw_sou_crtc_page_flip [vmwgfx]] *ERROR* Page flip error -16
.
[  155.243153] VBGL_IOCTL_ACQUIRE_GUEST_CAPABILITIES failed rc=-138
[  162.419872] rfkill: input handler disabled
[  164.540555] VBGL_IOCTL_ACQUIRE_GUEST_CAPABILITIES failed rc=-138
[  171.342535] ISO 9660 Extensions: Microsoft Joliet Level 3
[  171.385470] ISO 9660 Extensions: RRIP_1991A
[  242.069292] HEY AMIT _
```



Adding a New System Call to Linux Kernel

NOTE : KERNEL VERSION USED 4.17.4

System call added : sys_amit();

1. Switching to root to perform priviledged operations.

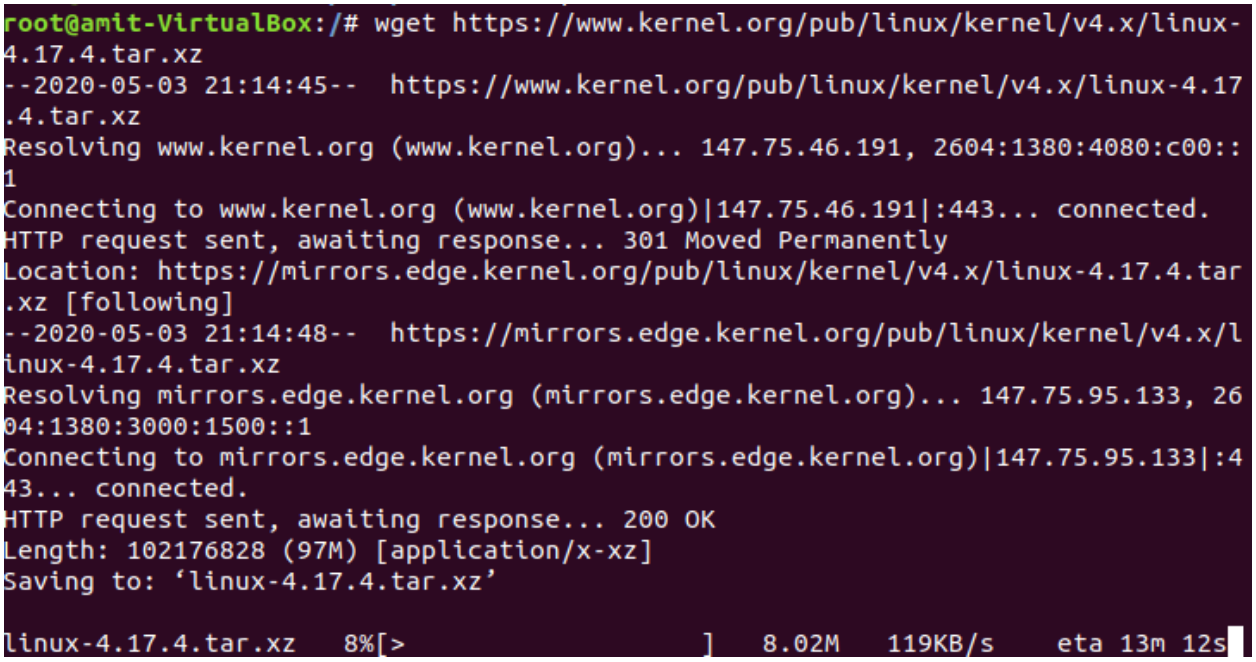
   Using ‘`sudo su`’.

2. Download the Linux Kernel

Open the terminal and use the following command to download the kernel course file.

`wget` https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.17.4.tar.xz

wget command : GNU Wget is a free utility for non-interactive download of files from the Web.



3. Extract the Kernel Source Code.

Extract the kernel source code from the linux-4.17.4.tar.xz file in /usr/src/ directory using the following command. Since the downloaded tar file will be in Downloads folder, use cd to change into Downloads folder before executing the below command.

```
sudo tar -xvf linux-4.17.4.tar.xz -C/usr/src/
```

sudo – to gain root access.

tar – Tar stores and extracts files from a tape or disk archive.

-x – extract files from an archive

-v – requested using the –verbose option, when extracting archives

-f – –file archive; use archive file or device archive

-C, –directory DIR,change to directory DIR(here to change to /usr/src/)

Now after extraction change to the kernel source directory using,

cd /usr/src/linux-4.17.4/

4. Define a New System Call sys_amit()

1.Create a directory amit in the kernel source directory:-

mkdir amit

Change into this directory

cd amit

2. Create a “amit.c” file in this folder and add the definition of the system call to it as given
below (you can use any text editor).

```
root@amit-VirtualBox:/usr/src/linux-4.17.4/amit# gedit Makefile

** (gedit:26659): WARNING **: 08:34:11.115: Set document metadata failed: Settin
g attribute metadata::gedit-spell-language not supported

** (gedit:26659): WARNING **: 08:34:11.117: Set document metadata failed: Settin
g attribute metadata::gedit-encoding not supported

** (gedit:26659): WARNING **: 08:34:13.117: Set document metadata failed: Settin
g attribute metadata::gedit-position not supported
```

gedit amit.c

```
root@amit-VirtualBox:/usr/src/linux-4.17.4# mkdir amit
root@amit-VirtualBox:/usr/src/linux-4.17.4# cd amit
root@amit-VirtualBox:/usr/src/linux-4.17.4/amit# gedit amit.c

** (gedit:26645): WARNING **: 08:33:44.231: Set document metadata failed: Settin
g attribute metadata::gedit-spell-language not supported

** (gedit:26645): WARNING **: 08:33:44.232: Set document metadata failed: Settin
g attribute metadata::gedit-encoding not supported

** (gedit:26645): WARNING **: 08:33:45.737: Set document metadata failed: Settin
g attribute metadata::gedit-position not supported
```

Add the following code:-

```
#include <linux/kernel.h>

asmlinkage long sys_amit(void)
{
    printk("HEY AMIT\n");
    return 0;
}
```

Note that printk prints to the kernel's log file.

2. Create a "Makefile" in the amit folder and add the given line to it.

```
gedit Makefile
```

add the following line to it:-

```
obj-y := amit.o
```

This is to ensure that the amit.c file is compiled and included in the kernel source code.

5. Add the amit directory to Kernel's make file

Change back into the linux-4.17.4 folder and open Makefile

```
gedit Makefile
```

Search for "**core-y**" in editor

Then change this "*core-y += kernel/ mm/ fs/ ipc/ security/ crypto/ block/* "

to "*core-y += kernel/ mm/ fs/ ipc/ security/ crypto/ block/* ***amit/***"

This is to tell the compiler that the source files of our new system call (sys_amit()) are in present in the amit directory.

```
ifeq ($(KBUILD_EXTMOD),)
core-y          += kernel/ certs/ mm/ fs/ ipc/ security/ crypto/ block/ amit/

vmlinux-dirs    := $(patsubst %/,%,$(filter %/, $(init-y) $(init-m) \
                     $(core-y) $(core-m) $(drivers-y) $(drivers-m) \
                     $(net-y) $(net-m) $(libs-y) $(libs-m) $(virt-y)))
```

6. Add the new system call (sys_amit()) into the system call table (syscall_64.tbl file)

If your system is a 64 bit system you will need to alter the syscall_64.tbl file.

cd arch/x86/syscalls

gedit syscall_64.tbl

add the following line in the end of the file :-

548 64 amit sys_amit

```
539     x32     process_vm_readv        __x32_compat_sys_process_vm_readv
540     x32     process_vm_writev       __x32_compat_sys_process_vm_writev
541     x32     setsockopt              __x32_compat_sys_setsockopt
542     x32     getsockopt              __x32_compat_sys_getsockopt
543     x32     io_setup                __x32_compat_sys_io_setup
544     x32     io_submit               __x32_compat_sys_io_submit
545     x32     execveat                __x32_compat_sys_execveat/ptregs
546     x32     preadv2                 __x32_compat_sys_preadv64v2
547     x32     pwritev2                __x32_compat_sys_pwritev64v2
548     64      amit                    sys_amit
```

7. Add the new system call(sys_amit()) in the system call header file.

```
cd  include/linux/
```

```
gedit syscalls.h
```

add the following line to the end of the file just before the #endif statement at the very bottom.

```
asmlinkage long sys_amit(void);
```

This defines the prototype of the function of our system call. “asmlinkage” is a key word used to indicate that all parameters of the function would be available on the stack.

```
extern long do_sys_truncate(const char __user *pathname, loff_t length);

static inline long ksys_truncate(const char __user *pathname, loff_t length)
{
        return do_sys_truncate(pathname, length);
}
asmlinkage long sys_amit(void);


#endif
```

8. Compile the Kernel On Your System.

Install Essential Packages

Hence do the following in the terminal.

```
1.    sudo apt-get install gcc
2.    sudo apt-get install libncurses5-dev
3.    sudo apt-get install bison
4.    sudo apt-get install flex
5.    sudo apt-get install libssl-dev
6.    sudo apt-get install libelf-dev
7.    sudo apt-get update
```

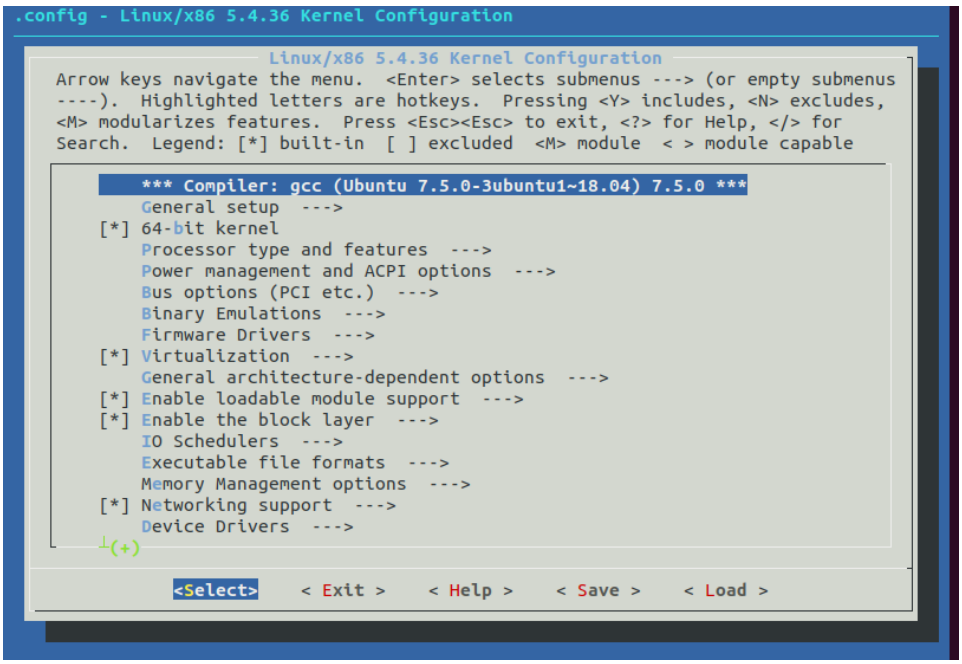
8. *sudo apt-get upgrade*

=> To configure your kernel use the following command:-

```
sudo make menuconfig
```

```
root@amit-VirtualBox:/usr/src/linux-4.17.4# sudo make menuconfig
  HOSTCC  scripts/basic/fixdep
  HOSTCC  scripts/kconfig/mconf.o
  YACC    scripts/kconfig/zconf.tab.c
  LEX     scripts/kconfig/zconf.lex.c
  HOSTCC  scripts/kconfig/zconf.tab.o
  HOSTCC  scripts/kconfig/lxdialog/checklist.o
  HOSTCC  scripts/kconfig/lxdialog/util.o
  HOSTCC  scripts/kconfig/lxdialog/inputbox.o
  HOSTCC  scripts/kconfig/lxdialog/textbox.o
  HOSTCC  scripts/kconfig/lxdialog/yesno.o
  HOSTCC  scripts/kconfig/lxdialog/menubox.o
  HOSTLD  scripts/kconfig/mconf
scripts/kconfig/mconf  Kconfig
#
# using defaults found in /boot/config-5.0.0-23-generic
#
/boot/config-5.0.0-23-generic:5518:warning: symbol value 'm' invalid for RADIO_S
I470X
```

Select ext4 from pop up window save and exit



.

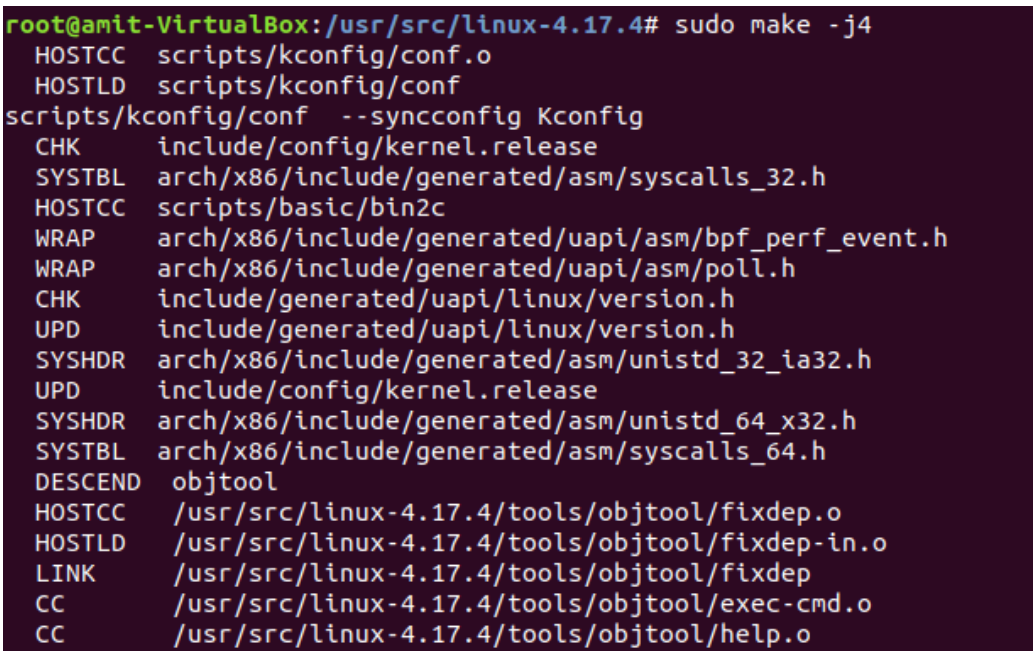
If you like to have your existing configuration then run the below command.

```
sudo make oldconfig
```

Now to compile the kernel ; do make .

cd /usr/src/linux-4.17.4/

```
sudo make
```



This might take several hours depending on your system. It can take 2-3 hours to get this compiled.

9. Install/Update Kernel Now.

To install this edited kernel run the following command:-

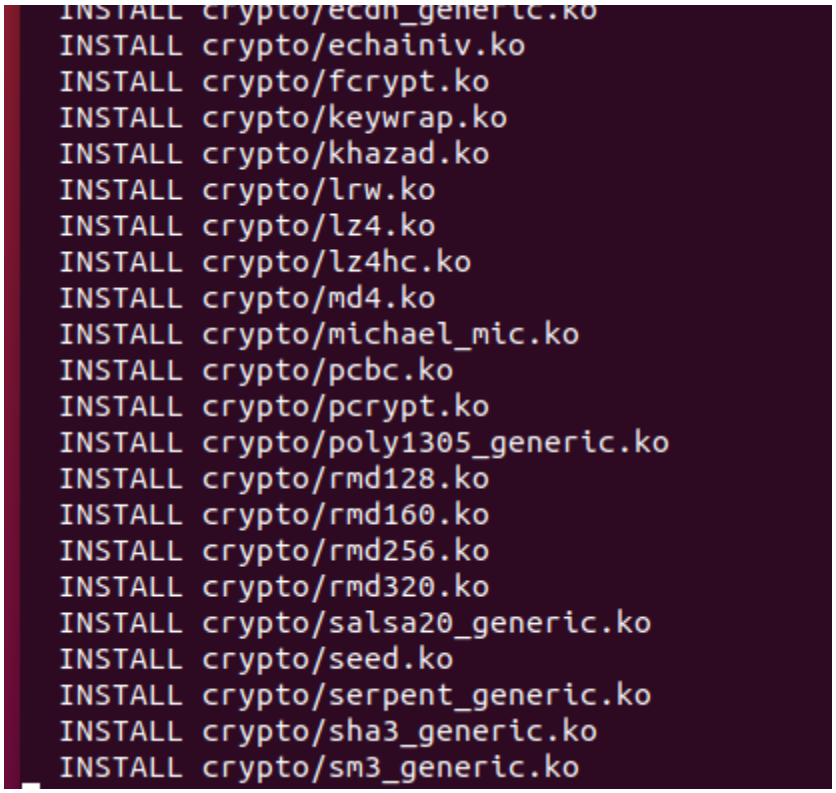
```
sudo make modules_install install
```

Now to update the kernel in your system reboot the system . You can use the following command.

```
shutdown -r now
```

After rebooting you can verify the kernel version using the following command;

```
uname -r
```



10. Testing System_Call

Create a “test_sys_amit.c” program in your home folder and type in the following code :-

```
#include<linux/kernel.h>
#include<stdio.h>
#include<sys/syscall.h>
#include<unistd.h>

int main(){

        long int return_value = syscall(548);
        printf("\n\n\tSystem call sys_amit returned value  : %ld \n\n\n",return_value);
        return 0;
}
```

Now compile this program using the following command.

gcc test_sys_amit.c

If all goes well you will not have any errors else, rectify the errors.

Now run the program using the following command.

./a.out

You will see the following line getting printed in the terminal if all the steps were followed
correctly.

"*System call sys_amit returned value 0*".

Now to check the message of the kernel you can run the following command.
dmesg
This will display "*HEY AMIT*" at the end of the kernel's message.