**Name: Sudipta Halder**

**Roll: 2021202011**

**OS ASSIGNMENT 2 PDF**

Step 1: First, I have downloaded all necessary dependent libraries.

sudo apt install –y build-essential flex bison libssl-dev

Step 2: update

sudo apt-get update && sudo apt-get upgrade

Step 3: get the kernel 4.9.210

wget <https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.9.210.tar.xz>

Step 4: extract the kernel

xz –v –d linux-4.9.210.tar.xz

tar xvf linux-4.9.210.tar

Step 5: Change the directory

cd linux-4.9.210

Step 6: add new system calls in the syscall\_64.tbl

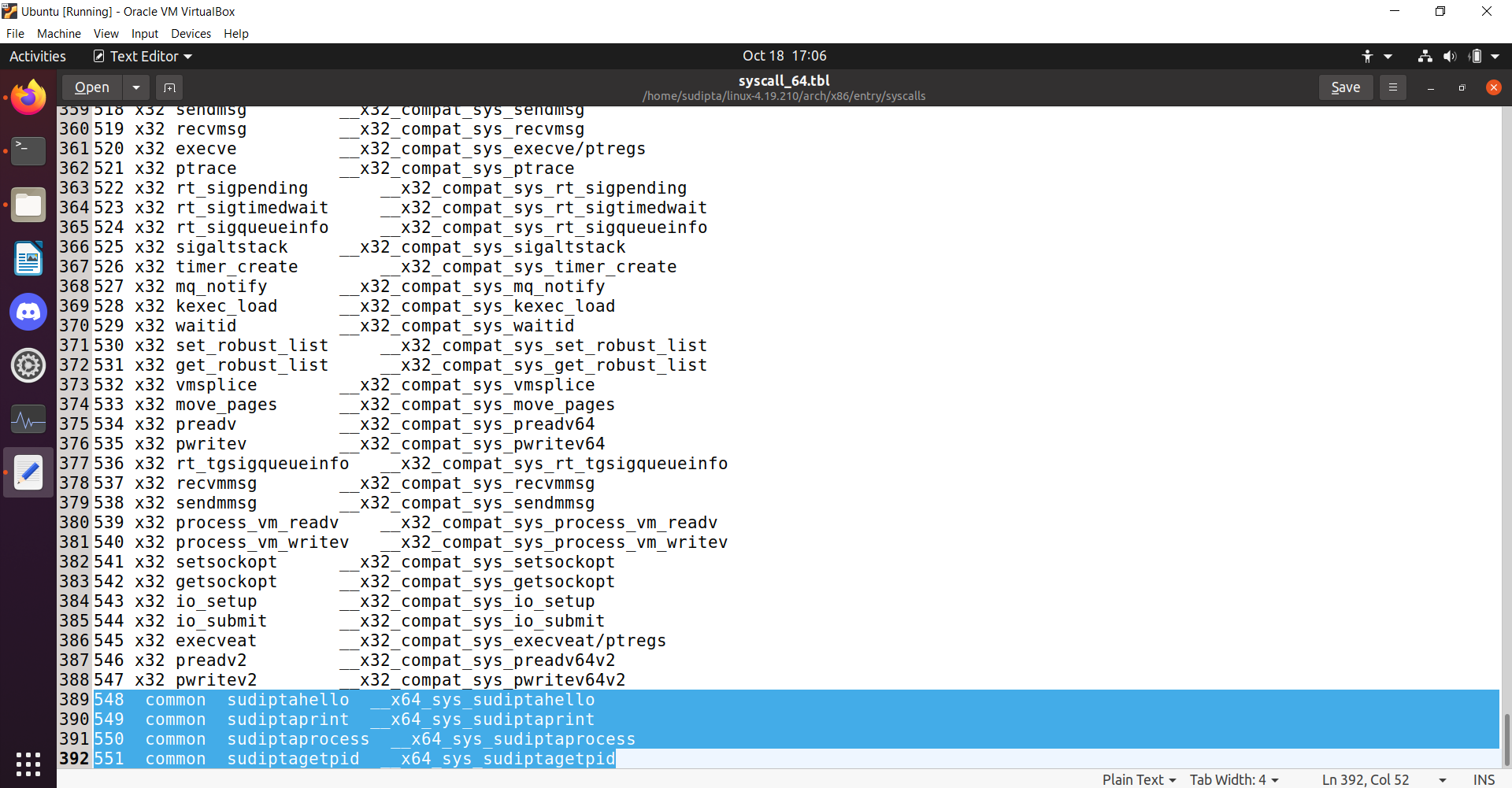
sudo gedit arch/x86/entry/syscalls/syscall\_64.tbl

548 common sudiptahello \_\_x64\_sys\_sudiptahello

549 common sudiptaprint \_\_x64\_sys\_sudiptaprint

550 common sudiptaprocess \_\_x64\_sys\_ sudiptaprocess

551 common sudiptagetpid \_\_x64\_sys\_ sudiptagetpid



Step 7: add new system calls to the system call header file

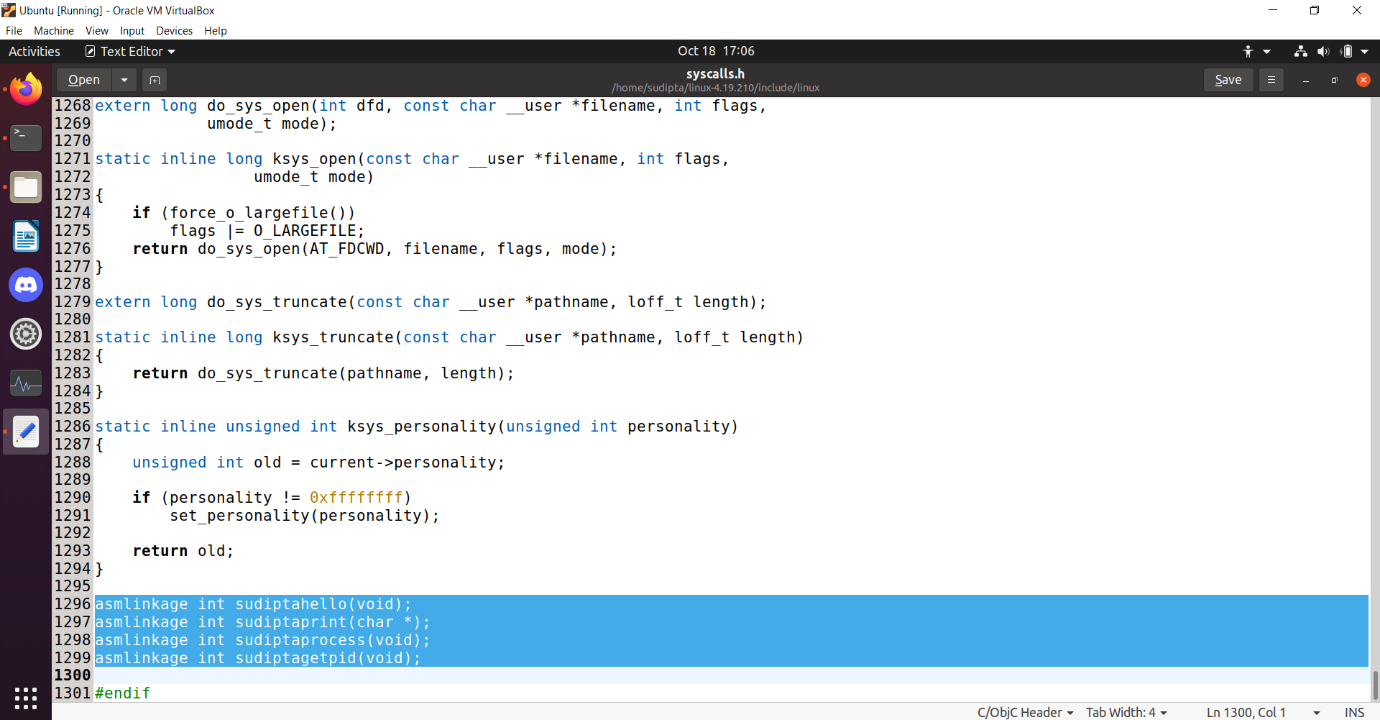
sudo gedit include/linux/syscalls.h

asmlinkage int sudiptahello(void);

asmlinkage int sudiptaprint(char \*);

asmlinkage int sudiptaprocess(void);

asmlinkage int sudiptagetpid(void);



Step 8: add four .c files in /kernel

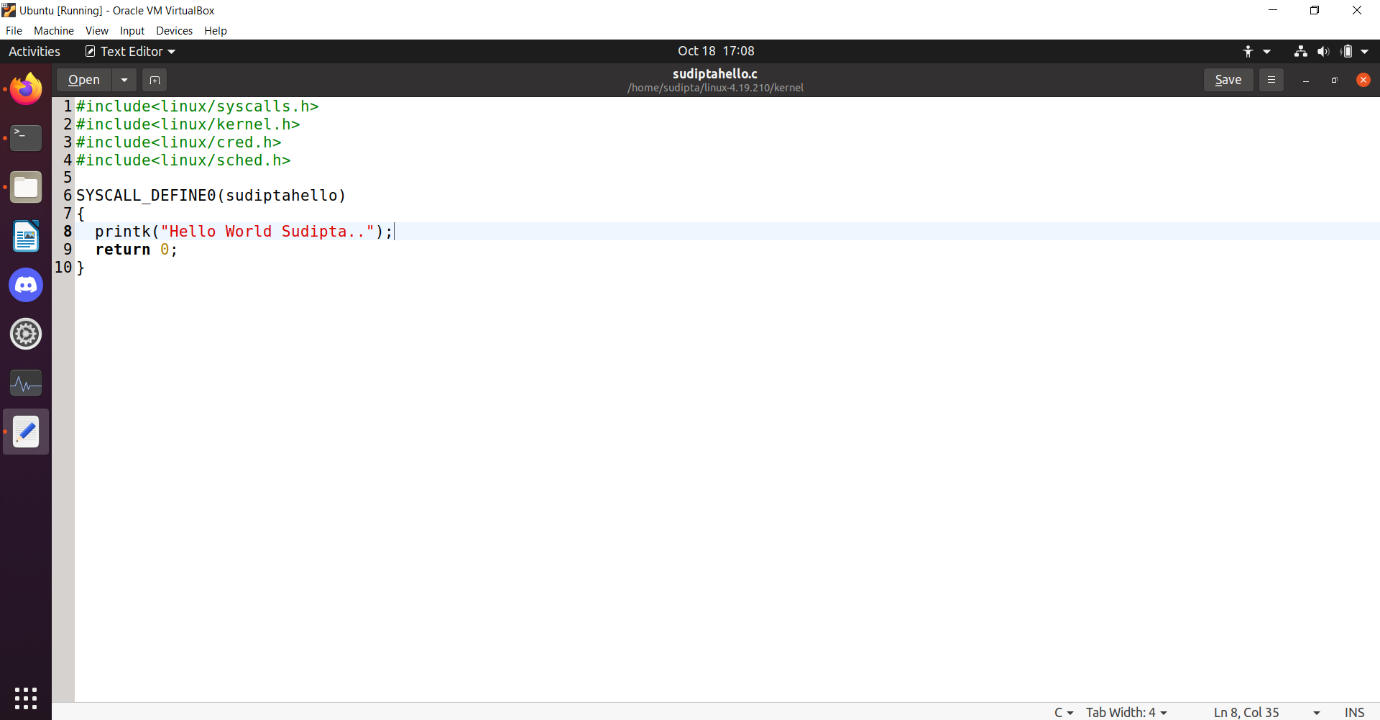
cd kernel

sudo gedit kernel/sudiptahello.c

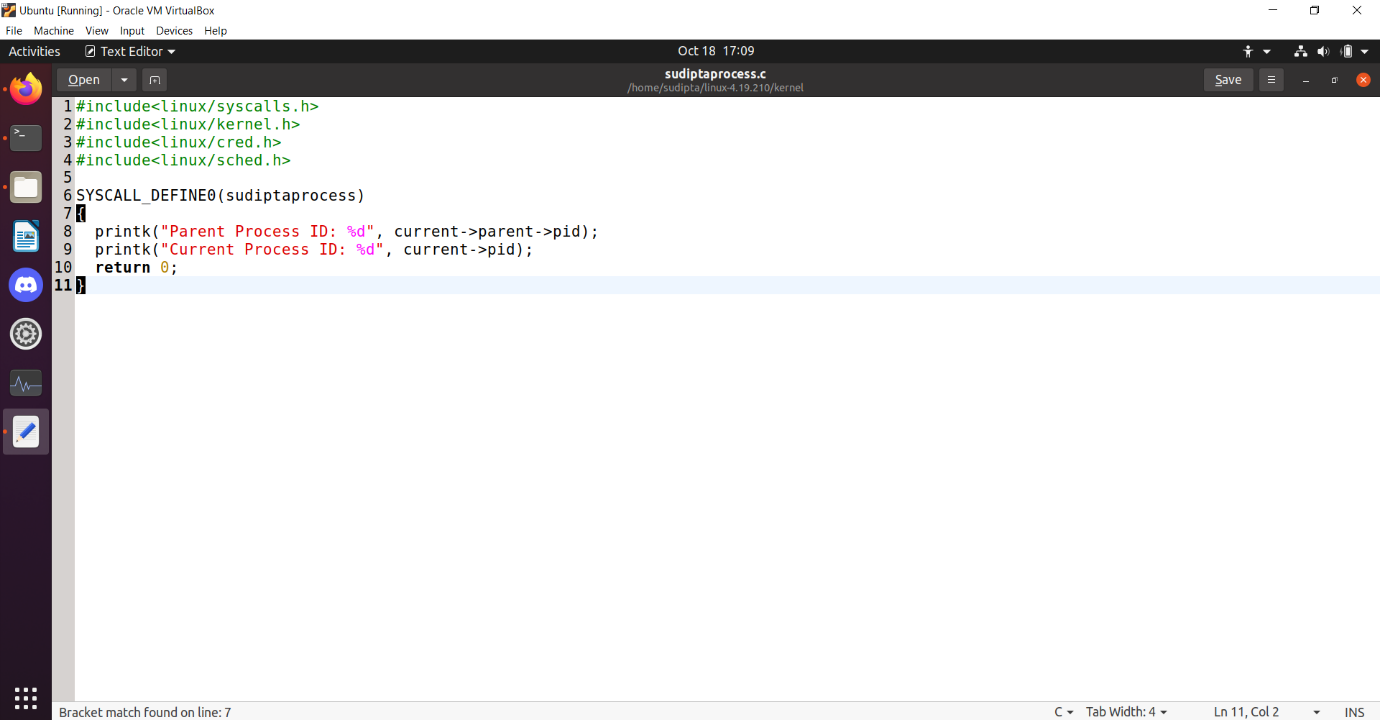
sudo gedit kernel/sudiptaprint.c

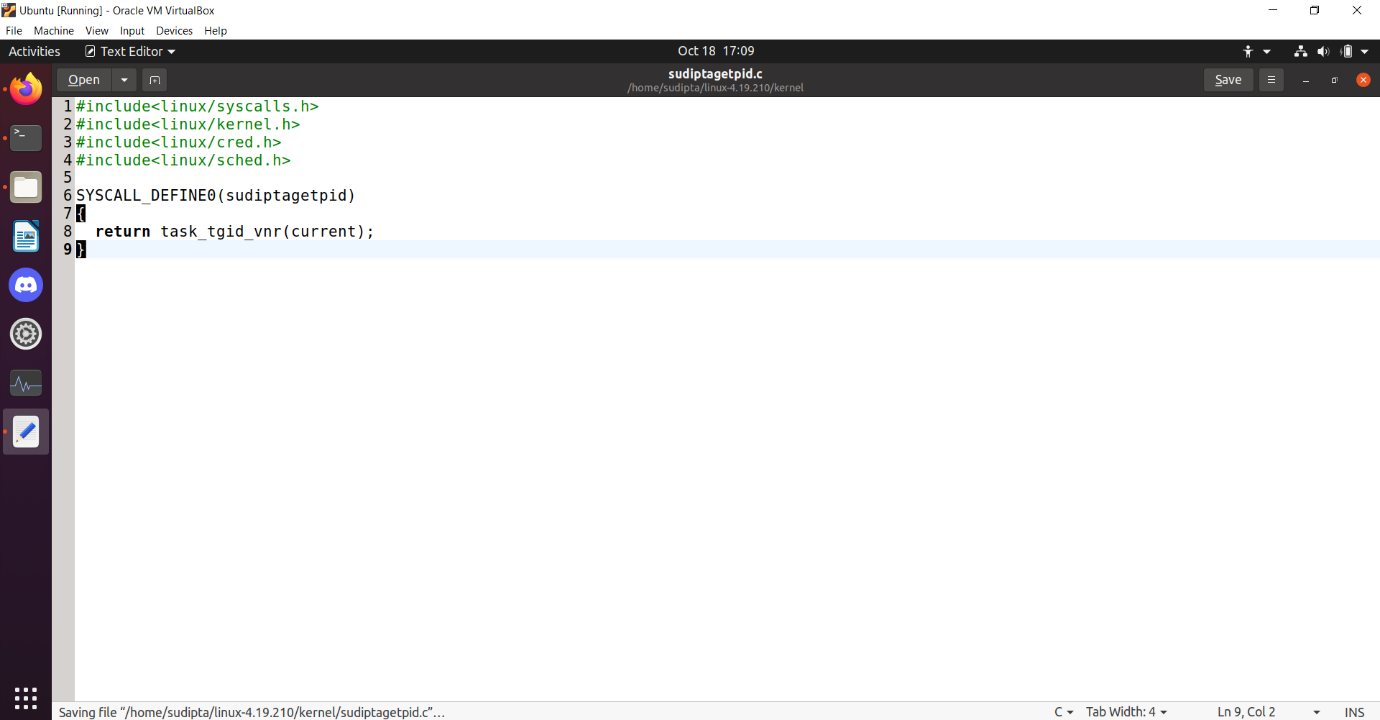
sudo gedit kernel/sudiptaprocess.c

sudo gedit kernel/sudiptagetpid.c









The codes are written in zip folder

Step 9: cp –v/boot/config-$(uname-r).config

Step 10: sudo gedit .config

Perform the below operation:

CONFIG\_SYSTEM\_TRUSTED\_KEYS=””;

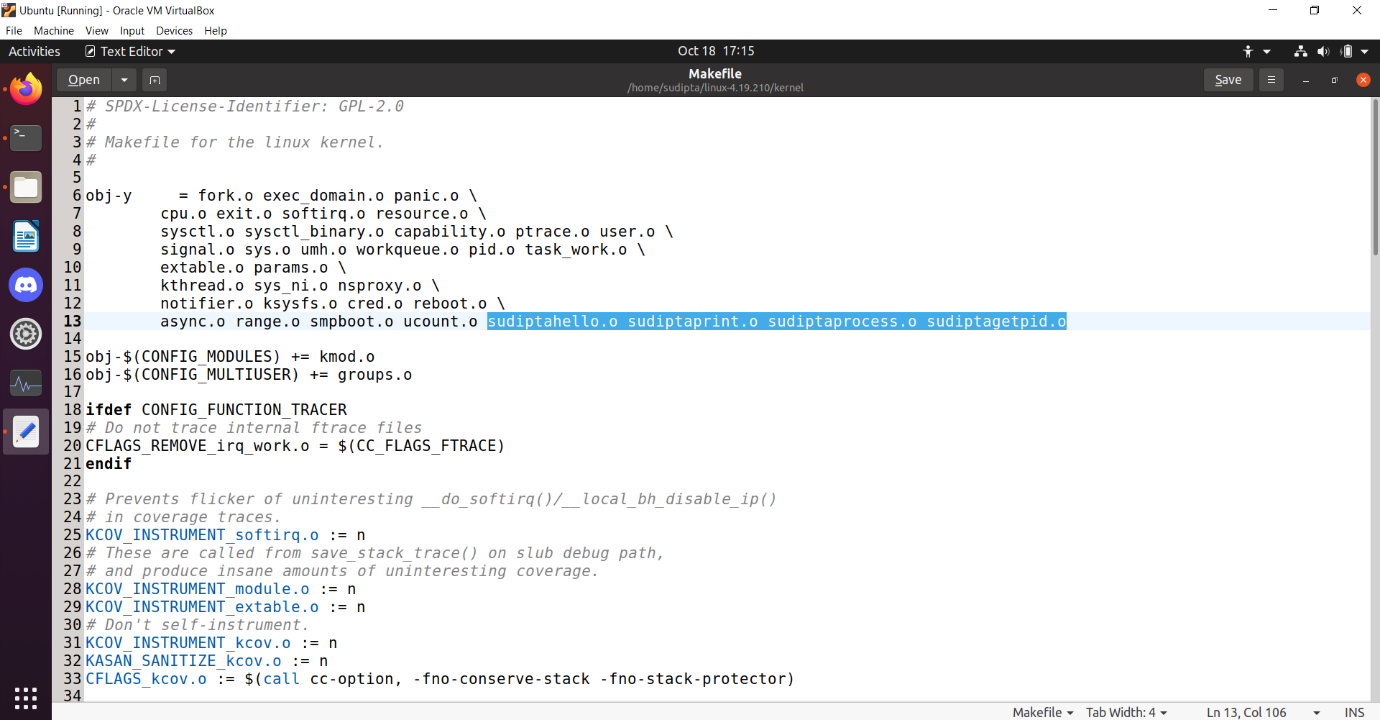
Step 11: sudo make olddefconfig

Step 12: change in makefile in kernel

sudo gedit kernel/Makefile

Append sudiptahello.o sudiptaprint.o sudiptaprocess.o sudiptagetpid.o at the end of obj-y :=

This is to ensure that the .c files are compiled and included in the kernel source code.



Step 13: sudo make prepare

Step 14: sudo make –j4

Step 15: sudo make –j4 modules\_install

Step 16: sudo make install

Step 17: sudo reebot

Step 18: Then make a main.c file to test program

#include<stdio.h>

#include<string.h>

int main()

{

int res;

res = syscall(548);

printf("SYSCALL1 : %d\n", res);

res = syscall(549, "HI SUDIPTA HALDER!!");

printf("SYSCALL2 : %d\n", res);

res = syscall(550);

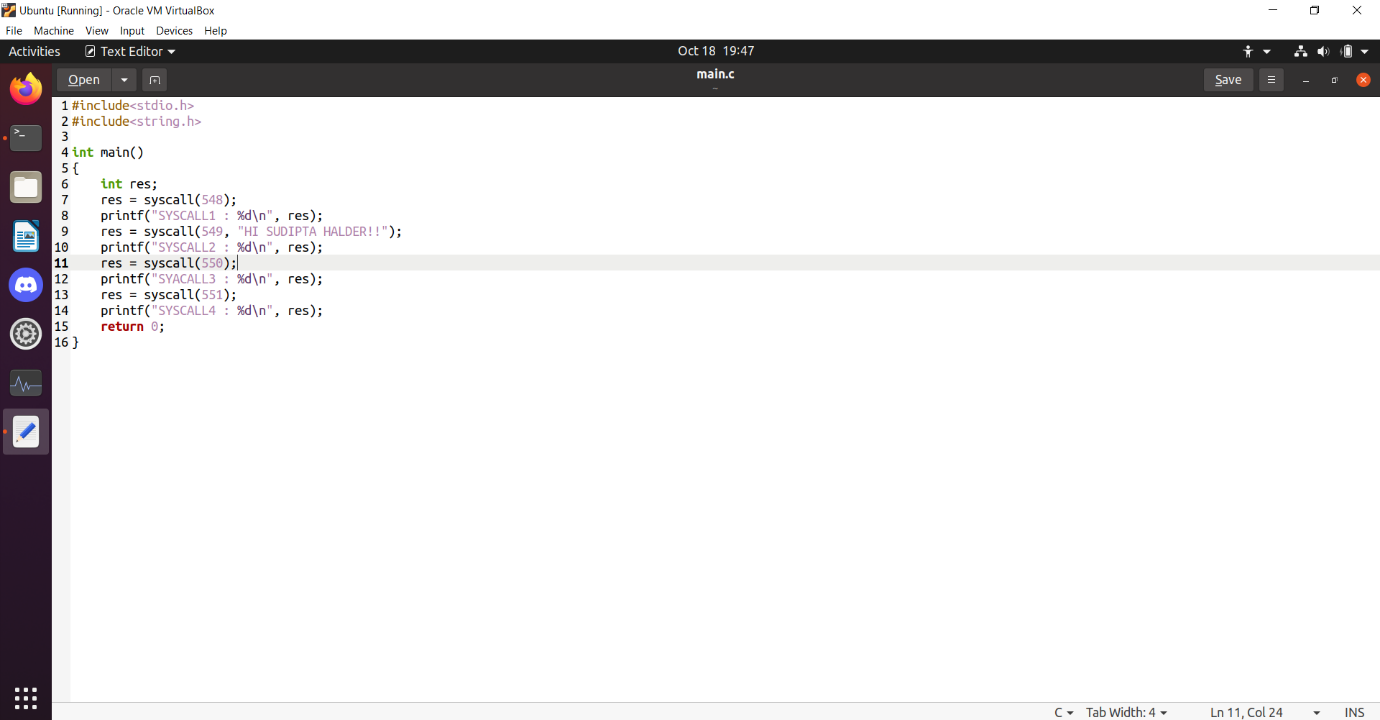
printf("SYSCALL3 : %d\n", res);

res = syscall(551);

printf("SYSCALL4 : %d\n", res);

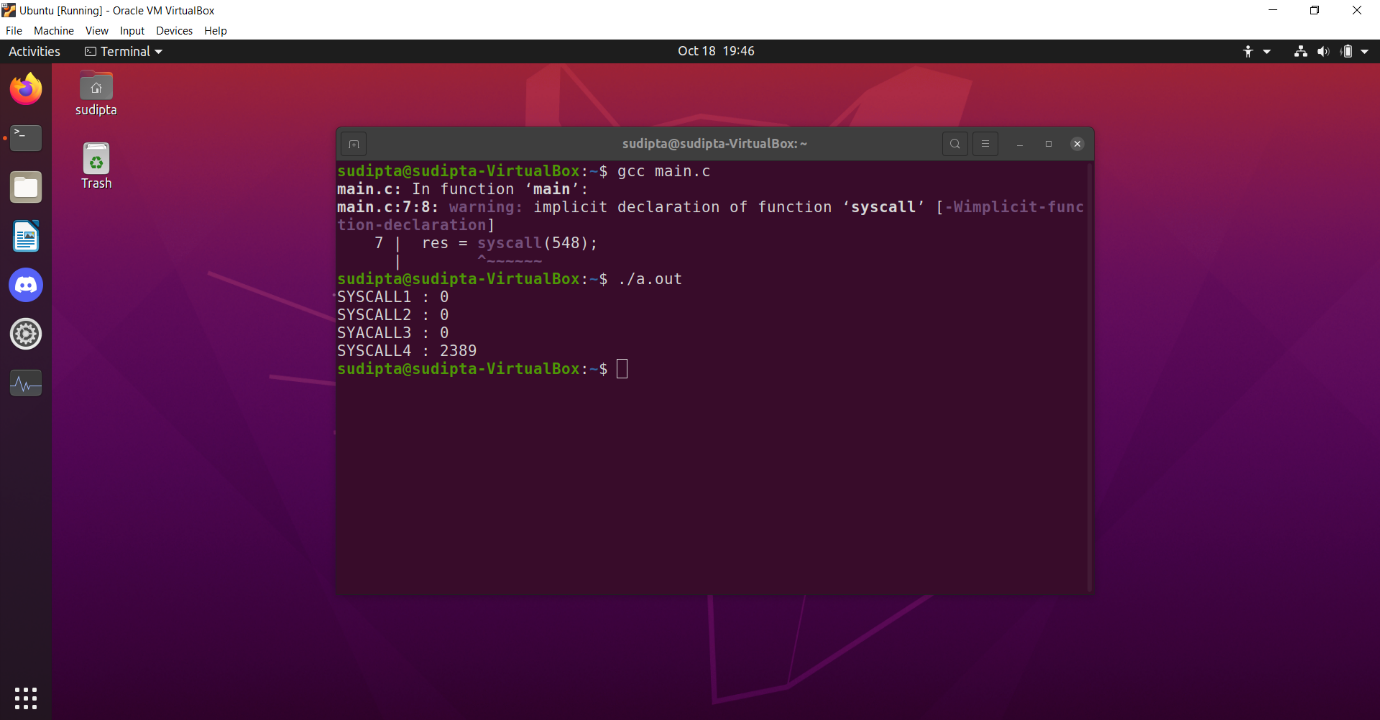
return 0;

}



Compile the program: gcc main.c

Execute the program: ./a.out



Step 19: type in terminal ‘dmesg’ to check the output written in kernel.

The output is visible in the screenshot below.

