**Name: Muhammad Hamza Aslam**

**ID: F2017266041**

**Add Birthday Module in Linux**

Step1: create directory

mkdir birthday

Step2: compiler kernel header

sudo apt-get install linux-header-$(uname -r)

Step3:Go to birthday directory

cd birthday

Step4:edit or create Makefile

gedit Makefile

Step5:enter this code

obj-m = birthday.o

KVERSION = $(shell uname -r)

all:

make -C /lib/modules/$(KVERSION)/build M=$(PWD) modules

clean:

make -C /lib/modules/$(KVERSION)/build M=$(PWD) clean

Note: there is space of 1 line between gedit obj-m KVERSION ….

After enter by typing “all” press ENTER then TAB and then write make command

Step7: make modules

make clean

make all

Step8: write birthday module

gedit birthday.c

Code:

#include <linux/init.h>

#include <linux/module.h>

#include <linux/kernel.h>

#include <linux/list.h>

#include <linux/slab.h>

struct birthday {

int day;

int month;

int year;

struct list\_head list;

};

struct list\_head birthday\_list;

struct birthday \*createBirthday(int day, int month, int year) {

struct birthday \*person = kmalloc(sizeof(struct birthday), GFP\_KERNEL);

person->day = day;

person->month = month;

person->year = year;

return person;

}

void printInfo(char \*str) {

printk(KERN\_INFO "OS Module: %s", str);

}

/\* This function is called when the module is loaded. \*/

int simple\_init(void)

{

printInfo("Loading Module\n");

LIST\_HEAD(birthday\_list);

struct birthday \*person = createBirthday(13, 4, 1987);

list\_add\_tail(&person->list, &birthday\_list);

person = createBirthday(14, 1, 1964);

list\_add\_tail(&person->list, &birthday\_list);

person = createBirthday(2, 6, 1964);

list\_add\_tail(&person->list, &birthday\_list);

person = createBirthday(13, 8, 1986);

list\_add\_tail(&person->list, &birthday\_list);

person = createBirthday(10, 6, 1990);

list\_add\_tail(&person->list, &birthday\_list);

struct birthday \*ptr;

list\_for\_each\_entry(ptr, &birthday\_list, list) {

printk(KERN\_INFO "OS Module: Day %d.%d.%d\n", ptr->day, ptr->month, ptr->year);

}

return 0;

}

/\* This function is called when the module is removed. \*/

void simple\_exit(void) {

printInfo("Removing Module\n");

struct birthday \*tmp;

struct list\_head \*ptr, \*next;

if (list\_empty(&birthday\_list)) {

printInfo("List is empty");

return;

}

list\_for\_each\_safe(ptr, next, &birthday\_list){

tmp = list\_entry(ptr, struct birthday, list);

printk(KERN\_INFO "OS Module: Removing %d.%d.%d\n", tmp->day, tmp->month, tmp->year);

list\_del(ptr);

kfree(tmp);

}

//list\_for\_each\_entry\_safe(ptr, next, &birthday\_list, list) {

// printk(KERN\_INFO "OS Module: Removing %d.%d.%d\n", ptr->day, ptr->month, ptr->year);

// list\_del(&ptr->list);

// kfree(ptr);

//}

printInfo("Module removed\n");

}

/\* Macros for registering module entry and exit points. \*/

module\_init( simple\_init );

module\_exit( simple\_exit );

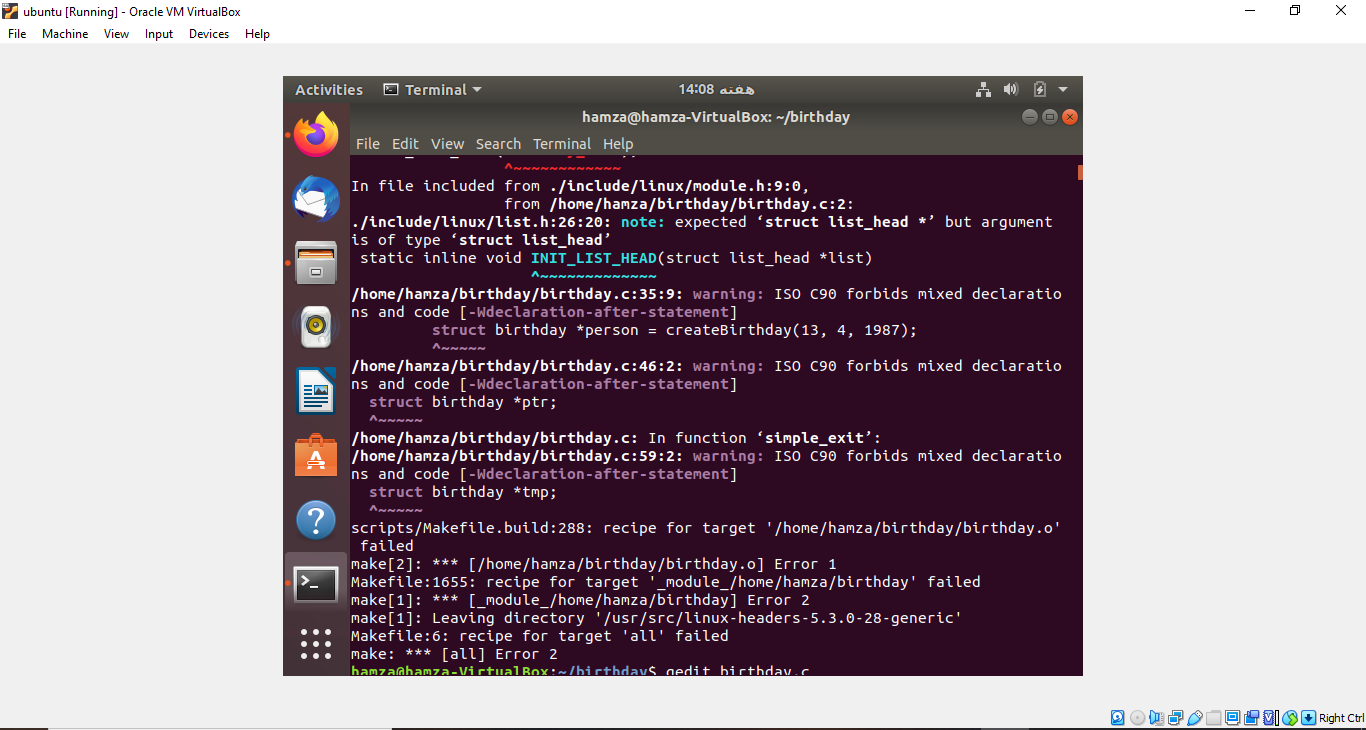
MODULE\_DESCRIPTION("Add Birthday Module");

MODULE\_AUTHOR("Hamza Aslam");

Step9: now we make module

Make

Note: if you get error after make command like this



Go to cd /usr/src/linux-headers-5.3.0-28-generic/include/linux

Note: I have kernel 5.3.0-28-generic

To see kernel type uname -r and whatever the kernel you have write that number

After going to that directory search LIST\_HEAD fucntion and add semi colon like this #deine LIST\_HEAD(name); \

Note: there are two fucnctin change in second fucntin which has struct list\_head in it

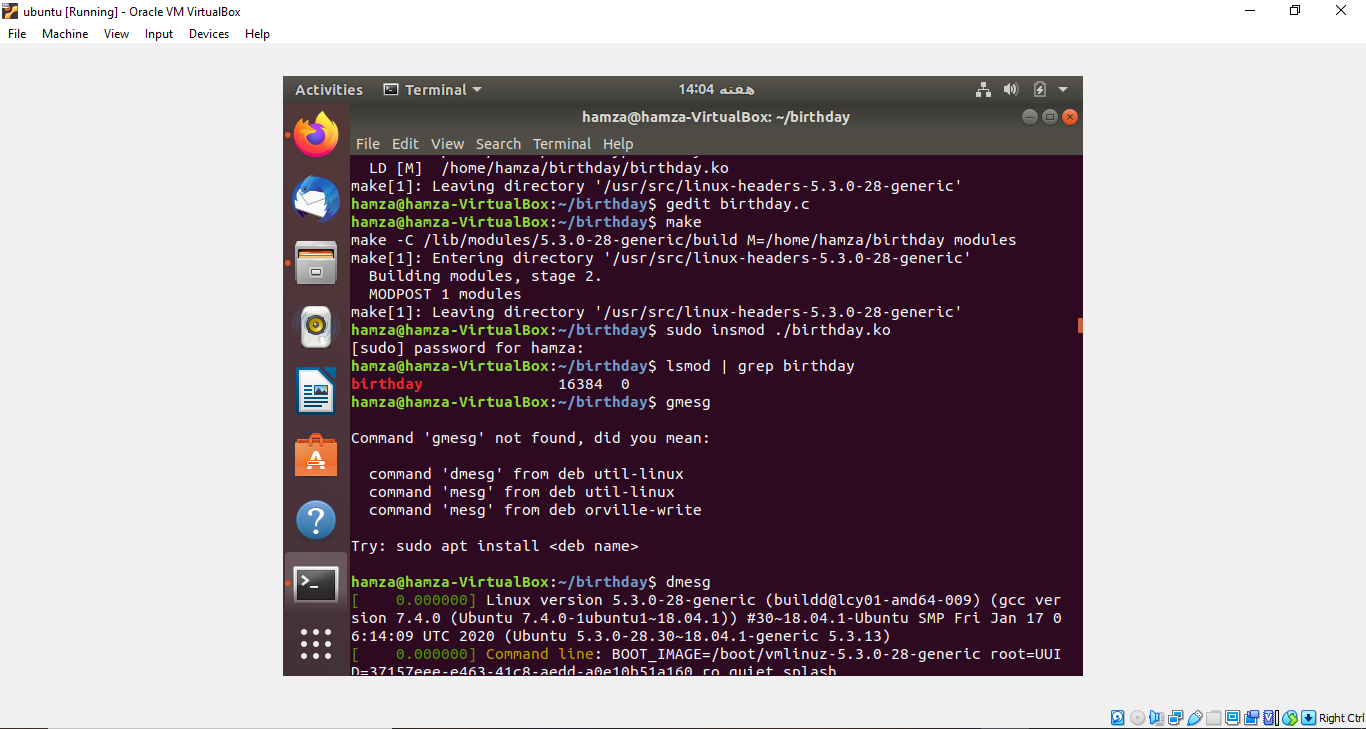
After that go to parent birthday directory

step10: insert module

sudo insmod ./birthday.ko

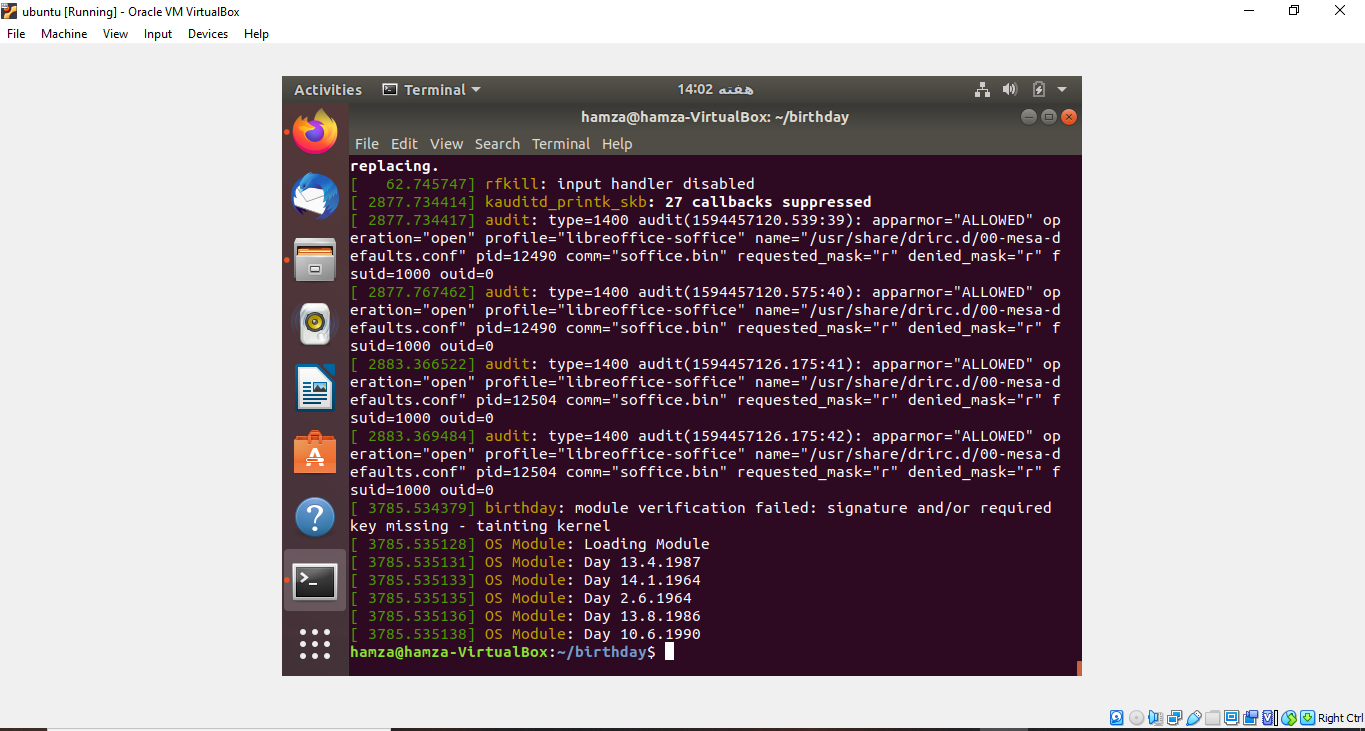
Step11: check status of module

lsmod | grep birthday



Step12:see kernel message how our module work

dmesg

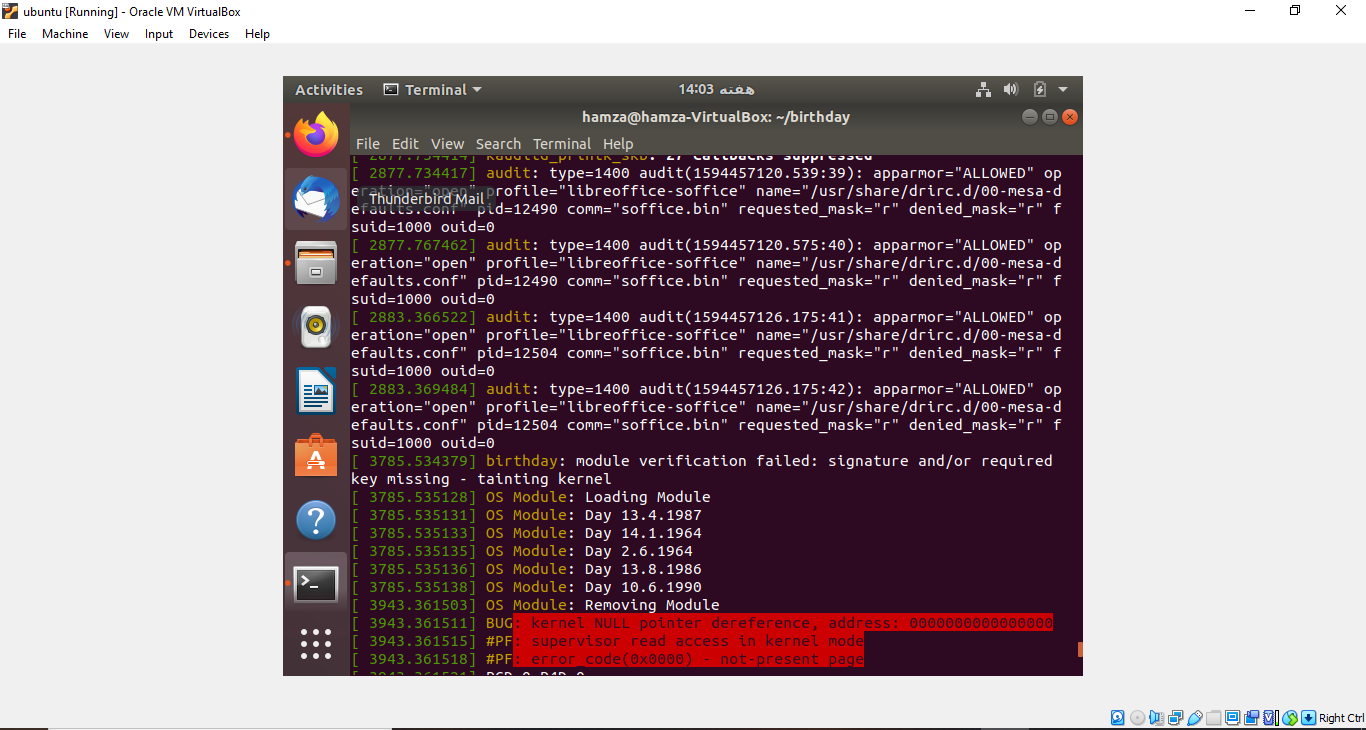


Step13:remove module after successfully added

sudo rmmod birthday

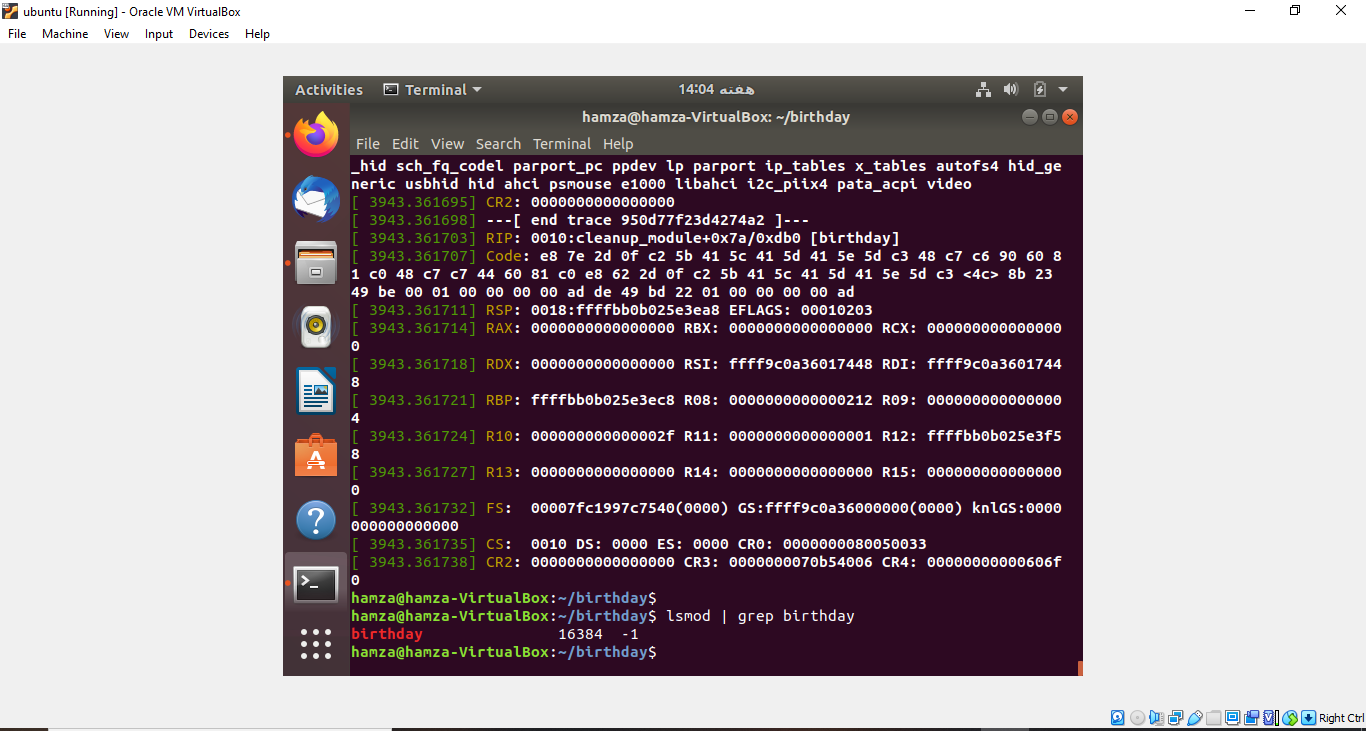
Step14:see kernel message how our module work

dmesg



Step15:check status of module

lsmod | grep birthday



**Helper : Afaq tahir**