

آزمایش 2

40013045

فاطمه راق

9931043

حامد فلاحي

خلاصه

در این گزارش، ما به ساخت و بارگذاری یک ماژول هسته لینوکس ساده پرداخته ایم. ابتدا کد های ماژول را توضیح داده و سپس مراحل توسعه، کامپایل، و اجرای ماژول هسته را بررسی کرده ایم. در نهایت، تحلیلی کوتاه از اهمیت ماژول های هسته و نکاتی که در اجرای این تمرین مهم بودند ارائه شده است.

(HW1)

در تمرین 1، برای ایجاد یک ماژول هسته، ابتدا کد مورد نظر را در یک برنامه تایپ C می اندازیم. سپس با استفاده از دستور make آن را کامپایل میکنیم... اما یک مشکل اساسی که وجود دارد، نصب و اپدیت نبودن Headers ها میباشد. پس از نصب، نیاز است از راه دیگری برویم و کامپایل فایل مورد نظر را به یک فایل جدیدی به اسم Makefile بسپاریم. پس از تولید آن، فایل را کنار آن میندازیم و با ترمینال به ادرس این فایل ها رفته و از انجا با دستور make کامپایل را انجام میدیم و میبینیم چندین فایل برای ما تولید میکند:

```
GNU nano 5.9 simple.c
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/module.h>

int simple_init(void)
{
    printk(KERN_INFO "LOADING Module\n");
    return 0;
}

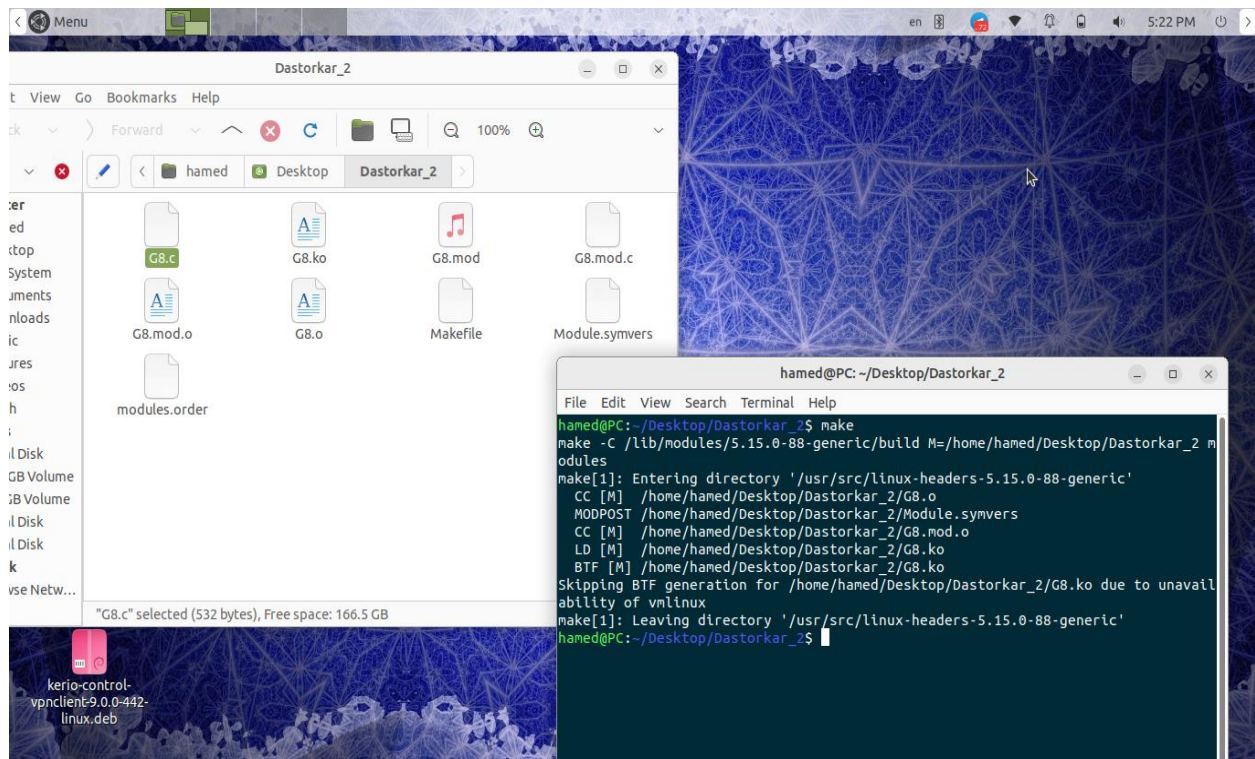
void simple_exit(void)
{
    printk(KERN_INFO "Removing Module\n");
}

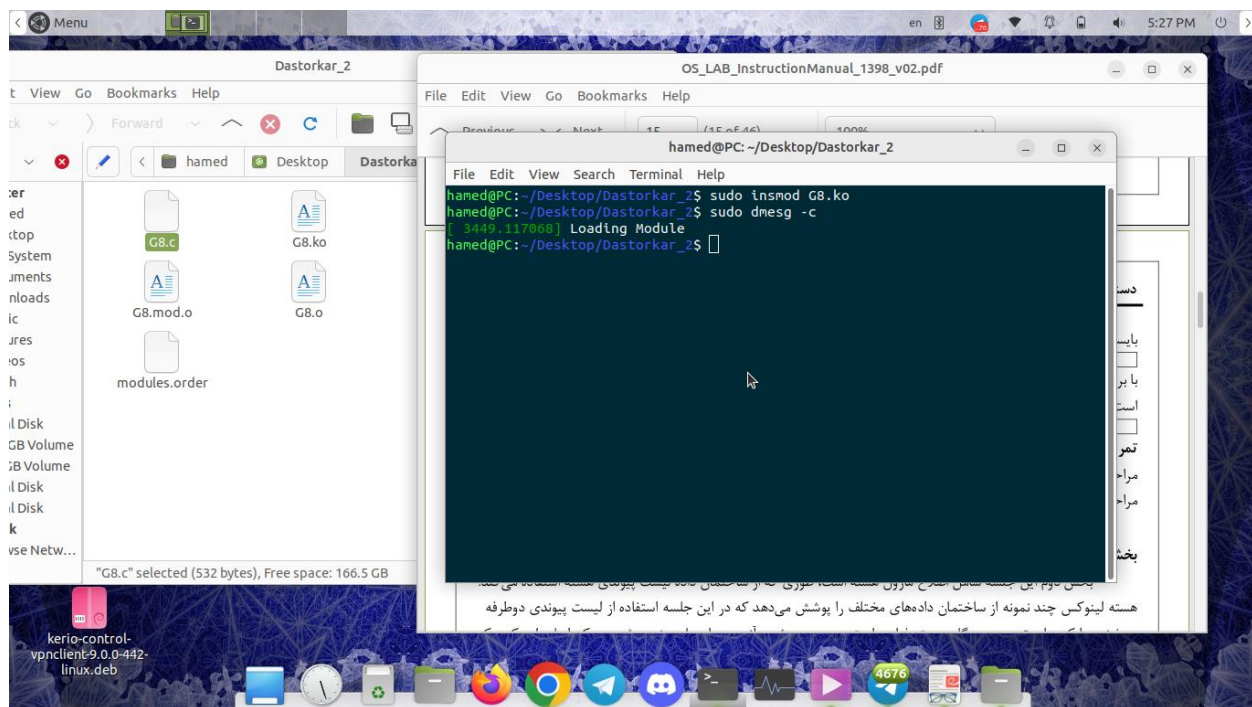
module_init(simple_init);
module_exit(simple_exit);
MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("simple module");
MODULE_AUTHOR("SGG");
```

```
GNU nano 5.9 Makefile
obj-m += simple.o

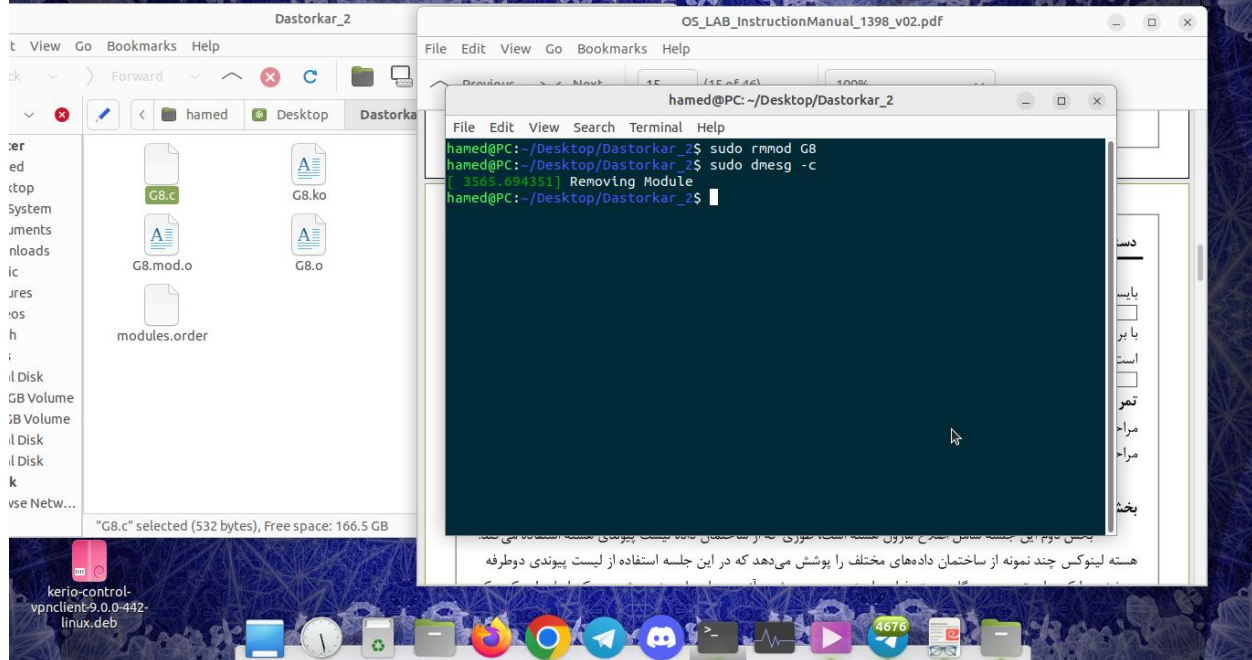
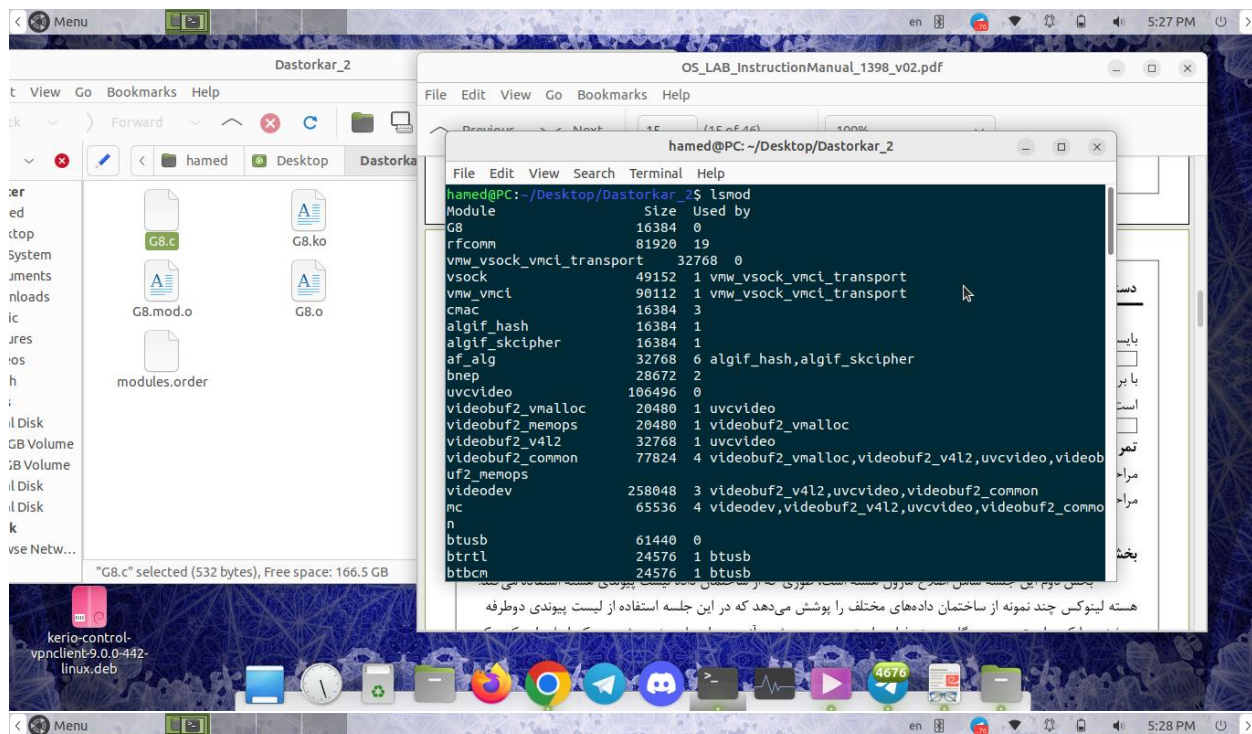
all:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

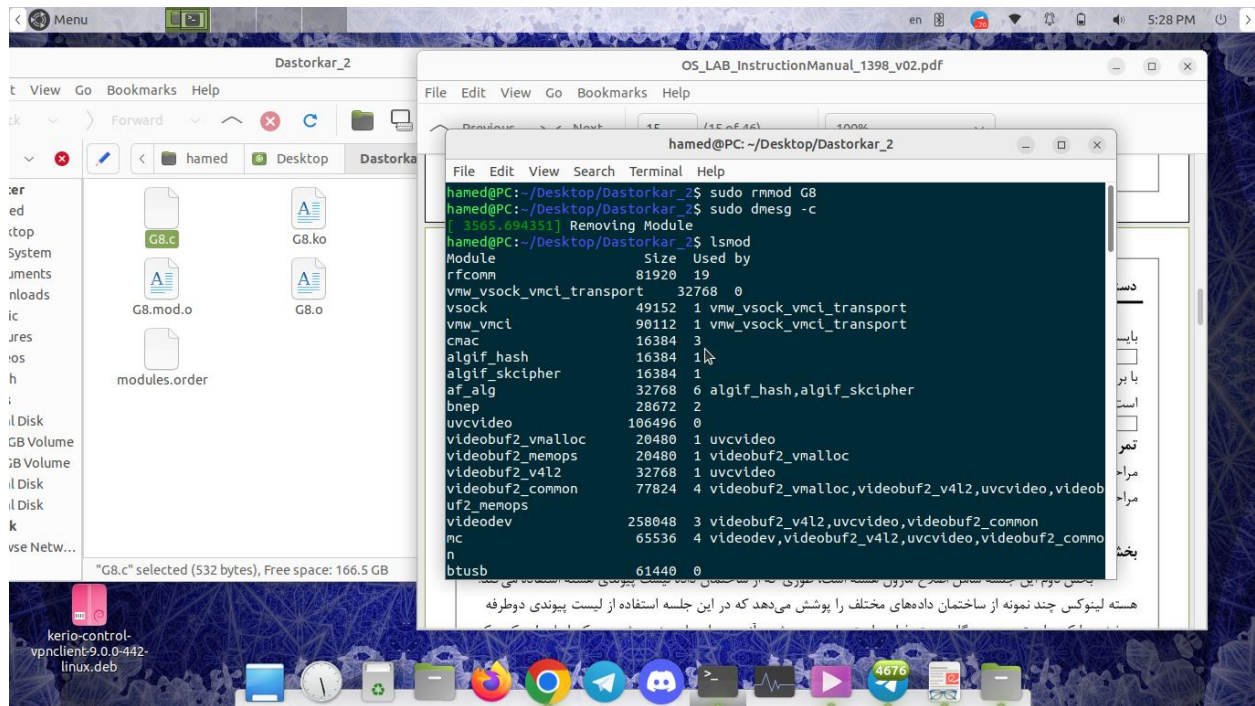




سپس با استفاده از دستور `insmod` ماژول را نصب میکنیم و با دستور `dmesg` پیغام لودینگ را دریافت میکنیم. همینطور با استفاده از دستور `rmmod` ماژول را پاک میکنیم و دوباره با دستور `dmesg` پیغام پاک شدن را دریافت میکنیم: (میبینیم که ماژول **G8** اضافه شده است)

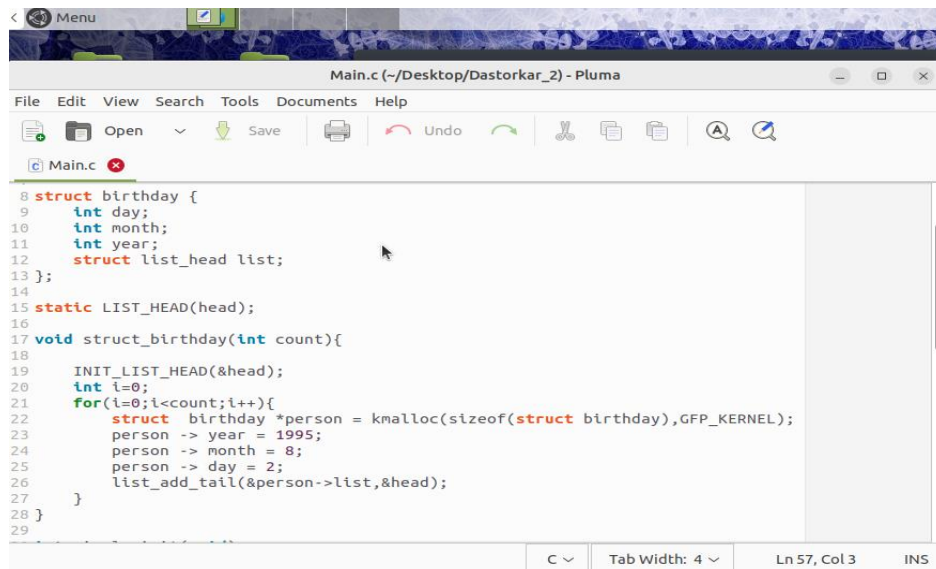


در اینجا میبینیم که ماژول G8 حذف شده است:



(HW2

در تمرین 2 باید کد تمرین 1 را با استفاده از ویژگی های زبان C از جمله ساختمان لیست پیوندی یا لینکد لیست و امالوگ و کالوگ و قابلیت پوینتر و... که در توضیحات دستور کار نیز آمده اصلاح و طراحی کنیم که به ما با ورودی های دستور کار و به تعداد 5 تا خط، سال و ماه و روز تولد را به اعداد گفته شده وارد کنیم:



Main.c (~/Desktop/Dastorkar_2) - Pluma

File Edit View Search Tools Documents Help

Open Save Undo

Main.c

```
24     person -> month = 8;
25     person -> day = 2;
26     list_add_tail(&person->list,&head);
27 }
28 }
29
30 int simple_init(void)
31 {
32     printk(KERN_INFO "Loading Module\n");
33     struct_birthday(5);
34     struct birthday *curr;
35     list_for_each_entry(curr,&head,list){
36         printk(KERN_INFO "add : %d %d %d\n",curr->year,curr->month,curr->day);
37     }
38
39     return 0;
40 }
41 /* this function is called when the module is removed*/
42 void simple_exit(void)
43 {
44     printk(KERN_INFO "Removing Module\n");
45
46     struct list_head *ptr,*next;
```

C Tab Width: 4 Ln 27, Col 6 INS

Main.c (~/Desktop/Dastorkar_2) - Pluma

File Edit View Search Tools Documents Help

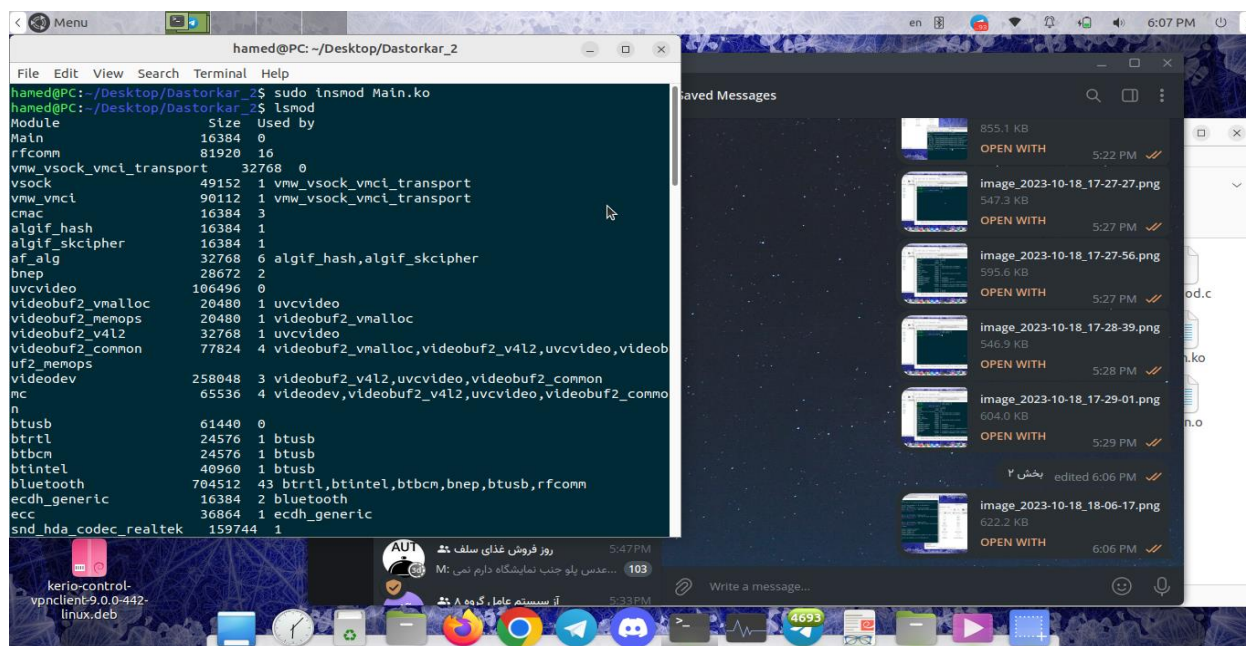
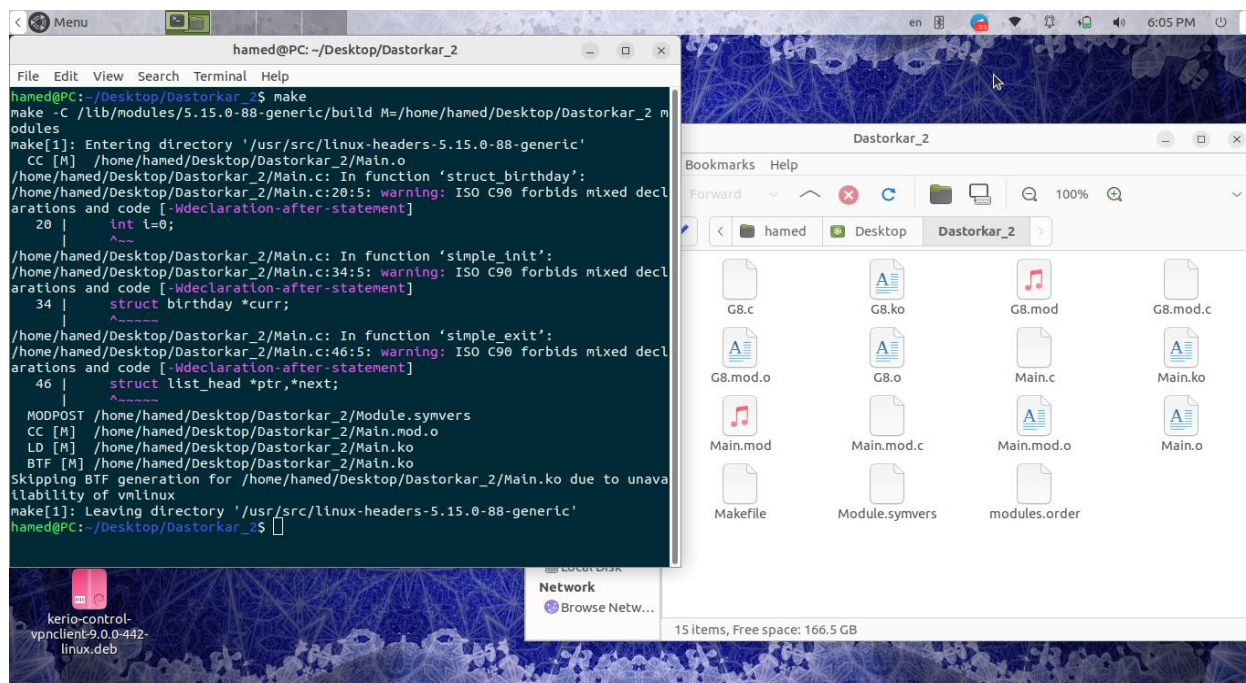
Open Save Undo

Main.c

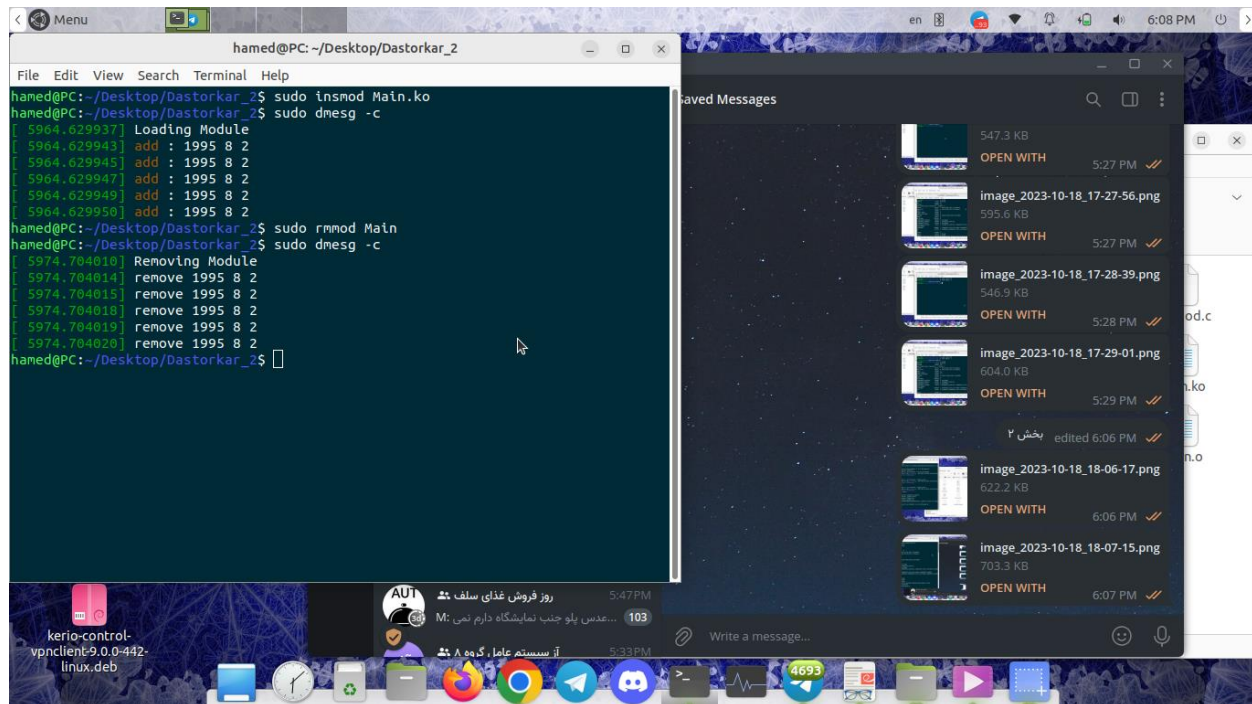
```
40 }
41 /* this function is called when the module is removed*/
42 void simple_exit(void)
43 {
44     printk(KERN_INFO "Removing Module\n");
45
46     struct list_head *ptr,*next;
47     struct birthday *temp;
48
49     list_for_each_safe(ptr,next,&head){
50         temp = list_entry(ptr,struct birthday,list);
51         printk(KERN_INFO "remove %d %d %d\n",temp->year,temp->month,temp->day);
52         list_del(ptr);
53         kfree(temp);
54     }
55 }
56 /* Macros for registering module entry and exit points.
57 */
58 module_init(simple_init);
59 module_exit(simple_exit);
60 MODULE_LICENSE("GPL");
61 MODULE_DESCRIPTION("simple module");
62 MODULE_AUTHOR("SGG");
```

C Tab Width: 4 Ln 38, Col 5 INS

و در نهایت با استفاده از دستوراتی که در تمرین اول انجام دادیم، برنامه را با استفاده از فایل نوشته شده Makefile کامپایل کرده (البته باید حواسمان باشد که برای اجرای برنامه دیگر و استفاده از این فایل باید نام ماژول داخل این فایل را به فایل کد جدید تولید شده عوض کنیم) و ماژول تولید شده به فرمت Main.ko را در کرنل لینوکس بارگذاری و نصب میکند و همچنین با اجرای دستوراتی که در تمرین 1 انجام شد، نتایج را در عکس ها میبینیم:



در عکس قبلی نیز میبینیم که ماژول اضافه شده است.



و در اخر نتایج مورد انتظار دستور کار را میبینیم که با اجرای dmesg پس از بارگذاری و حذف کردن به ما پیام های چاپ شده مورد انتظارمان را میدهد.