



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

گزارش کار آزمایشگاه سیستم عامل – شماره ۲

نوشتن انواع واحد های هسته و اجرای آنها روی هسته

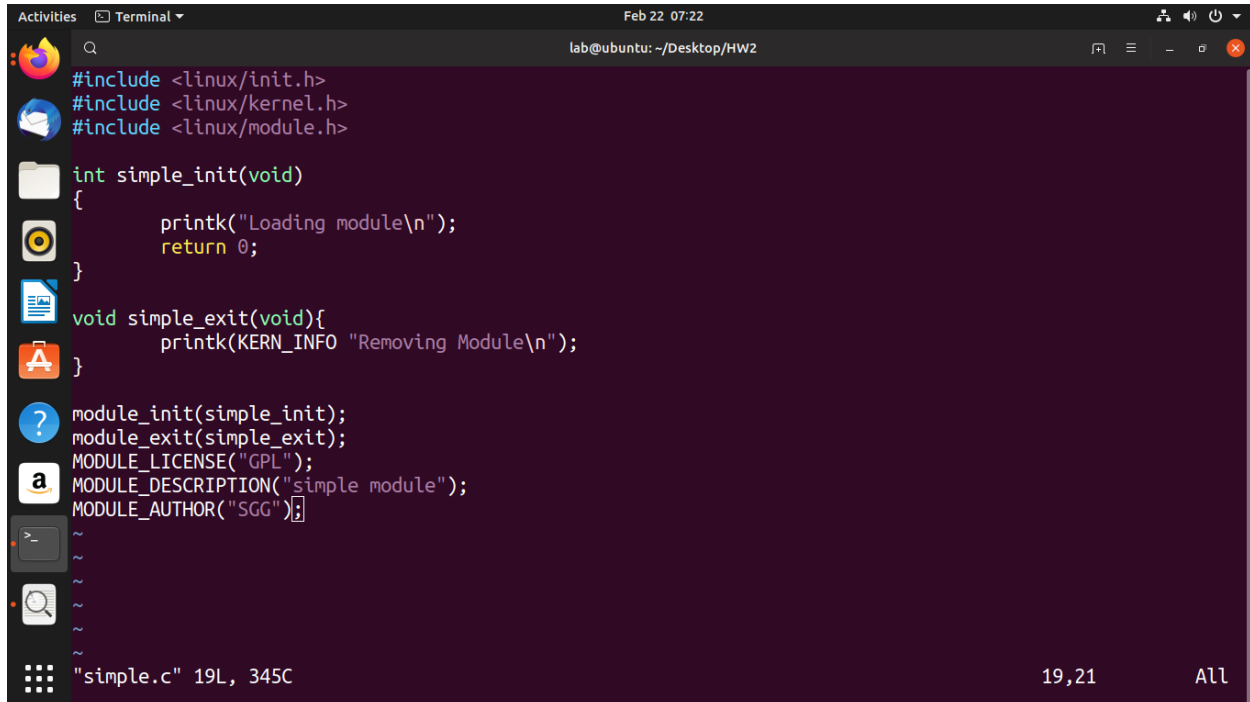
حسنا اویارحسینی - ۹۸۲۳۰۱۰

استاد درس: جناب آقای مهندس کیخا

نیمسال دوم سال تحصیلی ۱۴۰۰-۰۱

بخش ۱- ایجاد ماژولهای هسته)

در بخش اول آزمایش یک ماژول ساده نوشته و آن را به kernel اضافه کرده و سپس پاک میکنیم محتوای ماژول به صورت زیر می باشد که به اختصار به توضیح درباره آن میپردازیم:



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

int simple_init(void)
{
    printk("Loading module\n");
    return 0;
}

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

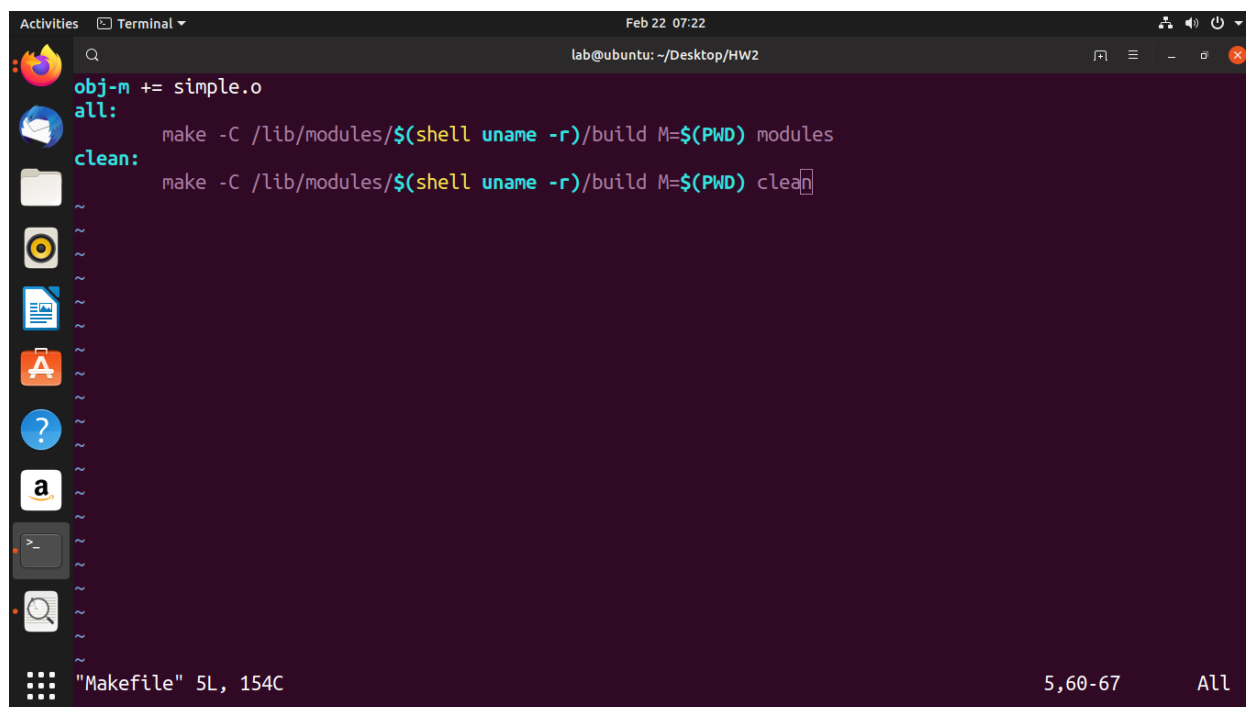
module_init(simple_init);
module_exit(simple_exit);
MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("simple module");
MODULE_AUTHOR("SGG");
```

The screenshot shows a terminal window with the code for a simple Linux kernel module. The code includes necessary headers, defines an initialization function, an exit function, and registers them with the kernel module system. The status bar at the bottom indicates the file is "simple.c" with 19 lines and 345 characters.

شکل 1 ماژول simple.c

تابع `simple_init` که نقطه شروع ماژول است در هنگام اضافه کردن ماژول به kernel اجرا می شود، که در اینجا ما فقط پیغام Loading Module را چاپ میکنیم و تابع `simple_exit` نیز در هنگام حذف کردن ماژول (به کمک دستور `rmmod`) اجرا می شود که در این برنامه ما فقط پیغام Removing Module را با اولویت `KERNEL_INFO` چاپ میکنیم. خطوط آخر معرف جزئیات مربوط به مجوز نرم افزار، توصیف ماژول و نویسنده می باشد.

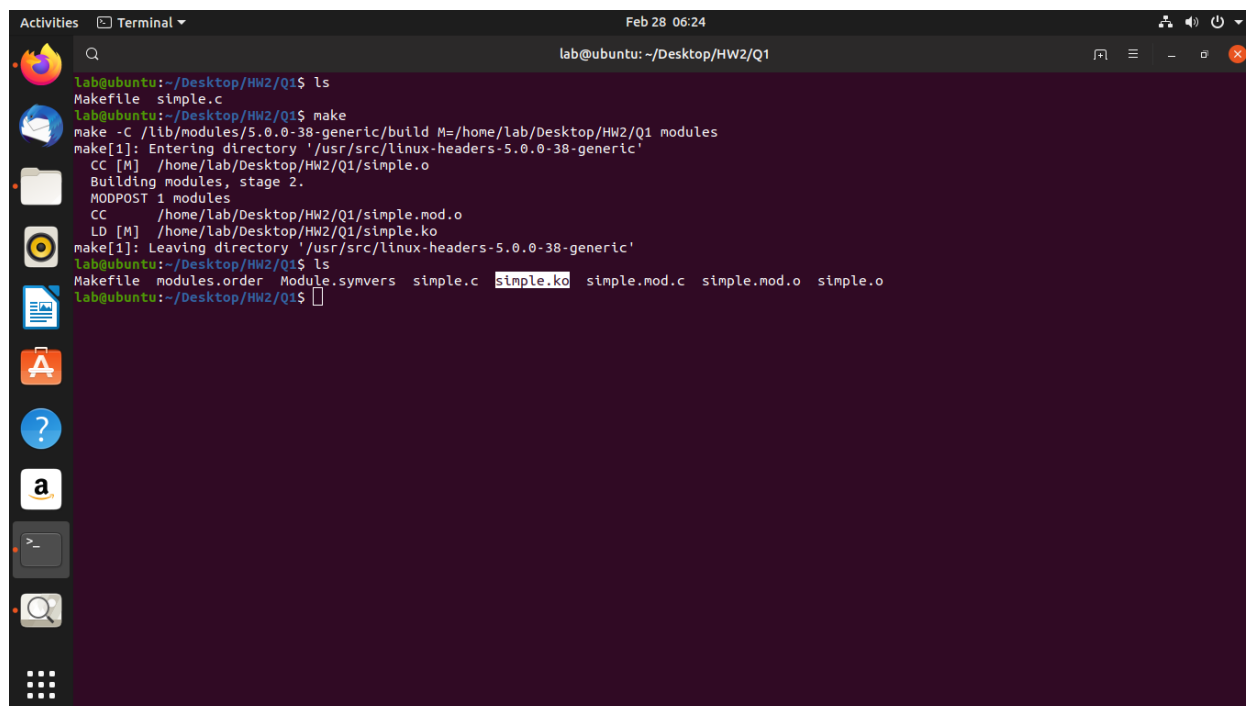
سپس برای کامپایل کردن برنامه ابتدا یک فایل با نام Makefile ساخته و در آن محتویات زیر را قرار میدهیم:



```
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
```

شکل ۲ فایل Makefile مربوط به ماژول simple

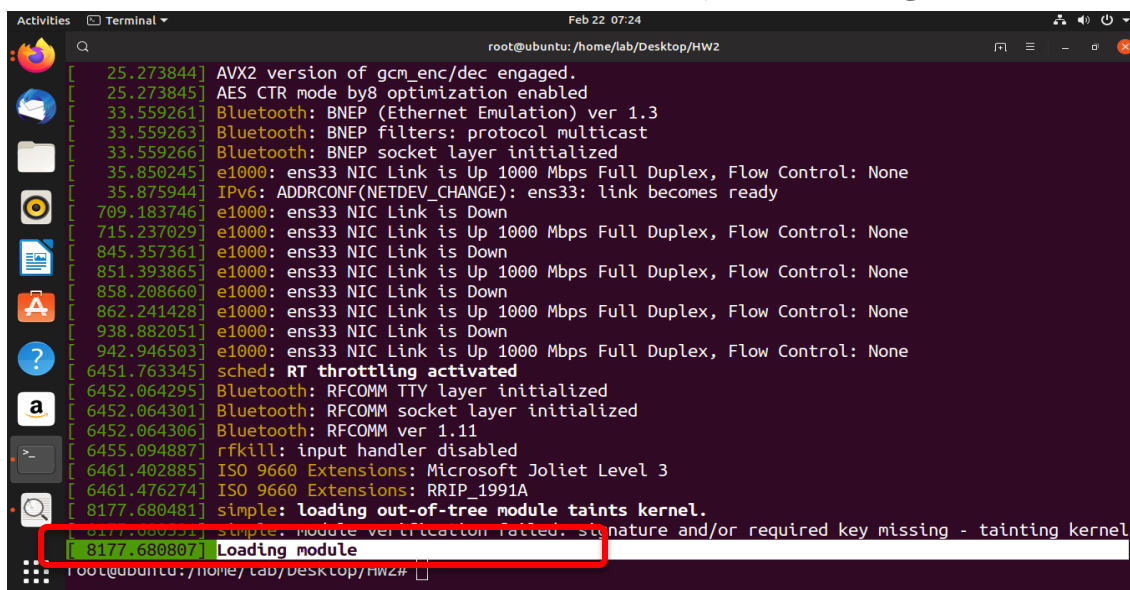
در نهایت به کمک دستور make ماژول را کامپایل کرده و فایل با پسوند ko ایجاد میشود.



```
lab@ubuntu:~/Desktop/HW2/Q1$ ls
Makefile simple.c
lab@ubuntu:~/Desktop/HW2/Q1$ make
make -C /lib/modules/5.0.0-38-generic/build M=/home/lab/Desktop/HW2/Q1 modules
make[1]: Entering directory '/usr/src/linux-headers-5.0.0-38-generic'
CC [M] /home/lab/Desktop/HW2/Q1/simple.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/lab/Desktop/HW2/Q1/simple.mod.o
LD [M] /home/lab/Desktop/HW2/Q1/simple.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.0.0-38-generic'
lab@ubuntu:~/Desktop/HW2/Q1$ ls
Makefile modules.order Module.symvers simple.c simple.ko simple.mod.c simple.mod.o simple.o
lab@ubuntu:~/Desktop/HW2/Q1$
```

شکل ۳ کامپایل کردن ماژول

به کمک دستور `insmod simple.ko` ماژول را به هسته اضافه میکنیم و سپس برای بررسی محتوی بافر سابقه هسته، از فرمان `dmesg` استفاده میکنیم و همانطور که در عکس ۱ مشخص است میتوان ببینیم که تابع `simple_init` به درستی اجرا شده و پیغام مدنظر چاپ شده است:



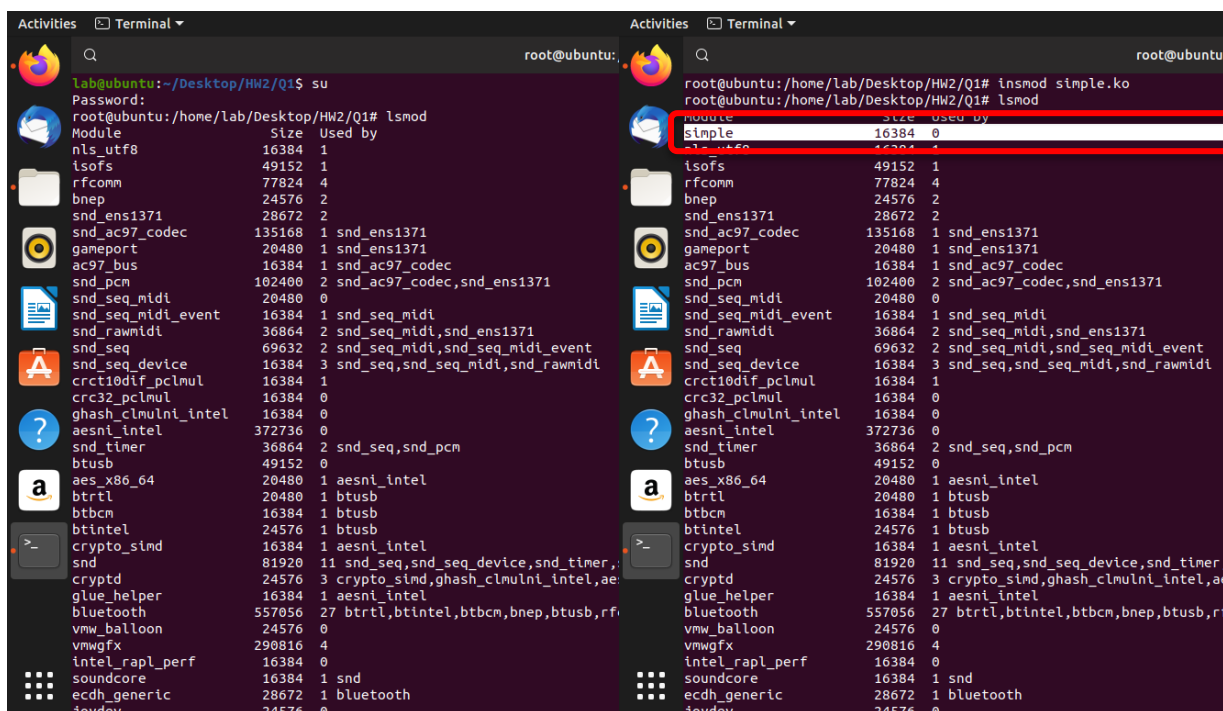
```

[ 25.273844] AVX2 version of gcm_enc/dec engaged.
[ 25.273845] AES CTR mode by8 optimization enabled
[ 33.559261] Bluetooth: BNEP (Ethernet Emulation) ver 1.3
[ 33.559263] Bluetooth: BNEP filters: protocol multicast
[ 33.559266] Bluetooth: BNEP socket layer initialized
[ 35.850245] e1000: ens33 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
[ 35.875944] IPv6: ADDRCONF(NETDEV_CHANGE): ens33: link becomes ready
[ 709.183746] e1000: ens33 NIC Link is Down
[ 715.237029] e1000: ens33 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
[ 845.357361] e1000: ens33 NIC Link is Down
[ 851.393865] e1000: ens33 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
[ 858.208660] e1000: ens33 NIC Link is Down
[ 862.241428] e1000: ens33 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
[ 938.882051] e1000: ens33 NIC Link is Down
[ 942.946503] e1000: ens33 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
[ 6451.763345] sched: RT throttling activated
[ 6452.064295] Bluetooth: RFCOMM TTY layer initialized
[ 6452.064301] Bluetooth: RFCOMM socket layer initialized
[ 6452.064306] Bluetooth: RFCOMM ver 1.11
[ 6455.094887] rfkill: input handler disabled
[ 6461.402885] ISO 9660 Extensions: Microsoft Joliet Level 3
[ 6461.476274] ISO 9660 Extensions: RRIP_1991A
[ 8177.680481] simple: loading out-of-tree module taints kernel.
[ 8177.680481] simple: module verification failed: signature and/or required key missing - tainting kernel
[ 8177.680807] Loading module

```

شکل ۴ اضافه کردن ماژول به کرنل

همچنین به کمک دستور `lsmod` میتوانیم لیست ماژول های فعلی را ببینیم و مشاهده میکنیم که ماژول ما نیز به این لیست اضافه شده:



```

lab@ubuntu: ~/Desktop/HW2/Q1$ su
root@ubuntu: /home/lab/Desktop/HW2/Q1# lsmod
Module      Size  Used by
nls_utf8    16384  1
iso9660fs   49152  1
rfcomm      77824  4
bnep        24576  2
snd_ens1371 28672  2
snd_ac97_codec 135168  1 snd_ens1371
gameport    20480  1 snd_ens1371
ac97_bus    16384  1 snd_ac97_codec
snd_pcm     102400  2 snd_ac97_codec,snd_ens1371
snd_seq_midi 20480  0
snd_seq_midi_event 16384  1 snd_seq_midi
snd_rawmidi 36864  2 snd_seq_midi,snd_ens1371
snd_seq     69632  2 snd_seq_midi,snd_seq_midi_event
snd_seq_device 16384  3 snd_seq,snd_seq_midi,snd_rawmidi
crc32_pclmul 16384  1
ghash_clmulni_intel 16384  0
aesni_intel 372736  0
snd_timer   36864  2 snd_seq,snd_pcm
btusb       49152  0
aes_x86_64 20480  1 aesni_intel
btrtl       20480  1 btusb
btbcm       16384  1 btusb
btintel     24576  1 btusb
crypto_simd 16384  1 aesni_intel
snd          81920  11 snd_seq,snd_seq_device,snd_timer,
cryptd      24576  3 crypto_simd,ghash_clmulni_intel,ae
glue_helper 16384  1 aesni_intel
bluetooth   557056  27 btrtl,btintel,btbcm,bnep,btusb,rf
vmw_balloon 24576  0
vmwgfx      290816  4
intel_rapl_perf 16384  0
soundcore   16384  1 snd
ecdh_generic 28672  1 bluetooth
i2cdev       24576  0

```

```

root@ubuntu: /home/lab/Desktop/HW2/Q1# insmod simple.ko
root@ubuntu: /home/lab/Desktop/HW2/Q1# lsmod
Module      Size  Used by
simple        16384  0
nls_utf8    16384  1
iso9660fs   49152  1
rfcomm      77824  4
bnep        24576  2
snd_ens1371 28672  2
snd_ac97_codec 135168  1 snd_ens1371
gameport    20480  1 snd_ens1371
ac97_bus    16384  1 snd_ac97_codec
snd_pcm     102400  2 snd_ac97_codec,snd_ens1371
snd_seq_midi 20480  0
snd_seq_midi_event 16384  1 snd_seq_midi
snd_rawmidi 36864  2 snd_seq_midi,snd_ens1371
snd_seq     69632  2 snd_seq_midi,snd_seq_midi_event
snd_seq_device 16384  3 snd_seq,snd_seq_midi,snd_rawmidi
crc32_pclmul 16384  1
ghash_clmulni_intel 16384  0
aesni_intel 372736  0
snd_timer   36864  2 snd_seq,snd_pcm
btusb       49152  0
aes_x86_64 20480  1 aesni_intel
btrtl       20480  1 btusb
btbcm       16384  1 btusb
btintel     24576  1 btusb
crypto_simd 16384  1 aesni_intel
snd          81920  11 snd_seq,snd_seq_device,snd_timer,
cryptd      24576  3 crypto_simd,ghash_clmulni_intel,ae
glue_helper 16384  1 aesni_intel
bluetooth   557056  27 btrtl,btintel,btbcm,bnep,btusb,rf
vmw_balloon 24576  0
vmwgfx      290816  4
intel_rapl_perf 16384  0
soundcore   16384  1 snd
ecdh_generic 28672  1 bluetooth
i2cdev       24576  0

```

شکل ۵ اضافه شدن ماژول به لیست ماژول ها

سپس با دستور `rmmod simple` ماژول `simple` را از `kernel` حذف کرده و دوباره به کمک فرمان `dmesg` از درستی اجرای برنامه اطمینان پیدا میکنیم:

```

[ 25.273845] AES CTR mode by8 optimization enabled
[ 33.559261] Bluetooth: BNEP (Ethernet Emulation) ver 1.3
[ 33.559263] Bluetooth: BNEP filters: protocol multicast
[ 33.559266] Bluetooth: BNEP socket layer initialized
[ 35.850245] e1000: ens33 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
[ 35.875944] IPv6: ADDRCONF(NETDEV_CHANGE): ens33: link becomes ready
[ 709.183746] e1000: ens33 NIC Link is Down
[ 715.237029] e1000: ens33 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
[ 845.357361] e1000: ens33 NIC Link is Down
[ 851.393865] e1000: ens33 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
[ 858.208660] e1000: ens33 NIC Link is Down
[ 862.241428] e1000: ens33 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
[ 938.882051] e1000: ens33 NIC Link is Down
[ 942.946503] e1000: ens33 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
[ 6451.763345] sched: RT throttling activated
[ 6452.064295] Bluetooth: RFCOMM TTY layer initialized
[ 6452.064301] Bluetooth: RFCOMM socket layer initialized
[ 6452.064306] Bluetooth: RFCOMM ver 1.11
[ 6455.094887] rfkill: input handler disabled
[ 6461.402885] ISO 9660 Extensions: Microsoft Joliet Level 3
[ 6461.476274] ISO 9660 Extensions: RRIP_1991A
[ 8177.680481] simple: loading out-of-tree module taints kernel.
[ 8177.680531] simple: module verification failed: signature and/or required key missing - tainting kernel
[ 8177.680807] Loading module
[ 8314.739604] Removing Module
root@ubuntu:/home/lab/Desktop/HW2#

```

شکل 6 حذف ماژول

و با اجرای دستور `lsmod` نیز میبینیم ماژول مدنظر دیگر در لیست ماژول ها وجود ندارد:

```

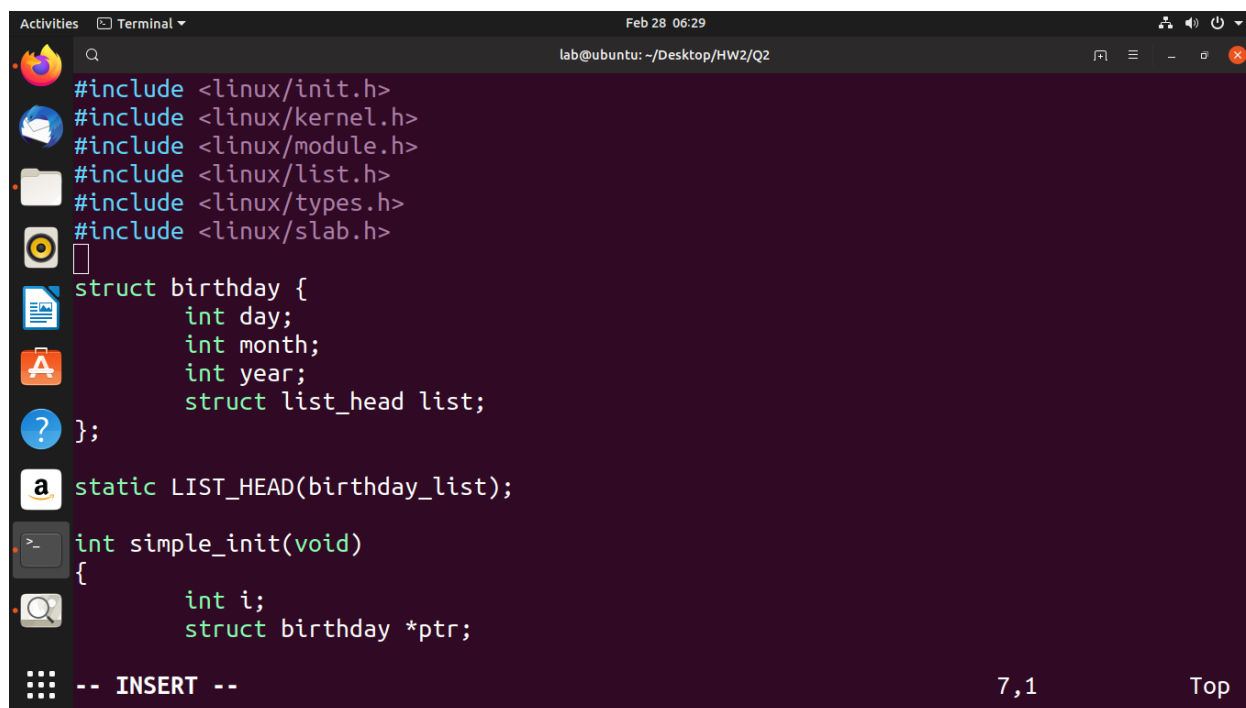
root@ubuntu:/home/lab/Desktop/HW2/Q1# rmmod simple
root@ubuntu:/home/lab/Desktop/HW2/Q1# ls
Makefile modules.order Module.symvers simple.c simple.ko simple.mod.c simple.mod.o simple.o
root@ubuntu:/home/lab/Desktop/HW2/Q1# lsmod
Module                  Size  Used by
nls_utf8                 16384  1
iso9660                 49152  1
rfcomm                  77824  4
bnep                    24576  2
snd_ens1371             28672  2
snd_ac97_codec         135168  1 snd_ens1371
gambpport               20480  1 snd_ens1371
ac97_bus                16384  1 snd_ac97_codec
snd_pcm                102400  2 snd_ac97_codec,snd_ens1371
snd_seq_midi            20480  0
snd_seq_midi_event      16384  1 snd_seq_midi
snd_rawmidi             36864  2 snd_seq_midi,snd_ens1371
snd_seq                 69632  2 snd_seq_midi,snd_seq_midi_event
snd_seq_device          16384  3 snd_seq,snd_seq_midi,snd_rawmidi
crct10dif_pclmul        16384  1
crc32_pclmul            16384  0
ghash_clmulni_intel     16384  0
aesni_intel            372736  0
snd_timer              36864  2 snd_seq,snd_pcm
btusb                   49152  0
aes_x86_64             20480  1 aesni_intel
btrtl                  20480  1 btusb
btbcm                  16384  1 btusb
btintel                24576  1 btusb
crypto_simd            16384  1 aesni_intel
snd                    81920  11 snd_seq,snd_seq_device,snd_timer,snd_ac97_codec,snd_pcm,snd_rawmidi,snd_ens1371
cryptd                 24576  3 crypto_simd,ghash_clmulni_intel,aesni_intel
glue_helper            16384  1 aesni_intel
bluetooth              557056  27 btrtl,btintel,btbcm,bnep,btusb,rfcomm
vmw_balloon            24576  0
vmwgfx                 290816  4
intel_rapl_perf         16384  0
soundcore              16384  1 snd
ac97_generic            28672  1 bluetooth

```

شکل 7 حذف ماژول از لیست ماژول ها

بخش ۲- ساختمان داده های هسته)

در بخش بعدی می‌خواهیم از ساختمان داده لیست پیوندی در هسته استفاده کنیم و یک لیست پیوندی ساخته، به آن گره‌هایی از نوع struct ای که خودمان تعریف میکنیم اضافه کنیم و سپس این عناصر را حذف کنیم. برای این کار ابتدا struct را تعریف میکنیم و دقت داریم برای اینکه لیست پیوندی در میان گره‌های سازنده لیست قرار گیرد باید struct list_head list را به عنوان عضو به struct اضافه کنیم و همچنین به کمک ماکرو LIST_HEAD شی birthday_list را که اشاره گری به ابتدای لیست میباشد را میسازیم:



```
lab@ubuntu: ~/Desktop/HW2/Q2
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/list.h>
#include <linux/types.h>
#include <linux/slab.h>

struct birthday {
    int day;
    int month;
    int year;
    struct list_head list;
};

static LIST_HEAD(birthday_list);

int simple_init(void)
{
    int i;
    struct birthday *ptr;

    -- INSERT --
7,1 Top
```

شکل 8 تعریف استراکت

سپس در تابع init_simple به کمک یک حلقه For ۵ شی از نوع struct birthday ساخته، مریال‌داردهی کرده و به آنها حافظه اختصاص داده، و سپس آنها را به لینکد لیستمان که به کمک birthday_list به آن دسترسی داریم اضافه میکنیم. پس از اضافه کردن هر شی یکبار وضعیت فعلی لینکد لیست را به کمک دستور list_for_each_entry(ptr, &birthday_list, list) در بافر چاپ میکنیم تا شاهد روند پر شدن لینکد لیست باشیم:

```
Activities Terminal Feb 28 06:32 lab@ubuntu: ~/Desktop/HW2/Q2
static LIST_HEAD(birthday_list);

int simple_init(void)
{
    int i;
    struct birthday *ptr;

    printk("Loading module\n");
    for (i = 0; i < 5; i++){

        struct birthday *person;
        person = kmalloc(sizeof(person), GFP_KERNEL);
        person->day = 2 + i;
        person->month = 3 + i;
        person->year = 1995 + 2 * i;
        INIT_LIST_HEAD(&person->list);
        list_add_tail(&person->list, &birthday_list);

        list_for_each_entry(ptr, &birthday_list, list)
            printk("> %d/%d/%d\n", ptr->year, ptr->month, ptr->day);
        printk("----");

    }

    return 0;
}
-- INSERT -- 34,1 34%
```

شکل 9 اضافه کردن گره ها به لیست

در بخش آخر نیز در تابع `simple_exit` گره‌هایی را که در لینکد لیست اضافه کرده بودیم به ترتیب چاپ می‌کنیم و در هر مرحله وضعیت فعلی لینکد لیست را در بافر چاپ می‌کنیم (لازم به ذکر است که برای حذف گره ها باید از دستور `list_for_each_entry_safe` استفاده کنیم):

```
Activities Terminal Feb 28 06:39 lab@ubuntu: ~/Desktop/HW2/Q2
void simple_exit(void){
    struct birthday *ptr, *tmp;
    printk(KERN_INFO "Removing Module\n");

    list_for_each_entry_safe(ptr, tmp, &birthday_list, list){

        list_del(&ptr->list);
        kfree(ptr);

        list_for_each_entry(ptr, &birthday_list, list){
            printk("> %d/%d/%d\n", ptr->year, ptr->month, ptr->day);
        }
        printk("----");

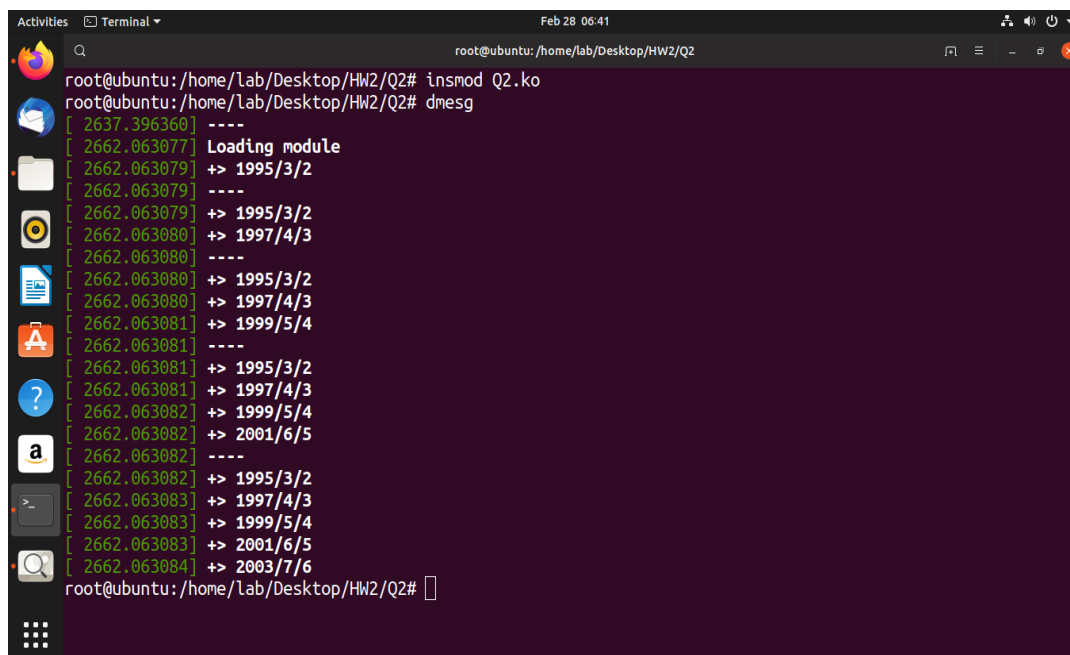
    }

}

module_init(simple_init);
module_exit(simple_exit);
MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("simple module");
MODULE_AUTHOR("SGG");
-- INSERT -- 46,1 97%
```

شکل 10 حذف گره ها از لینک لیست

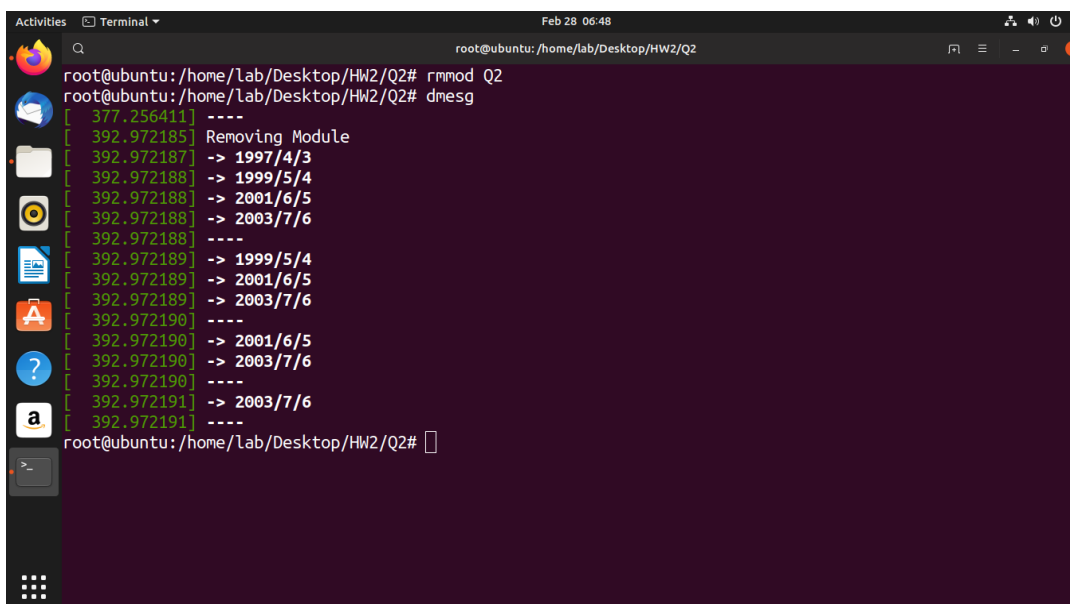
برای بررسی صحت کد ابتدا ماژول نوشته شده را به کمک یک Makefile کامپایل کرده و همانند مرحله‌ای که در بخش یک توضیح داده شد به kernel اضافه میکنیم، خروجی دستور dmesg در تصویر زیر نمایش داده شده است:



```
root@ubuntu:/home/Lab/Desktop/HW2/Q2# insmod Q2.ko
root@ubuntu:/home/Lab/Desktop/HW2/Q2# dmesg
[ 2637.396360] ----
[ 2662.063077] Loading module
[ 2662.063079] +> 1995/3/2
[ 2662.063079] ----
[ 2662.063079] +> 1995/3/2
[ 2662.063080] +> 1997/4/3
[ 2662.063080] ----
[ 2662.063080] +> 1995/3/2
[ 2662.063080] +> 1997/4/3
[ 2662.063081] +> 1999/5/4
[ 2662.063081] ----
[ 2662.063081] +> 1995/3/2
[ 2662.063081] +> 1997/4/3
[ 2662.063082] +> 1999/5/4
[ 2662.063082] +> 2001/6/5
[ 2662.063082] ----
[ 2662.063082] +> 1995/3/2
[ 2662.063083] +> 1997/4/3
[ 2662.063083] +> 1999/5/4
[ 2662.063083] +> 2001/6/5
[ 2662.063084] +> 2003/7/6
root@ubuntu:/home/Lab/Desktop/HW2/Q2#
```

شکل 11 اضافه شدن گره ها به لیست

سپس ماژول را حذف کرده و دوباره دستور dmesg را اجرا میکنیم و مشاهده میکنیم که به ترتیب گره ها حذف شده اند:



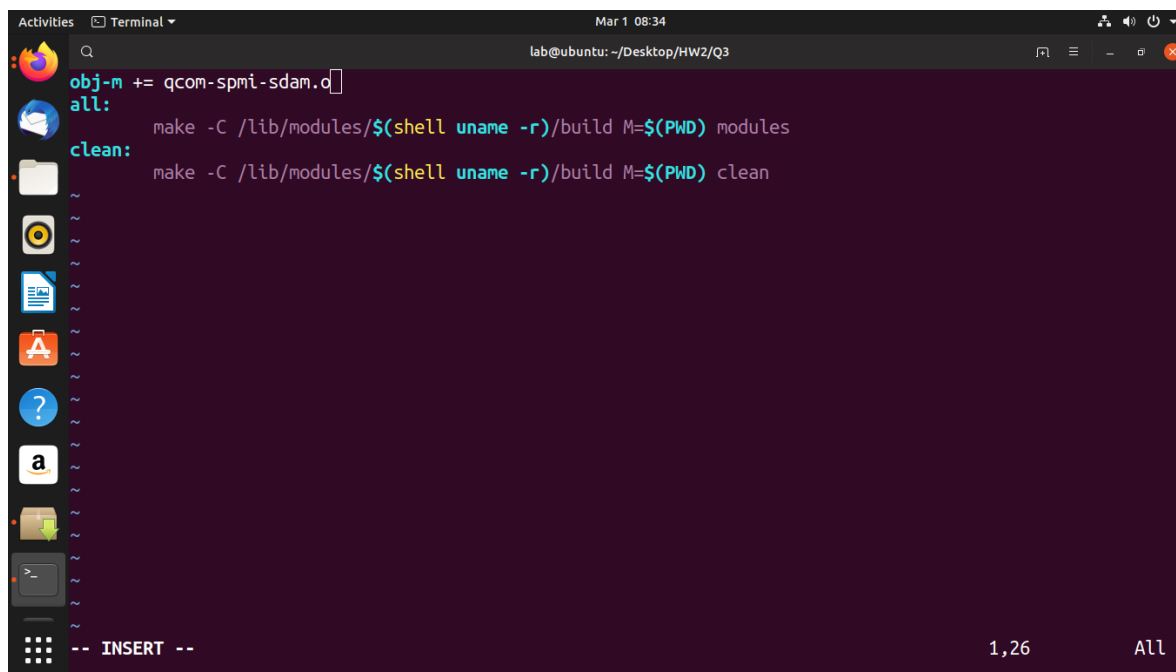
```
root@ubuntu:/home/Lab/Desktop/HW2/Q2# rmmod Q2
root@ubuntu:/home/Lab/Desktop/HW2/Q2# dmesg
[ 377.256411] ----
[ 392.972185] Removing Module
[ 392.972187] -> 1997/4/3
[ 392.972188] -> 1999/5/4
[ 392.972188] -> 2001/6/5
[ 392.972188] -> 2003/7/6
[ 392.972188] ----
[ 392.972189] -> 1999/5/4
[ 392.972189] -> 2001/6/5
[ 392.972189] -> 2003/7/6
[ 392.972190] ----
[ 392.972190] -> 2001/6/5
[ 392.972190] -> 2003/7/6
[ 392.972190] ----
[ 392.972191] -> 2003/7/6
[ 392.972191] ----
root@ubuntu:/home/Lab/Desktop/HW2/Q2#
```

شکل 12 حذف شدن گره ها از لیست

بخش ۳- اضافه کردن یک ماژول دلخواه به kernel

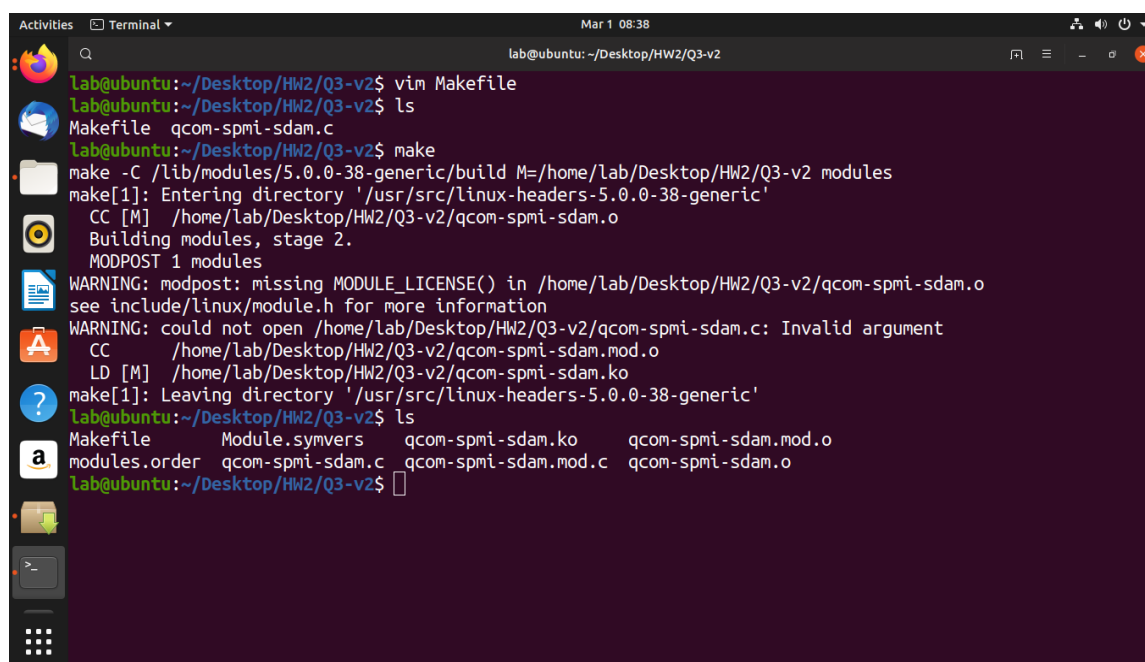
در این بخش یک ماژول دلخواه (در اینجا ما ماژول [qcom-spmi-sdam](#) را انتخاب کرده ایم) به kernel اضافه و سپس حذف میکنیم.

۱- ابتدا فایل C را به کمک Makefile کامپایل میکنیم:



```
obj-m += qcom-spmi-sdam.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
```

شکل ۱۳ فایل Makefile



```
lab@ubuntu:~/Desktop/HW2/Q3-v2$ vim Makefile
lab@ubuntu:~/Desktop/HW2/Q3-v2$ ls
Makefile  qcom-spmi-sdam.c
lab@ubuntu:~/Desktop/HW2/Q3-v2$ make
make -C /lib/modules/5.0.0-38-generic/build M=/home/lab/Desktop/HW2/Q3-v2 modules
make[1]: Entering directory '/usr/src/linux-headers-5.0.0-38-generic'
CC [M] /home/lab/Desktop/HW2/Q3-v2/qcom-spmi-sdam.o
Building modules, stage 2.
MODPOST 1 modules
WARNING: modpost: missing MODULE_LICENSE() in /home/lab/Desktop/HW2/Q3-v2/qcom-spmi-sdam.o
see include/linux/module.h for more information
WARNING: could not open /home/lab/Desktop/HW2/Q3-v2/qcom-spmi-sdam.c: Invalid argument
CC /home/lab/Desktop/HW2/Q3-v2/qcom-spmi-sdam.mod.o
LD [M] /home/lab/Desktop/HW2/Q3-v2/qcom-spmi-sdam.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.0.0-38-generic'
lab@ubuntu:~/Desktop/HW2/Q3-v2$ ls
Makefile  Module.symvers  qcom-spmi-sdam.ko  qcom-spmi-sdam.mod.o
modules.order  qcom-spmi-sdam.c  qcom-spmi-sdam.mod.c  qcom-spmi-sdam.o
lab@ubuntu:~/Desktop/HW2/Q3-v2$
```

شکل ۱۴ کامپایل کردن فایل C.

۲- سپس ماژول را به هسته اضافه میکنیم:

```
root@ubuntu: /home/lab/Desktop/HW2/Q3-v2# insmod qcom-spmi-sdam.ko
root@ubuntu: /home/lab/Desktop/HW2/Q3-v2# lsmod
Module                  Size  Used by
qcom_spmi_sdam          12288  0
nls_utf8                16384  1
isofs                   49152  1
rfcomm                  77824  4
bnep                    24576  2
snd_ens1371             28672  2
snd_ac97_codec          135168 1 snd_ens1371
gameport                20480 1 snd_ens1371
ac97_bus                16384 1 snd_ac97_codec
snd_pcm                 102400 2 snd_ac97_codec,snd_ens1371
snd_seq_midi            20480  0
crct10dif_pclmul        16384  1
snd_seq_midi_event      16384  1 snd_seq_midi
crc32_pclmul            16384  0
snd_rawmidi             36864  2 snd_seq_midi,snd_ens1371
ghash_clmulni_intel     16384  0
btusb                   49152  0
aesni_intel            372736 0
aes_x86_64              20480  1 aesni_intel
crypto_simd             16384  1 aesni_intel
btrtl                  20480  1 btusb
snd_seq                 69632  2 snd_seq_midi,snd_seq_midi_event
cryptd                  24576  3 crypto_simd,ghash_clmulni_intel,aesni_intel
btbcm                   16384  1 btusb
```

شکل 15 اضافه شدن ماژول به هسته

۲- در نهایت ماژول را حذف میکنیم:

```
root@ubuntu: /home/lab/Desktop/HW2/Q3-v2# rmmod qcom-spmi-sdam
root@ubuntu: /home/lab/Desktop/HW2/Q3-v2# lsmod
Module                  Size  Used by
floppy                  81920  0
nls_utf8                16384  1
isofs                   49152  1
rfcomm                  77824  4
bnep                    24576  2
snd_ens1371             28672  2
snd_ac97_codec          135168 1 snd_ens1371
gameport                20480 1 snd_ens1371
ac97_bus                16384 1 snd_ac97_codec
snd_pcm                 102400 2 snd_ac97_codec,snd_ens1371
snd_seq_midi            20480  0
crct10dif_pclmul        16384  1
snd_seq_midi_event      16384  1 snd_seq_midi
crc32_pclmul            16384  0
snd_rawmidi             36864  2 snd_seq_midi,snd_ens1371
ghash_clmulni_intel     16384  0
btusb                   49152  0
aesni_intel            372736 0
aes_x86_64              20480  1 aesni_intel
crypto_simd             16384  1 aesni_intel
btrtl                  20480  1 btusb
snd_seq                 69632  2 snd_seq_midi,snd_seq_midi_event
cryptd                  24576  3 crypto_simd,ghash_clmulni_intel,aesni_intel
btbcm                   16384  1 btusb
```

شکل 16 حذف ماژول از هسته