



**KERNEL MODULE PROGRAMMING**

# **OPERATING SYSTEM**



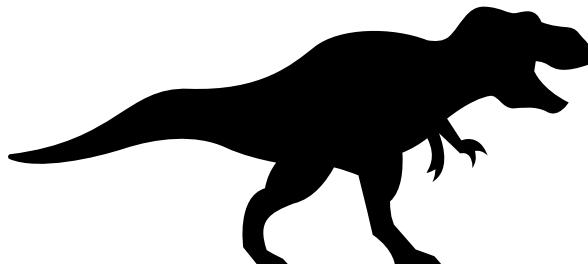
**COLLABORATORS NAME & ID :**  
**MARYAM ALIKARAMI 9731045**  
**MAHAN AHMADVAND 9731071**



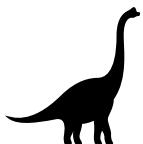


# KERNEL MODULE PROGRAMMING

# OPERATING SYSTEM



**COLLABORATORS NAME & ID :**  
**MARYAM ALIKARAMI 9731045**  
**MAHAN AHMADVAND 9731071**



# **OPERATING SYSTEM LABORATORY**

## **KERNEL MODULE PROGRAMMING**



**INSTRUCTOR :**  
**ARMAN GHEISARI**

**COLLABORATORS :**  
**MARYAM ALIKARAMI**  
**MAHAN AHMADVAND**

# QUESTION 1

در این بخش از آزمایش قصد داریم تا یک ماژول هسته را بارگذاری و حذف کنیم و با بررسی بافر سابقه‌ی هسته مطمئن شویم که این فرایند را به درستی طی کرده‌ایم.

در ابتدا بهتر است اشاره کنیم که ماژول‌های هسته چه هستند. Kernel Modules یا ماژول‌های هسته عبارتند از تکه کدهایی که در هنگام run-time می‌توانند در کرنل بارگذاری یا از آن حذف شوند و برای اجرای آن‌ها نیازی نیست که سیستم مجددا راه اندازی شود.

حال به انجام این بخش از آزمایش می‌پردازیم. از آن جایی که ماژول‌های هسته به صورت مستقیم با کرنل در ارتباط هستند و هرگونه خطا و اشکال در آن‌ها می‌تواند باعث بروز اشکال در کل سیستم گردد، قبل از شروع کار از وضعیت فعلی ماشین مجازی خود snapshot می‌گیریم تا در صورت نیاز بتوانیم به شرایطی که قبل از شروع آزمایش داشتیم بازگردیم.



Tools

Ubuntu 19 (pre lab 2) Running



Take	Delete	Restore	Properties	Clone	Settings	Discard	Show
Name	Taken						
pre lab 2	21/03/31 3:24 AM (1 hour ago)						
Current State (changed)							

در مرحله‌ی بعدی برای اطمینان از اینکه هدراهای موردنیاز ما نصب شده و آپدیت هستند، مراحل زیر را طی می‌کنیم :

ابتدا دستور `sudo sed -i -re 's/([a-z]{2}\.).? archive.ubuntu.com|security.ubuntu.com/old-releases.ubuntu.com/g' /etc/apt/sources.list` و سپس به ترتیب دستورات `sudo apt-get update` و `sudo apt-get install linux-headers-$(uname -r)` کنیم.

Q

oslab@OSLab-VirtualBox:/



```
oslab@OSLab-VirtualBox:/$ sudo sed -i -re 's/([a-z]{2}\.).)?archive.ubuntu.com|security.ubuntu.com/old-releases.ubuntu.com/g' /etc/apt/sources.list
oslab@OSLab-VirtualBox:/$ sudo apt-get update
Hit:1 http://old-releases.ubuntu.com/ubuntu disco InRelease
Hit:2 http://old-releases.ubuntu.com/ubuntu disco-updates InRelease
Hit:3 https://download.sublimetext.com apt/stable/ InRelease
Hit:4 http://old-releases.ubuntu.com/ubuntu disco-backports InRelease
Hit:5 http://old-releases.ubuntu.com/ubuntu disco-security InRelease
Reading package lists... Done
oslab@OSLab-VirtualBox:/$ sudo apt-get install linux-headers-$(uname -r)
Reading package lists... Done
Building dependency tree
Reading state information... Done
linux-headers-5.0.0-13-generic is already the newest version (5.0.0-13.14).
0 upgraded, 0 newly installed, 0 to remove and 326 not upgraded.
oslab@OSLab-VirtualBox:/$
```

دستور **sed** یا **stream editor** دستوری است که این قابلیت را به کاربر می دهد تا بدون بازکردن یک فایل در ادیتور و فقط با اجرای این دستور، فایل را ادیت کند و فرآیندهایی همچون **search**, **replacement**, **insert**, **deletion** را به روی آن فایل اجرا کند.

از دستور **apt-get** برای مدیریت پکیج ها به واسطه‌ی کار با بخش (APT) Advanced Packaging Tool در لینوکس، استفاده می شود.

از این دستور برای حذف، نصب و به روزرسانی پکیج ها و نرم افزارها استفاده می شود.

بعد از اینکه مطمئن شدیم هدرهای موردنظر ما نصب هستند، ادیتور **sublime-text** را با استفاده از دستور **sudo snap install sublime-text** نصب می کنیم تا بتوانیم ادیت فایلهای خود را در محیطی کارآمدتر انجام دهیم.

```
Q oslab@OSLab-VirtualBox:~$ sudo snap install sublime-text
error: This revision of snap "sublime-text" was published using classic
confinement and thus may perform arbitrary system changes outside of the
security sandbox that snaps are usually confined to, which may put your
system at risk.

If you understand and want to proceed repeat the command including
--classic.
oslab@OSLab-VirtualBox:~$ sudo snap install --classic sublime-text
Download snap "sublime-text" (97) from channel "stable"          3%   305kB/s 3m10s
```

snap یک سیستم مدیریت پکیج در لینوکس است که به واسطه‌ی آن می‌توان به راحتی بسیاری از برنامه‌هایی را که در این سیستم پکیج شده اند را دانلود و نصب کرد. این سیستم از Ubuntu 16.04 LTS به بعد روی تمام ورژن‌های اوبونتو نصب است و تمام برنامه‌هایی که با استفاده از snap نصب شوند به صورت اتوماتیک آپدیت خواهند شد.

پس از نصب، sublime-text را باز می‌کنیم تا کد مربوط به ماژول خود را در آن وارد و ادیت کنیم. (از آنجایی که کد این ماژول را آماده داریم آن را کپی می‌کنیم و در فرمت یک فایل c ذخیره می‌کنیم).

## دستور PRINTK معادل PRINTF است با این تفاوت که پیام خود را در بافر سابقه هسته چاپ می‌کند.

The screenshot shows a code editor with a dark theme. The file is named 'simplemod.c'. The code contains two `printk` statements. The first is at line 10: `printk(KERN_INFO "Loading Module\n");`. The second is at line 20: `printk(KERN_CRIT "Removing Module\n");`. Both lines have orange boxes around them. A large grey arrow points from the bottom right towards these two lines. At the bottom left, there is a yellow box containing Persian text. The status bar at the bottom shows 'Line 34, Column 22; Saved'.

```
File Edit Selection Find View Goto Tools Project Preferences Help
```

```
simplemod.c
```

```
1 #include <linux/init.h>
2 #include <linux/kernel.h>
3 #include <linux/module.h>
4
5
6
7 /* this function is called when the module is loaded*/
8 int simple_init(void)
9 {
10     printk(KERN_INFO "Loading Module\n");
11     return 0;
12 }
13
14
15
16
17 /* this function is called when the module is removed*/
18 void simple_exit(void)
19 {
20     printk(KERN_CRIT "Removing Module\n");
21 }
22
23
24
25 /* Macros for registering module entry and exit points.
26 */
27 module_init(simple_init);
28 module_exit(simple_exit);
```

این دو ماکرو محل ورود و خروج به مازول SIMPLEMOD را ثبت می‌کنند و در محل ورود و خروج، توابع SIMPLE\_INIT و SIMPLE\_EXIT را فراخوانی می‌کنند.

```
31
32 MODULE_LICENSE("GPL");
33 MODULE_DESCRIPTION("simple module");
34 MODULE_AUTHOR("SGG");
```

این ماکروها نیز حاوی اطلاعات و مشخصات مربوط به مازول می‌باشند

این عبارت، نشان‌دهندهی حالت و اهمیت پیام چاپ شده است. طبق جدول اولویت‌ها `kern_info` نشان دهنده‌ی یک پیام عادی است اما `kern_crit` یک پیام `critical` است که اولویت بالاتری دارد و در صورت مشاهده باید فوراً به آن رسیدگی شود. در اینجا این عبارت را تغییر دادیم تا مازول ما چند حالت متفاوت را در بگیرد. به همین دلیل است که پیام `removing` به رنگ قرمز چاپ خواهد شد.

فایل بدون پسوند دیگری شامل قوانین مربوط به کامپایل simplemod.c را در همین دایرکتوری ایجاد می کنیم و نام آن Makefile می گذاریم.(به همین دلیل تغییر دادن نام ماژول نیاز است تا در کد آماده ای که برای makefile در اختیار داریم نیز تغییر ایجاد کنیم)

هرگاه که می خواهیم از دستور make استفاده کنیم به یک نیز نیاز است که ارتباط فایلهای درون برنامه‌ی ما را توضیح می دهد.

```
simplemod.c
obj-m += mymod.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
```

simplemode.o

حالا در همین دایرکتوری فعلی (Desktop/lab) که فایل‌های make درون آن قرار دارند، ترمینال را باز می کنیم و دستور make را وارد می کنیم. از این دستور برای کامپایل کردن ماژول‌ها و پروژه‌های بزرگ استفاده می شود. پس از کامپایل، فایل simplemod.ko در همین دایرکتوری ایجاد می شود. پسوند .ko برای ماژول‌های هسته در نظر گرفته می شود.

در همین مرحله، آدرس entering directory را در جایی سیو می کنیم چون برای مراحل بعدی به آن نیاز داریم.

```
oslab@OSLab-VirtualBox:~/Desktop/lab$ make -C /lib/modules/5.0.0-13-generic/build M=/home/oslab/Desktop/lab modules
make[1]: Entering directory '/usr/src/linux-headers-5.0.0-13-generic'
  CC [M]  /home/oslab/Desktop/lab/simplemod.o
Building modules, stage 2...
MODPOST 1 modules
  LD [M]  /home/oslab/Desktop/lab/simplemod.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.0.0-13-generic'
oslab@OSLab-VirtualBox:~/Desktop/lab$
```

حالا که ماژول هسته‌ی ما ساخته شده، می‌توانیم آن را به کرنل اضافه و از آن حذف کنیم. برای اینکه به این فرآیند نظارت داشته باشیم، تب جدیدی در ترمینال باز می‌کنیم و دستور `tail -f /var/log/syslog` را وارد می‌کنیم تا لگ‌های سیستم را زیر نظر بگیریم و ببینیم که آیا خروجی دلخواه خود را می‌گیریم یا خیر (برای همین منظور می‌توان از دستور `dmesg` استفاده کرد که محتویات بافر سابقه هسته را چاپ می‌کند) طبق کد `simplemod.c`، انتظار داریم که با لود کردن ماژول پیام **Removing Module** و با حذف آن پیام **Loading Module** چاپ شوند.

حالا با استفاده از دستور `sudo insmod simplemod.ko` این ماژول را در کرنل لود می‌کنیم.

```
oslab@OSLab-VirtualBox:~/Desktop/lab$ make
make -C /lib/modules/5.0.0-13-generic/build M=/home/oslab/Desktop/lab modules
make[1]: Entering directory '/usr/src/linux-headers-5.0.0-13-generic'
  CC [M]  /home/oslab/Desktop/lab/simplemod.o
Building modules, stage 2.
MODPOST 1 modules
  CC      /home/oslab/Desktop/lab/simplemod.mod.o
  LD [M]  /home/oslab/Desktop/lab/simplemod.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.0.0-13-generic'
oslab@OSLab-VirtualBox:~/Desktop/lab$ sudo insmod simplemod.ko
[sudo] password for oslab:
oslab@OSLab-VirtualBox:~/Desktop/lab$ 
```

حالا اگر به تب دوم که دستور tail را در آن اجرا کرده بودیم نگاه کنیم، می بینیم که عبارت "loading module" طبق انتظار ما چاپ شده است.

```
oslab@OSLab-VirtualBox:~/Desktop/lab  x  oslab@OSLab-VirtualBox:~/Desktop/lab  x  ▾
Mar 31 15:18:01 OSLab-VirtualBox dbus-daemon[665]: [system] Activating via syste
md: service name='org.freedesktop.hostname1' unit='dbus-org.freedesktop.hostname
1.service' requested by ':1.868' (uid=1000 pid=29288 comm="rhythmbox" label="un
confined")
Mar 31 15:18:01 OSLab-VirtualBox systemd[1]: Starting Hostname Service...
Mar 31 15:18:02 OSLab-VirtualBox dbus-daemon[665]: [system] Successfully activat
ed service 'org.freedesktop.hostname1'
Mar 31 15:18:02 OSLab-VirtualBox systemd[1]: Started Hostname Service.
Mar 31 15:18:05 OSLab-VirtualBox rhythmbox[29288]: Can't set a parent on widget
which has a parent
Mar 31 15:18:32 OSLab-VirtualBox systemd[1]: systemd-hostnamed.service: Succeede
d.
Mar 31 15:21:30 OSLab-VirtualBox kernel: [49460.876060] Loading Module
Mar 31 15:21:48 OSLab-VirtualBox dbus-daemon[1630]: [session uid=1000 pid=1630]
Activating via systemd: service name='org.freedesktop.Tracker1.Miner.Extract' un
it='tracker-extract.service' requested by ':1.53' (uid=1000 pid=2074 comm="/usr/
lib/tracker/tracker-miner-fs" label="unconfined")
Mar 31 15:21:48 OSLab-VirtualBox systemd[1601]: Starting Tracker metadata extrac
tor...
Mar 31 15:21:49 OSLab-VirtualBox dbus-daemon[1630]: [session uid=1000 pid=1630]
Successfully activated service 'org.freedesktop.Tracker1.Miner.Extract'
Mar 31 15:21:49 OSLab-VirtualBox systemd[1601]: Started Tracker metadata extract
or.

```

در این حالت اگر دستور `sudo lsmod` را وارد کنیم که مژول های فعلی هسته را لیست می کند، می توانیم مژول خود را میان آن ها ببینیم.

```
oslab@OSLab-VirtualBox: ~/Desktop/lab × oslab@OSLab-VirtualBox: ~/Desktop/lab ×
make[1]: Entering directory '/usr/src/linux-headers-5.0.0-13-generic'
  CC [M]  /home/oslab/Desktop/lab/simplemod.o
Building modules, stage 2.
MODPOST 1 modules
  CC      /home/oslab/Desktop/lab/simplemod.mod.o
  LD [M]  /home/oslab/Desktop/lab/simplemod.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.0.0-13-generic'
oslab@OSLab-VirtualBox:~/Desktop/lab$ sudo insmod simplemod.ko
[sudo] password for oslab:
oslab@OSLab-VirtualBox:~/Desktop/lab$ sudo lsmod
Module           Size  Used by
simplemod        16384  0
nls_utf8         16384  1
iso9fs          49152  1
vboxsf          81920  1
vboxvideo        36864  0
snd_intel8x0     45056  2
crc1t0dif_pcilmul 16384  1
crc32_pcilmul   16384  0
snd_ac97_codec  135168  1 snd_intel8x0
ghash_clmulni_intel 16384  0
ac97_bus         16384  1 snd_ac97_codec
aesni_intel     372736  0
snd_pcm          102400  2 snd_intel8x0,snd_ac97_codec
```

با دستور `sudo lsmod | grep [module name]` می توانیم به دنبال یک مژول خاص که در حال حاضر در هسته لود شده است بگردیم، اگر این دستور را برای `simplemod` اجرا کنیم، آن را می یابیم.

```
autofs4          45056  2
hid_generic      16384  0
usbhid          53248  0
hid             126976  2 usbhid,hid_generic
psmouse         151552  0
ahci            40960  3
libahci         32768  1 ahci
i2c_piix4       28672  0
e1000          139264  0
pata_acpi       16384  0
video           45056  0
oslab@OSLab-VirtualBox:~/Desktop/lab$ sudo lsmod | grep simplemod
simplemod        16384  0
oslab@OSLab-VirtualBox:~/Desktop/lab$
```

در این مرحله قصد داریم مازولی که بارگذاری کردیم را از کرنل حذف کنیم. برای این کار از دستور `sudo rmmod [module name]` استفاده می کنیم.

```
pata_acpi      16384  0
video          45056  0
oslab@OSLab-VirtualBox:~/Desktop/lab$ sudo lsmod | grep simplemod
simplemod      16384  0
oslab@OSLab-VirtualBox:~/Desktop/lab$ sudo rmmod simplemod
```

پس از اجرای این دستور در تب دوم می بینیم که عبارت چاپ شده است.

```
oslab@OSLab-VirtualBox:~/Desktop/lab  x  oslab@OSLab-VirtualBox:~/Desktop/lab  x  ▾
it='tracker-extract.service' requested by ':1.53' (uid=1000 pid=2074 comm="/usr/lib/tracker/tracker-miner-fs" label="unconfined")
Mar 31 15:23:06 OSLab-VirtualBox systemd[1601]: Starting Tracker metadata extractor...
Mar 31 15:23:06 OSLab-VirtualBox dbus-daemon[1630]: [session uid=1000 pid=1630] Successfully activated service 'org.freedesktop.Tracker1.Miner.Extract'
Mar 31 15:23:06 OSLab-VirtualBox systemd[1601]: Started Tracker metadata extractor.
Mar 31 15:23:16 OSLab-VirtualBox systemd[1601]: tracker-extract.service: Succeeded.
Mar 31 15:23:22 OSLab-VirtualBox kernel: [49572.998118] Removing Module
Mar 31 15:23:25 OSLab-VirtualBox dbus-daemon[1630]: [session uid=1000 pid=1630] Activating via systemd: service name='org.freedesktop.Tracker1.Miner.Extract' unit='tracker-extract.service' requested by ':1.53' (uid=1000 pid=2074 comm="/usr/lib/tracker/tracker-miner-fs" label="unconfined")
Mar 31 15:23:25 OSLab-VirtualBox systemd[1601]: Starting Tracker metadata extractor...
Mar 31 15:23:25 OSLab-VirtualBox dbus-daemon[1630]: [session uid=1000 pid=1630] Successfully activated service 'org.freedesktop.Tracker1.Miner.Extract'
Mar 31 15:23:25 OSLab-VirtualBox systemd[1601]: Started Tracker metadata extractor.
Mar 31 15:23:35 OSLab-VirtualBox systemd[1601]: tracker-extract.service: Succeeded.
```

بعد از حذف کردن `simplemod` اگر از مازول های هسته لیست بگیریم یا مجددا به دنبال `simplemod` بگردیم، آن را نخواهیم یافت.

```
glue_helper          16384  1 aesni_intel
snd_seq              69632  2 snd_seq_midi,snd_seq_midi_event
drm_kms_helper      180224  2 vmwgfx,vboxvideo
snd_seq_device       16384  3 snd_seq,snd_seq_midi,snd_rawmidi
drm                 47136   6 vmwgfx,drm_kms_helper,vboxvideo,ttm
snd_timer            36864  3 snd_seq,snd_pcm
intel_rapl_perf     16384  0
snd                 81920  13 snd_seq,snd_seq_device,snd_intel8x0,snd_timer,s
nd_ac97_codec,snd_pcm,snd_rawmidi
fb_sys_fops          16384  1 drm_kms_helper
syscopyarea          16384  1 drm_kms_helper
sysfillrect          16384  1 drm_kms_helper
soundcore            16384  1 snd
input_leds            16384  0
sysimgblt            16384  1 drm_kms_helper
serio_raw             20480  0
vboxguest            339968  6 vboxsf
mac_hid               16384  0
sch fq_codel         20480  2
parport_pc            40960  0
ppdev                24576  0
lp                   20480  0
parport              53248  3 parport_pc,lp,ppdev
ip_tables             28672  0
x_tables              40960  1 ip_tables
autofs4               45056  2
hid_generic           16384  0
usbhid                53248  0
hid                  126976  2 usbhid,hid_generic
psmouse               151552  0
ahci                  40960  3
libahci                32768  1 ahci
i2c_piix4             28672  0
e1000                 139264  0
ata_acpi              16384  0
video                  45056  0
oslab@OSLab-VirtualBox:~/Desktop/lab$ sudo lsmod | grep simplemod
oslab@OSLab-VirtualBox:~/Desktop/lab$
```

همانطور که گفتیم با دستور dmesg نیز می توانیم پیام های بافر سابقه ی هسته را رویت کنیم. دلیل قرمز بودن عبارت remove را در بخش توضیحات کد شرح دادیم.

```
oslab@OSLab-VirtualBo... x oslab@OSLab-VirtualBo... x oslab@OSLab-VirtualBo... x ▾
6485.139969] usb 1-1: new full-speed USB device number 4 using xhci_hcd
6485.298999] usb 1-1: New USB device found, idVendor=80ee, idProduct=0021, bcd
device= 1.00
6485.299001] usb 1-1: New USB device strings: Mfr=1, Product=3, SerialNumber=0
6485.299002] usb 1-1: Product: USB Tablet
6485.299003] usb 1-1: Manufacturer: VirtualBox
6485.300727] input: VirtualBox USB Tablet as /devices/pci0000:00/0000:00:0c.0/
usb1/1-1/1-1:1.0/0003:80EE:0021.0003/input/input9
6485.362221] hid-generic 0003:80EE:0021.0003: input,hidraw0: USB HID v1.10 Mou
se [VirtualBox USB Tablet] on usb-0000:00:0c.0-1/input0
15782.279123] audit: type=1400 audit(1617154211.894:45): apparmor="STATUS" oper
ation="profile_load" profile="unconfined" name="snap.sublime-text.subl" pid=2651
comm="apparmor_parser"
15782.280039] audit: type=1400 audit(1617154211.894:46): apparmor="STATUS" oper
ation="profile_load" profile="unconfined" name="snap-update-ns.sublime-text" pid
=26516 comm="apparmor_parser"
46865.684277] gedit[24661]: segfault at 0 ip 00007f052d4f566a sp 00007ffda4d32e
0 error 4 in libgedit-3.14.so[7f052d4d3000+49000]
46865.684298] Code: 55 49 89 c1 48 8d 35 45 e5 02 00 48 89 c3 4c 8d 05 f3 e5 02
00 31 c0 bf 00 04 00 00 e8 bf 38 fe ff 48 8b 05 d0 d9 0a 00 5a 59 <8b> 30 85 f6
0f 84 dc 00 00 00 48 8b 78 08 48 89 de e8 d0 0a fe ff
49460.876060 Loading Module
49572.998118 Removing Module
slab@OSLab-VirtualBox:~/Desktop/lab$
```

# QUESTION 2

در این قسمت قصد داریم در نقطه ورود مازول، یک لیست پیوند شامل پنج عنصر `struct birthday` ایجاد کنیم، لیست پیوندی را پیمایش کنیم و محتوای آن را به بافر سابقه‌ی هسته انتقال دهیم و فرمان `dmesg` را احضار کنیم تا مطمئن شویم که به محض بار شدن مازول هسته، لیست به درستی ایجاد شده است، در نقطه خروج مازول عناصر لیست را از لیست پیوندی حذف کرده و دوباره حافظه آزاد شده را به هسته بر می‌گردانیم و سپس دوباره فرمان `dmesg` را احضار می‌کنیم تا بررسی کنیم به محض برداشتن مازول هسته، لیست حذف شده است.  
در این قسمت می‌خواهیم از لیست پیوندی دوطرفه استفاده کنیم.

برای تعریف لیست دوطرفه از دستور `static LIST_HEAD(birthday_list);` (این دستور برای ما یک لیست پیوند خالی دوطرفه ایجاد می‌کند) استفاده می‌کنیم و برای اینکه بتوانیم از این دستور استفاده کنیم باید `hdr<linux/list.h>` را `include` کنیم.

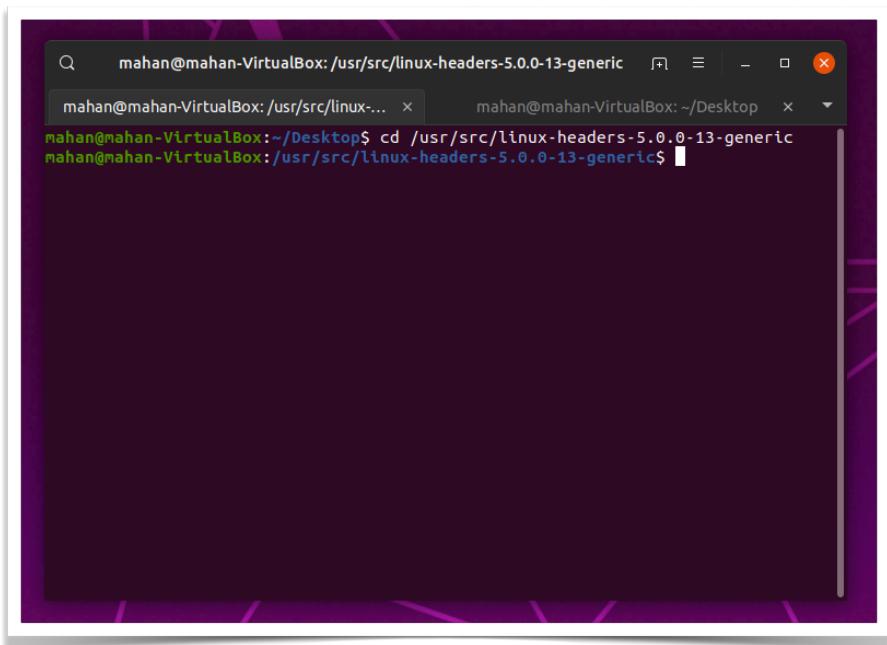
حال باید در این لیست پیوندی داده ذخیره کنیم و برای این کار باید یک `struct` تعریف کنیم نمونه‌ای از این `struct` را در صفحه‌ی بعد می‌بینیم :

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

حال `struct list_head` باعث می شود که عناصر لیست پیوندی بتوانند به عناصر قبل و بعد از خودشان اشاره کنند و در هدر `<linux/type.h>` تعریف شده است.

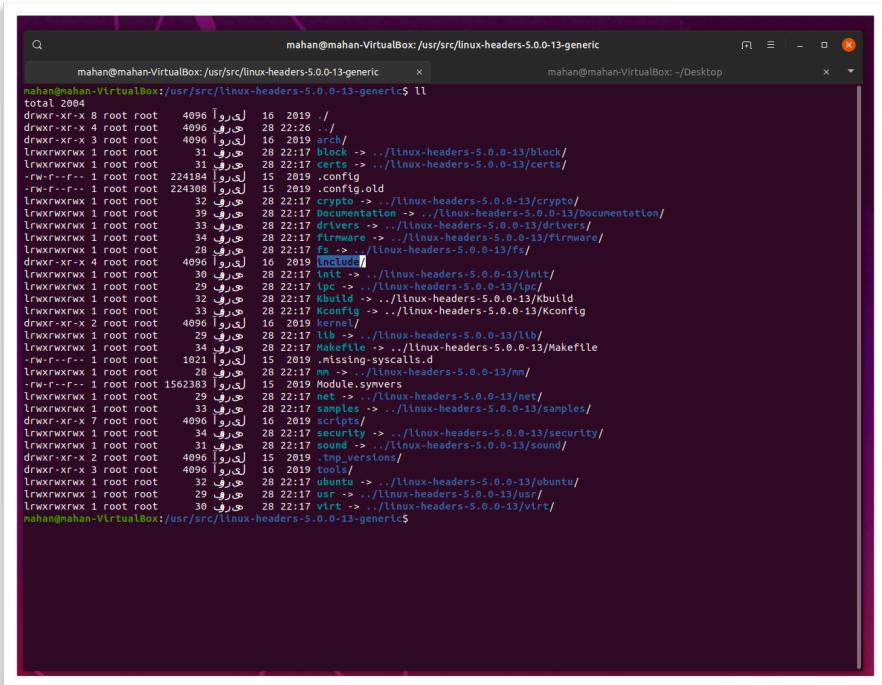
در قسمت اول گزارش کار گفته شد که به آدرس `/usr/src//linux-headers-5.0.0-13-generic` نیاز خواهیم داشت.

حال به این آدرس می رویم.



حال با استفاده از دستور `ll`، دایرکتوری های موجود در این آدرس را لیست می کنیم.

دستور `ll` در واقع alias command است برای `ls -Fla` واقع ا- در `ls` برای نشان دادن جزئیات بیشتر، a- برای نشان دادن فایل های مخفی و F-، کاراکتر / را در آخر دایرکتوری ها قرار می دهد.



```
mahan@mahan-VirtualBox: /usr/src/linux-headers-5.0.0-13-generic
mahan@mahan-VirtualBox: /usr/src/linux-headers-5.0.0-13-generic$ ll
total 2004
drwxr-xr-x 8 root root 4096 لبروآ 16 2019 .
drwxr-xr-x 4 root root 4096 لبروآ 28 22:26 ../
drwxr-xr-x 3 root root 4096 لبروآ 16 2019 arch/
lrwxrwxrwx 1 root root 31 لبروآ 28 22:17 block -> ../../linux-headers-5.0.0-13/block/
lrwxrwxrwx 1 root root 31 لبروآ 28 22:17 certs -> ../../linux-headers-5.0.0-13/certs/
-rw-r--r-- 1 root root 22184 لبروآ 16 2019 config
-rw-r--r-- 1 root root 2243648 لبروآ 15 2019 config.old
lrwxrwxrwx 1 root root 39 لبروآ 28 22:17 crypto -> ../../linux-headers-5.0.0-13/crypto/
lrwxrwxrwx 1 root root 39 لبروآ 28 22:17 Documentation -> ../../linux-headers-5.0.0-13/documentation/
lrwxrwxrwx 1 root root 33 لبروآ 28 22:17 drivers -> ../../linux-headers-5.0.0-13/drivers/
lrwxrwxrwx 1 root root 34 لبروآ 28 22:17 firmware -> ../../linux-headers-5.0.0-13/firmware/
lrwxrwxrwx 1 root root 28 لبروآ 28 22:17 fs -> ../../linux-headers-5.0.0-13/fs/
drwxr-xr-x 4 root root 4096 لبروآ 16 2019 include [include]
drwxr-xr-x 2 root root 4096 لبروآ 28 22:17 init -> ../../linux-headers-5.0.0-13/init/
lrwxrwxrwx 1 root root 39 لبروآ 28 22:17 ipc -> ../../linux-headers-5.0.0-13/ip/
lrwxrwxrwx 1 root root 32 لبروآ 28 22:17 Kbuild -> ../../linux-headers-5.0.0-13/Kbuild
lrwxrwxrwx 1 root root 33 لبروآ 28 22:17 Kconfig -> ../../linux-headers-5.0.0-13/Kconfig
drwxr-xr-x 2 root root 4096 لبروآ 16 2019 kernel/
lrwxrwxrwx 1 root root 29 لبروآ 28 22:17 lib -> ../../linux-headers-5.0.0-13/lib/
lrwxrwxrwx 1 root root 34 لبروآ 28 22:17 Makefile -> ../../linux-headers-5.0.0-13/Makefile
-rw-r--r-- 1 root root 1621 لبروآ 16 2019 missing-symlinks.d
lrwxrwxrwx 1 root root 28 لبروآ 28 22:17 mips -> ../../linux-headers-5.0.0-13/mips/
-rw-r--r-- 1 root root 1562383 لبروآ 15 2019 Module-Servers
lrwxrwxrwx 1 root root 29 لبروآ 28 22:17 net -> ../../linux-headers-5.0.0-13/net/
lrwxrwxrwx 1 root root 33 لبروآ 28 22:17 samples -> ../../linux-headers-5.0.0-13/samples/
drwxr-xr-x 7 root root 4096 لبروآ 16 2019 scripts/
lrwxrwxrwx 1 root root 34 لبروآ 28 22:17 security -> ../../linux-headers-5.0.0-13/security/
lrwxrwxrwx 1 root root 31 لبروآ 28 22:17 sound -> ../../linux-headers-5.0.0-13/sound/
drwxr-xr-x 3 root root 4096 لبروآ 16 2019 sysfs/
lrwxrwxrwx 1 root root 32 لبروآ 28 22:17 ubuntu -> ../../linux-headers-5.0.0-13/ubuntu/
lrwxrwxrwx 1 root root 29 لبروآ 28 22:17 usr -> ../../linux-headers-5.0.0-13/usr/
lrwxrwxrwx 1 root root 38 لبروآ 28 22:17 virt -> ../../linux-headers-5.0.0-13/virt/
mahan@mahan-VirtualBox: /usr/src/linux-headers-5.0.0-13-generic$
```

حال به دایرکتوری `include` می رویم و سپس به `linux` می رویم و سپس دستور `ll` را می زنیم.

```
mahan@mahan-VirtualBox:~/usr/src/linux-headers-5.0.0-13-generic$ cd include/linux
mahan@mahan-VirtualBox:~/usr/src/linux-headers-5.0.0-13-generic/include/linux$ ll
total 11620
-rwxr-xr-x 60 root root 36864 [لیرو]
-rwxr-xr-x 27 root root 4096 [لیرو]
-rw-r--r-- 1 root root 1038 سرمه
-rw-r--r-- 1 root root 2551 سرمه
-rw-r--r-- 1 root root 3300 سرمه
-rw-r--r-- 1 root root 39180 سرمه
-rw-r--r-- 1 root root 2416 سرمه
-rw-r--r-- 1 root root 674 سرمه
-rw-r--r-- 1 root root 1832 سرمه
-rw-r--r-- 1 root root 574 سرمه
-rw-r--r-- 1 root root 310 سرمه
-rw-r--r-- 1 root root 1772 سرمه
-rw-r--r-- 1 root root 3534 سرمه
-rw-r--r-- 1 root root 3988 سرمه
-rw-r--r-- 1 root root 1648 سرمه
-rw-r--r-- 1 root root 607 سرمه
-rw-r--r-- 1 root root 651 سرمه
-rw-r--r-- 1 root root 1878 سرمه
-rw-r--r-- 1 root root 9610 سرمه
-rw-r--r-- 1 root root 379 سرمه
-rw-r--r-- 1 root root 397 سرمه
-rw-r--r-x 2 root root 4096 [لیرو]
-rw-r--r-- 1 root root 6945 سرمه
-rw-r--r-- 1 root root 494 سرمه
-rw-r--r-- 1 root root 354 سرمه
-rw-r--r-- 1 root root 2746 سرمه
-rw-r--r-- 1 root root 1575 سرمه
-rw-r--r-- 1 root root 498 سرمه
-rw-r--r-- 1 root root 1459 سرمه
-rw-r--r-- 1 root root 855 سرمه
-rw-r--r-- 1 root root 2058 سرمه
-rw-r--r-- 1 root root 2653 [لیرو]
16 2019 ./ 
16 2019 ../
4 2019 acctch.h
4 2019 acpl_dma.h
4 2019 acpl.h
4 2019 acpi_iort.h
4 2019 acpi_pmtmr.h
4 2019 adb.h
4 2019 adfe_fs.h
4 2019 adxl.h
4 2019 aer.h
4 2019 agg_backend.h
4 2019 agggarth.h
4 2019 ahci_platform.h
4 2019 ahci_remap.h
4 2019 ato.h
4 2019 alarmitimer.h
4 2019 alcor_pcif.h
4 2019 altera_jtaguart.h
4 2019 altera_uart.h
16 2019 amba/
4 2019 and-iommu.h
4 2019 anon_inodes.h
4 2019 a.out.h
4 2019 apm_bios.h
4 2019 apm-emulation.h
4 2019 apple_bt.h
4 2019 apple_gmux.h
4 2019 arch_topology.h
4 2019 arm_cci.h
4 2019 arm_sdei.h
```

در واقع این فایل ها هدر های ما هستند.

حال با استفاده از دستور `grep list.h | ll` می توان

را مشاهده کرد، در واقع از دستور `ll` برای لیست کردن و از `grep` برای پیدا کردن فایل هایی که شامل `list.h` هستند، استفاده کرده ایم و با استفاده از دستور `gedit list.h` فایل را باز می کنیم.

```
mahan@mahan-VirtualBox:~/usr/src/linux-headers-5.0.0-13-generic/include/linux$ ll | grep list.h
mahan@mahan-VirtualBox:~/usr/src/linux-headers-5.0.0-13-generic/include/linux$ ll | grep list.h
mahan@mahan-VirtualBox:~/usr/src/linux-headers-5.0.0-13-generic/include/linux$ ll
total 1926
-rw-r--r-- 1 root root 9368 4 2019 list.h
-rw-r--r-- 1 root root 2186 4 2019 quicklist.h
-rw-r--r-- 1 root root 23315 4 2019 rCU.h
-rw-r--r-- 1 root root 13784 4 2019 scatterlist.h
mahan@mahan-VirtualBox:~/usr/src/linux-headers-5.0.0-13-generic/include/linux$ gedit list.h
```

```
list[Read Only]
=====
* SPDX-License-Identifier: GPL-2.0 *
#ifndef _LINUX_LIST_H_
#define _LINUX_LIST_H_

#include <linux/types.h>
#include <linux/stddef.h>
#include <linux/list.h>
#include <linux/init.h>
#include <linux/kernel.h>

/*
 * Single doubly linked list implementation.
 * Some of the internal functions (i.e., __) are useful when manipulating whole lists rather than single entries, as sometimes we already know the next/prev entries and we can manipulate them directly by using them directly rather than using the generic single-entry routines.
 */
#define LIST_HEAD_INIT(name) { __list_head, __list_head }

#define LIST_HEAD(name) \
    struct list_head name = LIST_HEAD_INIT(name)

static inline void INIT_LIST_HEAD(struct list_head *list)
{
    WRITE_ONCE(list->next, list);
    list->prev = list;
}

#ifndef CONFIG_DEBUG_LIST
extern bool __list_add_validate(struct list_head *new,
                               struct list_head *prev,
                               struct list_head *next);
extern bool __list_del_entry_validate(struct list_head *entry);
#endif
```

```
/* SPDX-License-Identifier: GPL-2.0 */
#ifndef LINUX_LIST_H
#define LINUX_LIST_H

#include <linux/types.h>
#include <linux/stddef.h>
#include <linux/poison.h>
#include <linux/const.h>
#include <linux/kernel.h>

/*
 * simple doubly linked list implementation.
 *
 * Some of the internal functions ("__xxx") are useful when
 * manipulating whole lists rather than single entries.  As
 * sometimes we already know the next/prev entries and we can
 * generate better code by using them directly rather than
 * using the generic single-entry routines.
 */

#define LIST_HEAD_INIT(name) { __name, __name }

#define __LIST_HEAD(name) \
    struct list_head name = LIST_HEAD_INIT(name)

static inline void INIT_LIST_HEAD(struct list_head *list)
{
    WRITE_ONCE(list->next, list);
    list->prev = list;
}

#ifndef CONFIG_DEBUG_LIST
extern bool __list_add_valid(struct list_head *new,
                           struct list_head *prev,
                           struct list_head *next);
extern bool __list_del_entry_valid(struct list_head *entry);
#else
static inline bool __list_add_valid(struct list_head *new,
                           struct list_head *prev,
                           struct list_head *next)
{
    return true;
}
static inline bool __list_del_entry_valid(struct list_head *entry)
{
    return true;
}
#endif

/*
 * Insert a new entry between two known consecutive entries.
 *
 * This is only for internal list manipulation where we know
 * the prev/next entries already!
 */

# define LIST_HEAD_INIT(name) { __name, __name }

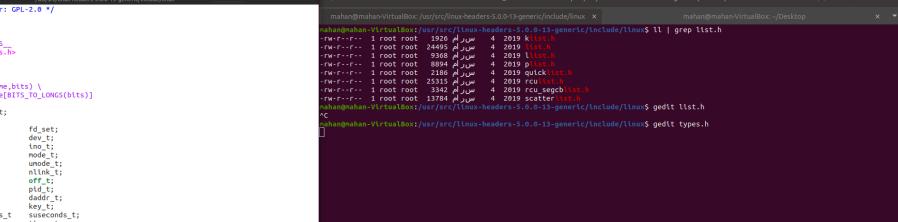
# define __LIST_HEAD(name) \
    struct list_head name = LIST_HEAD_INIT(name)

static inline void INIT_LIST_HEAD(struct list_head *list)
{
    WRITE_ONCE(list->next, list);
    list->prev = list;
}

#ifndef CONFIG_DEBUG_LIST
extern bool __list_add_valid(struct list_head *new,
                           struct list_head *prev,
                           struct list_head *next);
extern bool __list_del_entry_valid(struct list_head *entry);
#else
static inline bool __list_add_valid(struct list_head *new,
                           struct list_head *prev,
                           struct list_head *next)
{
    return true;
}
static inline bool __list_del_entry_valid(struct list_head *entry)
{
    return true;
}
#endif
```

و در واقع همانطور که می بینیم **List\_Head** ماکرویی است که در این فایل تعریف شده است.(در شکل بالا در فایل (list.h) حال می خواهیم سناریوی بالا را برای **struct list\_head** نیز اجرا کنیم و ببینیم که **struct list\_head** در کجا تعریف شده است.

کافیست با استفاده از دستور gedit types.h فایل types.h را باز کنیم.



The screenshot shows a terminal window with two panes. The left pane displays the C code for types.h, which includes definitions for various kernel data structures like fd\_set, dev\_t, mode\_t, and file operations. The right pane shows the command line interface where the user is navigating through the kernel headers directory and running commands like 'grep', 'ls', and 'gedit' to inspect and edit header files such as list.h and types.h.

```
SPDK [1]: types.h (Read-Only) mahan@mahan-VirtualBox: /usr/src/linux-headers-5.0.0-13-generic/include/linux
mahan@mahan-VirtualBox: /usr/src/linux-headers-5.0.0-13-generic/include/linux
mahan@mahan-VirtualBox: /usr/src/linux-headers-5.0.0-13-generic/include/linux$ ll | grep list.h
mahan@mahan-VirtualBox: /usr/src/linux-headers-5.0.0-13-generic/include/linux$ mahan@mahan-VirtualBox: ~/Desktop
mahan@mahan-VirtualBox: /usr/src/linux-headers-5.0.0-13-generic/include/linux$ ll
mahan@mahan-VirtualBox: /usr/src/linux-headers-5.0.0-13-generic/include/linux$ gedit list.h
mahan@mahan-VirtualBox: /usr/src/linux-headers-5.0.0-13-generic/include/linux$ gedit types.h
```

types.h [Read-Only] /usr/src/linux-headers-5.0.0-13-generic/include/linux

```
Open ▾ Save ▾ ⌂ ⌄ ⌍
```

```
typedef phys_addr_t resource_size_t;

/*
 * This type is the placeholder for a hardware interrupt number. It has to be
 * big enough to enclose whatever representation is used by a given platform.
 */
typedef unsigned long irq_hw_number_t;

typedef struct {
    int counter;
} atomic_t;

#ifndef CONFIG_64BIT
typedef struct {
    long counter;
} atomic64_t;
#endif

struct list_head {
    struct list_head *next, *prev;
};

struct hlist_head {
    struct hlist_node *first;
};

struct hlist_node {
    struct hlist_node *next, **pprev;
};

struct ustat {
    __kernel_daddr_t      f_tfree;
    __kernel_lno_t        f_tinode;
    char                 f_fname[6];
    char                 f_fpack[6];
};
```

C/ObjC Header ▾ Tab Width: 8 ▾ Ln 186, Col 1 ▾ INS

حال مشاهده می کنیم که `struct list_head` در فایل بالا تعریف شده است و با دو اشاره گر به خودش اشاره می کند،

در واقع این struct به ما اجازه می دهد که لیست پیوندی دوطرفه (دوطرفه یعنی هم به قبل و هم به بعد از خودش اشاره کند) داشته باشیم که این کار با تعریف struct از نوع list\_head برای ما امکانپذیر است.

حال یک تابع به اسم add تعریف می کنیم و محتویات

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

را در این تابع قرار می دهیم، حال باید نمونه هایی از struct birthday ایجاد کنیم. (طبق گفته‌ی سوال باید 5 عنصر struct birthday ایجاد کنیم)  
در شکل صفحه‌ی بعد می توانیم محتوای تابع add را ببینیم.

همانطور که مشاهده می کنیم در این تابع 5 عنصر struct birthday ایجاد کرده ایم.

سپس با استفاده از دستور

```
struct birthday *ptr;  
list_for_each_entry(ptr, &birthday_list, list) {  
/*on each iteration ptr points to the next birthday struct*/  
}
```

لیست پیوند را پیمایش کرده ایم و محتوای لیست پیوند را در کرنل چاپ کرده ایم.

```
1 int add_birth(void){  
2     static LIST_HEAD(birthday_list);  
3  
4     struct birthday {  
5         int day;  
6         int month;  
7         int year;  
8         struct list_head list;  
9     };  
10  
11    struct birthday *person = NULL;  
12  
13    person = kmalloc(sizeof(*person), GFP_KERNEL);  
14    person->day = 2;  
15    person->month = 8;  
16    person->year = 1995;  
17    INIT_LIST_HEAD(&person->list);  
18    list_add_tail(&person->list, &birthday_list);  
19  
20    person = kmalloc(sizeof(*person), GFP_KERNEL);  
21    person->day = 21;  
22    person->month = 6;  
23    person->year = 1998;  
24    INIT_LIST_HEAD(&person->list);  
25    list_add_tail(&person->list, &birthday_list);  
26  
27    person = kmalloc(sizeof(*person), GFP_KERNEL);  
28    person->day = 12;  
29    person->month = 4;  
30    person->year = 1999;  
31    INIT_LIST_HEAD(&person->list);  
32    list_add_tail(&person->list, &birthday_list);  
33  
34    person = kmalloc(sizeof(*person), GFP_KERNEL);  
35    person->day = 11;  
36    person->month = 3;  
37    person->year = 2000;  
38    INIT_LIST_HEAD(&person->list);  
39    list_add_tail(&person->list, &birthday_list);  
40  
41    person = kmalloc(sizeof(*person), GFP_KERNEL);  
42    person->day = 9;  
43    person->month = 4;  
44    person->year = 2001;  
45    INIT_LIST_HEAD(&person->list);  
46    list_add_tail(&person->list, &birthday_list);  
47  
48    struct birthday *ptr;  
49    list_for_each_entry(ptr, &birthday_list, list) {  
50        /*on each iteration ptr points to the next birthday struct*/  
51  
52        printk(KERN_INFO "DATA: %2d/%2d/%4d\n", ptr->day, ptr->month, ptr->year);  
53    }  
54}
```

تابع `list_for_each_entry` در واقع تمام `entry` های که در لیست وجود دارد را لیست می کند و آن ها را یکی یکی در متغیر `ptr` قرار می دهد و سپس در `body` تابع می توانیم با این متغیر کار دلخواه خود را انجام دهیم.

حال ماثولی را که تا به اینجا ساخته ایم را با استفاده از دستور `make` کامپایل کرده و سپس با استفاده از دستور `sudo insmod simplemod.ko` این ماثول را به هسته اضافه می کنیم.

```
mahan@mahan-VirtualBox: ~/Desktop$ make
make -C /lib/modules/5.0.0-13-generic/build M=/home/mahan/Desktop modules
make[1]: Entering directory '/usr/src/linux-headers-5.0.0-13-generic'
  Building modules, stage 2.
    MODPOST 1 modules
make[1]: Leaving directory '/usr/src/linux-headers-5.0.0-13-generic'
mahan@mahan-VirtualBox: ~/Desktop$ sudo insmod simplemod.ko
mahan@mahan-VirtualBox: ~/Desktop$ lsmod
Module           Size  Used by
simplemod        16384  0
vboxsf          81920  1
vboxvideo        36864  0
snd_intel8x0     45056  2
snd_ac97_codec   135168  1 snd_intel8x0
ac97_bus         16384  1 snd_ac97_codec
snd_pcm          102400  2 snd_intel8x0,snd_ac97_codec
snd_seq_midi      20480  0
snd_seq_midi_event 16384  1 snd_seq_midi
snd_rawmidi       36864  1 snd_seq_midi
crc32_pclmul     16384  1
crc32_pclmul     16384  0
ghash_clmulni_intel 16384  0
snd_seq            69632  2 snd_seq_midi,snd_seq_midi_event
snd_seq_device    16384  3 snd_seq,snd_seq_midi,snd_rawmidi
aesni_intel       372736  0
snd_timer         36864  2 snd_seq,snd_pcm
aes_x86_64        20480  1 aesni_intel
joydev            24576  0
crypto_simd       16384  1 aesni_intel
cryptd            24576  3 crypto_simd,ghash_clmulni_intel,aesni_intel
glue_helper       16384  1 aesni_intel
intel_rapl_perf    16384  0
input_leds         16384  0
snd                81920  11 snd_seq,snd_seq_device,snd_intel8x0,snd_timer,snd_ac97_codec,snd_pcm,snd_rawmidi
vmwgfx            29816  2
ttm                102400  2 vmwgfx,vboxvideo
serio_raw          20480  0
drm_kms_helper    180224  2 vmwgfx,vboxvideo
drm                475136  6 vmwgfx,drm_kms_helper,vboxvideo,ttm
fb_sys_fops        16384  1 drm_kms_helper
syscopyarea        16384  1 drm_kms_helper
sysfillrect        16384  1 drm_kms_helper
sysqfillbit        16384  1 drm_kms_helper
soundcore          16384  1 snd
vboxguest         33968  6 vboxsf
mac_hid            16384  0
sch fq_codel       20480  2
```

```
mahan@mahan-VirtualBox: ~/Desktop
```

```
[mahn@mahn-VirtualBox: ~/Desktop] Code: 55 49 89 c1 48 8d 35 45 e5 02 00 48 89 c3 4c 8d 05 f3 e5 02 00 31 c0 bf 00 04 00 00 e8 bf 38 fe ff 48 8b 05 d0 d9  
[mahn@mahn-VirtualBox: ~/Desktop] DATA: 55 49 89 c1 48 8d 35 45 e5 02 00 48 89 c3 4c 8d 05 f3 e5 02 00 31 c0 bf 00 04 00 00 e8 bf 38 fe ff 48 8b 05 d0 d9  
[mahn@mahn-VirtualBox: ~/Desktop] DATA: 30 85 f6 0f 84 dc 00 00 48 8b 78 08 48 89 de e8 00 0a fe ff  
[mahn@mahn-VirtualBox: ~/Desktop] DATA: 2 / 6/1995  
[mahn@mahn-VirtualBox: ~/Desktop] DATA: 21 / 6/1998  
[mahn@mahn-VirtualBox: ~/Desktop] Removing Module  
[43279. 863685] hrtimer: interrupt took 5843457 ns  
[43258. 166946] usb 2-1: USB disconnect, device number 6  
[43258. 236270] 22:00:29.413564 control Session 0 is about to close ...  
[43258. 238521] 22:00:29.427549 control Stopping all guest processes ...  
[43258. 238521] 22:00:29.427559 control Unmounting all guest files ...  
[43258. 238521] 22:00:29.427559 control vhd130guestfiledetectPeekGetCancelSupport: Supported (#1)  
[43258. 807866] usb 2-1: new full-speed USB device number 7 using ohci-pci  
[43259. 407940] e1000: enp0s3 NIC Link is Down  
[43259. 407859] e1000: 0000:00:03:0 enp0s3: Reset adapter  
[43259. 569731] usb 2-1: New USB device found, idVendor=80ee, idProduct=0021, bcdDevice= 1.00  
[43259. 569735] usb 2-1: New USB device strings: Mfr=1, Product=3, SerialNumber=0  
[43259. 569736] usb 2-1: Product: USB Tablet  
[43259. 569737] usb 2-1: Manufacturer: VirtualBox  
[43259. 797708] Input: VirtualBox USB Tablet as /devices/pci0000:00/0000:00:06.0/usb2/2-1/1.0/0003:80EE:0021.0006/input/input12  
[43259. 858087] hid-generic 0003:80EE:0021:0006: input_hidraw: USB HID v1.10 Mouse [VirtualBox USB Tablet] on usb-0000:00:06.0-1/input0  
[43261. 519023] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX  
[43262. 312234] 05:51:09.230094 timesync vgsvt!meSyncWorker: Radical host time change: 28 245 697 000 000ns (HostNow=1 617 256 269 228  
000 000 ns GuestLast=1 617 228 023 531 000 000 ns)  
[43272. 315584] 05:51:19.233454 timesync vgsvt!meSyncWorker: Radical guest time change: 28 245 732 162 000ns (GuestNow=1 617 256 279 23  
3 432 000 ns GuestLast=1 617 228 033 501 270 000 ns fsetTimeLastLoop=true )  
mahan@mahn-VirtualBox: ~/Desktop$
```

با استفاده از دستور **dmesg** می بینیم که با لود شدن ماژول در هسته، لیست به درستی ایجاد شده است.

```
mahan@mahan-VirtualBox: ~/Desktop
```

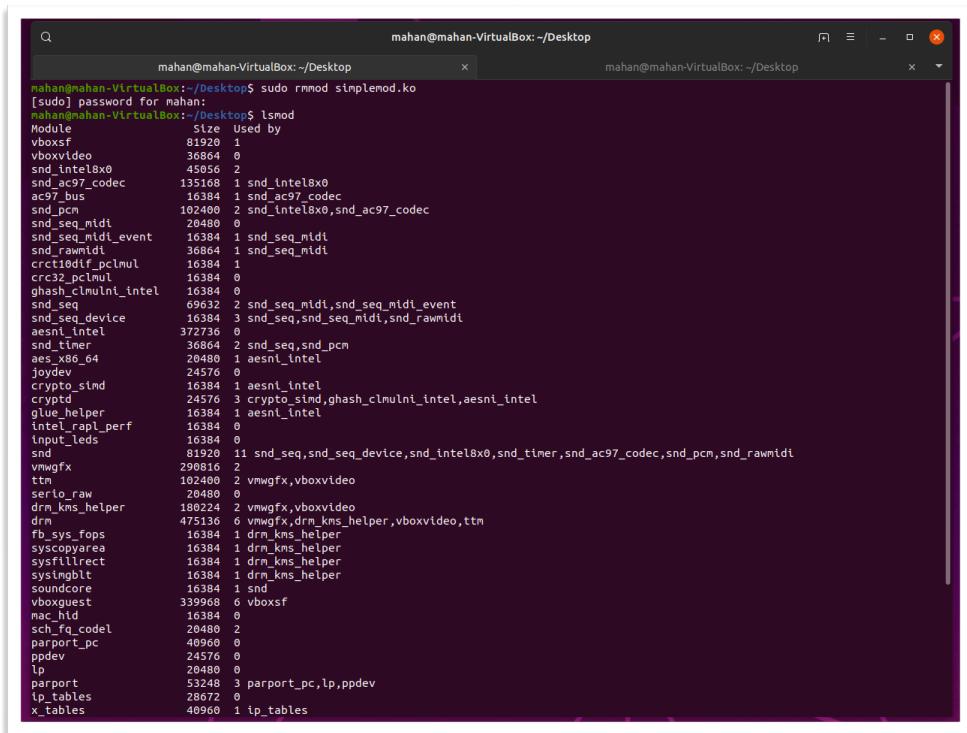
```
mahan@mahan-VirtualBox: ~/Desktop
```

```
Apr 1 10:47:36 mahan-VirtualBox kernel: [44849.094095] Removing Module
Apr 1 10:47:40 mahan-VirtualBox dbus-daemon[1373]: [session uid=1000 pid=1373] Activating via systemd: service name='org.freedesktop.Tracker1.Miner.Extract' unit='tracker-extract.service' requested by ':1.54' (uid=1000 pid=1792 comm="/usr/lib/tracker/tracker-miner-fs " label="unconfined")
Apr 1 10:47:40 mahan-VirtualBox systemd[1340]: Starting Tracker metadata extractor...
Apr 1 10:47:40 mahan-VirtualBox dbus-daemon[1373]: [session uid=1000 pid=1373] Successfully activated service 'org.freedesktop.Tracker1.Miner.Extract'
Apr 1 10:47:40 mahan-VirtualBox systemd[1340]: Started Tracker metadata extractor.
Apr 1 10:48:00 mahan-VirtualBox systemd[1340]: tracker-extract.service: Succeeded.
Apr 1 10:48:00 mahan-VirtualBox dbus-daemon[1373]: [session uid=1000 pid=1373] Activating via systemd: service name='org.freedesktop.Tracker1.Miner.Extract' unit='tracker-extract.service' requested by ':1.54' (uid=1000 pid=1792 comm="/usr/lib/tracker/tracker-miner-fs " label="unconfined")
Apr 1 10:48:00 mahan-VirtualBox systemd[1340]: Starting Tracker metadata extractor...
Apr 1 10:48:01 mahan-VirtualBox dbus-daemon[1373]: [session uid=1000 pid=1373] Successfully activated service 'org.freedesktop.Tracker1.Miner.Extract'
Apr 1 10:48:10 mahan-VirtualBox systemd[1340]: Started Tracker metadata extractor.
Apr 1 10:48:35 mahan-VirtualBox dbus-daemon[1373]: [session uid=1000 pid=1373] Activating via systemd: service name='org.freedesktop.Tracker1.Miner.Extract' unit='tracker-extract.service' requested by ':1.54' (uid=1000 pid=1792 comm="/usr/lib/tracker/tracker-miner-fs " label="unconfined")
Apr 1 10:48:35 mahan-VirtualBox systemd[1340]: Starting Tracker metadata extractor...
Apr 1 10:48:35 mahan-VirtualBox dbus-daemon[1373]: [session uid=1000 pid=1373] Successfully activated service 'org.freedesktop.Tracker1.Miner.Extract'
Apr 1 10:48:35 mahan-VirtualBox systemd[1340]: Started Tracker metadata extractor.
Apr 1 10:48:45 mahan-VirtualBox systemd[1340]: tracker-extract.service: Succeeded.
Apr 1 10:49:03 mahan-VirtualBox kernel: [44936.980446] Loading Module
Apr 1 10:49:03 mahan-VirtualBox kernel: [44936.980448] DATA: 2/ 8/1995
Apr 1 10:49:03 mahan-VirtualBox kernel: [44936.980449] DATA: 21/ 6/1998
Apr 1 10:49:03 mahan-VirtualBox kernel: [44936.980449] DATA: 12/ 4/1999
Apr 1 10:49:03 mahan-VirtualBox kernel: [44936.980450] DATA: 11/ 3/2000
Apr 1 10:49:03 mahan-VirtualBox kernel: [44936.980450] DATA: 9/ 4/2001
Apr 1 10:49:31 mahan-VirtualBox kernel: [44965.047327] Removing Module
Apr 1 10:51:54 mahan-VirtualBox dbus-daemon[1373]: [session uid=1000 pid=1373] Activating via systemd: service name='org.freedesktop.Tracker1.Miner.Extract' unit='tracker-extract.service' requested by ':1.54' (uid=1000 pid=1792 comm="/usr/lib/tracker/tracker-miner-fs " label="unconfined")
Apr 1 10:51:54 mahan-VirtualBox systemd[1340]: Starting Tracker metadata extractor...
Apr 1 10:51:54 mahan-VirtualBox dbus-daemon[1373]: [session uid=1000 pid=1373] Successfully activated service 'org.freedesktop.Tracker1.Miner.Extract'
Apr 1 10:51:54 mahan-VirtualBox systemd[1340]: Started Tracker metadata extractor.
Apr 1 10:52:04 mahan-VirtualBox systemd[1340]: tracker-extract.service: Succeeded.
Apr 1 10:52:18 mahan-VirtualBox kernel: [45131.695885] Loading Module
Apr 1 10:52:18 mahan-VirtualBox kernel: [45131.695887] DATA: 2/ 8/1995
Apr 1 10:52:18 mahan-VirtualBox kernel: [45131.695887] DATA: 21/ 6/1998
Apr 1 10:52:18 mahan-VirtualBox kernel: [45131.695888] DATA: 12/ 4/1999
Apr 1 10:52:18 mahan-VirtualBox kernel: [45131.695888] DATA: 11/ 3/2000
Apr 1 10:52:18 mahan-VirtualBox kernel: [45131.695889] DATA: 9/ 4/2001
```

همچنین لاگ سیستم را نیز زیرنظر داشته ایم و می بینیم که خروجی دلخواه را گرفته ایم.

حال می خواهیم تابع `remove_birth()` را پیاده سازی کنیم.

ابتدا ماژول لود شده را حذف می کنیم، زیرا می خواهیم ماژول جدیدی با قابلیت جدید به هسته اضافه کنیم.



```
mahan@mahan-VirtualBox: ~/Desktop
mahan@mahan-VirtualBox: ~/Desktop
```

```
mahan@mahan-VirtualBox: ~/Desktop$ sudo rmmod simplemod.ko
[sudo] password for mahan:
mahan@mahan-VirtualBox: ~/Desktop$ lsmod
Module           Size  Used by
vboxsf          81920  1
vboxvideo        36864  0
snd_intel8x0    45056  2
snd_ac97_codec   135168  1 snd_intel8x0
ac97_bus         16384  1 snd_ac97_codec
snd_pcm          102400  2 snd_intel8x0,snd_ac97_codec
snd_seq_midi     26480  0
snd_seq_midi_event 16384  1 snd_seq_midi
snd_rawmidi      36864  1 snd_seq_midi
crcr10dif_pcml  16384  1
crc32_pcml     16384  0
ghash_clmulni_intel 16384  0
snd_seq          16384  2 snd_seq_midi,snd_seq_midi_event
snd_seq_device   16384  3 snd_seq,snd_seq_midi,snd_rawmidi
aesnl_intel     372726  0
snd_timer        36864  2 snd_seq,snd_pcm
aes_x86_64       20480  1 aesnl_intel
joydev          24576  0
crypto_simd     16384  1 aesnl_intel
cryptd          24576  3 crypto_simd,ghash_clmulni_intel,aesni_intel
glue_helper     16384  1 aesnl_intel
intel_rapl_perf 16384  0
input_leds       16384  0
snd             81920  11 snd_seq,snd_seq_device,snd_intel8x0,snd_timer,snd_ac97_codec,snd_pcm,snd_rawmidi
vmwgfx          296816  2
ttm            102400  2 vmwgfx,vboxvideo
serio_raw        20480  0
drm_kms_helper  186224  2 vmwgfx,vboxvideo
drm            475136  6 drm_kms_helper,vmwgfx,drm_kms_helper,vboxvideo,ttm
fb_sys_fops     16384  1 drm_kms_helper
syscopyarea     16384  1 drm_kms_helper
sysfillrect     16384  1 drm_kms_helper
sysingblt       16384  1 drm_kms_helper
soundcore       16384  1 snd
vboxguest       339968  6 vboxsf
mac_hid          16384  0
sch_tq_codel    20480  2
parport_pc      40960  0
ppdev          24576  0
lp              20480  0
parport         53248  3 parport_pc,lp,ppdev
lp_tables       28672  0
x_tables        40960  1 ip_tables
```

با استفاده از دستور `sudo rmmod simplemod.ko` ماژول `simplemod` را حذف کرده ایم و سپس با دستور `lsmod` مشاهده می کنیم که این ماژول دیگر در لیست ماژول های کرنل وجود ندارد.

حال تابع `remove_birth()` را تعریف می کنیم.

قبل از آن محتویات

```
static List_Head(birthday_list);
```

و

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

را به بیرون از تابع `add` می بریم زیرا می خواهیم از این  
در تابع `remove` استفاده کنیم.  
حال برای `remove` باید از پیمایش به روش `safe` استفاده  
کنیم و برای این کار از دستور `list_for_each_safe` استفاده  
کرده ایم.

به این صورت که محتویات `struct` را `remove` کرده ایم و  
سپس حافظه را از آن پس گرفته ایم.

اما به چه دلیل از دستور `list_for_each_safe` استفاده  
کرده ایم؟ نمی توانستیم از دستور معمولی `list_for_each`  
برای `remove` کردن استفاده کنیم؛ پاسخ منفی است، زیرا بعد  
از اجرای یکبار `body` دیگر پوینتر `ptr`، `invalid` خواهد بود و  
شرط `pos = pos->next` را می شکند. (به دلیل وجود  
`(body) list_del(ptr)`  
اما `list_for_each_safe` در واقع مقدار `pos->next` را  
در یک متغیر `temporary` ذخیره می کند.

```

int remove_birth(void){
    struct birthday *tmp;
    struct list_head *ptr, *n;

    if (list_empty(&birthday_list)) {
        return 0;
    }

    list_for_each_safe(ptr, n, &birthday_list){
        tmp = list_entry(ptr, struct birthday, list);
        list_del(ptr);
        kfree(tmp);
    }

    return 0;
}

/* this function is called when the module is loaded*/
int simple_init(void)
{
    printk(KERN_INFO "Loading Module\n");
    add_birth();
    return 0;
}

/* this function is called when the module is removed*/
void simple_exit(void)
{
    remove_birth();
    printk(KERN_INFO "Removing Module\n");
}

```

همچنین در ابتدا چک کرده ایم که اگر لیست خالی بود از تابع بیرون بیاییم، هرچند نیازی به این کار نبود:)).

حال ماژول کامل شده‌ی خود را با دستور make کامپایل می‌کنیم، سپس با دستور sudo insmod simplemod.ko آن را به هسته اضافه می‌کنیم و سپس با دستور lsmod چک می‌کنیم که ماژول مورد نظر اضافه شده باشد.

مراحل گفته شده را در شکل زیر مشاهده می‌کنیم.

```
mahan@mahan-VirtualBox:~/Desktop$ make
make -C /lib/modules/5.0.0-13-generic/build M=/home/mahan/Desktop modules
make[1]: Entering directory '/usr/src/linux-headers-5.0.0-13-generic'
  Building modules, stage 2.
MODPOST 1 modules
make[1]: Leaving directory '/usr/src/linux-headers-5.0.0-13-generic'
mahan@mahan-VirtualBox:~/Desktop$ sudo insmod simplemod.ko
mahan@mahan-VirtualBox:~/Desktop$ lsmod
Module           Size  Used by
simplemod        16384  0
vboxsf          81920  1
vboxvideo        36864  0
nd_intel8x0      45056  2
snd_ac97_codec   135168  1 snd_intel8x0
c97_bus          16384  1 snd_ac97_codec
snd_pcm          102400  2 snd_intel8x0,snd_ac97_codec
snd_seq_midi      20480  0
snd_seq_midi_event 16384  1 snd_seq_midi
snd_rawmidi       36864  1 snd_seq_midi
rcti0dif_pclmul  16384  1
crc32_pclmul     16384  0
hash_clmulni_intel 16384  0
snd_seq            69632  2 snd_seq_midi,snd_seq_midi_event
snd_seq_device    16384  3 snd_seq,snd_seq_midi,snd_rawmidi
aesni_intel       372736  0
snd_timer         36864  2 snd_seq,snd_pcm
es_x86_64         20480  1 aesni_intel
oydev             24576  0
crypto_simd       16384  1 aesni_intel
ryptd             24576  3 crypto_simd,ghash_clmulni_intel,aesni_intel
lue_helper        16384  1 aesni_intel
intel_rapl_perf   16384  0
input_leds         16384  0
nd                81920  11 snd_seq,snd_seq_device,snd_intel8x0,snd_timer,snd_ac97_codec,snd_pcm,snd_rawmidi
vmwgfx           290816  2
tm                102400  2 vmwgfx,vboxvideo
erio_raw          20480  0
rm_kms_helper    180224  2 vmwgfx,vboxvideo
rm                475136  6 vmwgfx,drm_kms_helper,vboxvideo,ttm
b_sys_fops        16384  1 drm_kms_helper
yscopyarea        16384  1 drm_kms_helper
ysfillrect        16384  1 drm_kms_helper
ysingblt         16384  1 drm_kms_helper
oundcore          16384  1 snd
boxquest          339968  6 vboxsf
ac_hd              16384  0
sch_tq_code1      20480  2
parport_pc        40960  0
pudev             24576  0
lp                20480  0
parport          53248  3 parport_pc,lp,ppdev
lp_tables         28672  0
_ip_tables        40960  1 ip_tables
utofs4            45056  2
hid_generic       16384  0
sbhid             53248  0
hid               126976  2 usbhid,hid_generic
smouse            151552  0
hci                40960  1
ibahci            32768  1 ahci
zc_pti4x          28672  0
i1000             139264  0
ata_acpi          16384  0
video              45056  0
mahan@mahan-VirtualBox:~/Desktop$
```

حال لاگ سیستم را چک میکنیم که مطمئن شویم لیست به درستی ایجاد شده است، همچنین برای این کار از دستور dmesg نیز استفاده می کنیم.

```
mahan@mahan-VirtualBox: ~/Desktop
```

```
mahan@mahan-VirtualBox: ~/Desktop
```

```
Apr 1 12:55:59 mahan-VirtualBox org.gnome.Shell.desktop[1575]: Window manager warning: Overwriting existing binding of keysym 32 with keysym 32 (keycode b).
Apr 1 12:55:59 mahan-VirtualBox org.gnome.Shell.desktop[1575]: Window manager warning: Overwriting existing binding of keysym 33 with keysym 33 (keycode c).
Apr 1 12:55:59 mahan-VirtualBox org.gnome.Shell.desktop[1575]: Window manager warning: Overwriting existing binding of keysym 34 with keysym 34 (keycode d).
Apr 1 12:55:59 mahan-VirtualBox org.gnome.Shell.desktop[1575]: Window manager warning: Overwriting existing binding of keysym 35 with keysym 35 (keycode e).
Apr 1 12:55:59 mahan-VirtualBox org.gnome.Shell.desktop[1575]: Window manager warning: Overwriting existing binding of keysym 36 with keysym 36 (keycode f).
Apr 1 12:55:59 mahan-VirtualBox org.gnome.Shell.desktop[1575]: Window manager warning: Overwriting existing binding of keysym 37 with keysym 37 (keycode 10).
Apr 1 12:56:05 mahan-VirtualBox dbus-daemon[1373]: [session uid=1000 pid=1373] Activating via systemd: service name='org.freedesktop.Tracker1.Miner.Extract' unit='tracker-extract.service' requested by ':1.54' (uid=1000 pid=1792 comm="/usr/lib/tracker/tracker-miner-fs" label="unconfined")
Apr 1 12:56:05 mahan-VirtualBox systemd[1340]: Starting Tracker metadata extractor...
Apr 1 12:56:05 mahan-VirtualBox dbus-daemon[1373]: [session uid=1000 pid=1373] Successfully activated service 'org.freedesktop.Tracker1.Miner.Extract'
Apr 1 12:56:05 mahan-VirtualBox systemd[1340]: Started Tracker metadata extractor.
Apr 1 12:56:19 mahan-VirtualBox systemd[1340]: tracker-extract.service: Succeeded.
Apr 1 12:56:24 mahan-VirtualBox systemd[1]: fprintd.service: Succeeded.
Apr 1 12:56:38 mahan-VirtualBox dbus-daemon[1373]: [session uid=1000 pid=1373] Activating via systemd: service name='org.freedesktop.Tracker1.Miner.Extract' unit='tracker-extract.service' requested by ':1.54' (uid=1000 pid=1792 comm="/usr/lib/tracker/tracker-miner-fs" label="unconfined")
Apr 1 12:56:38 mahan-VirtualBox systemd[1340]: Starting Tracker metadata extractor...
Apr 1 12:56:38 mahan-VirtualBox dbus-daemon[1373]: [session uid=1000 pid=1373] Successfully activated service 'org.freedesktop.Tracker1.Miner.Extract'
Apr 1 12:56:38 mahan-VirtualBox systemd[1340]: Started Tracker metadata extractor.
Apr 1 12:56:48 mahan-VirtualBox systemd[1340]: tracker-extract.service: Succeeded.
Apr 1 12:57:04 mahan-VirtualBox dbus-daemon[1373]: [session uid=1000 pid=1373] Activating via systemd: service name='org.freedesktop.Tracker1.Miner.Extract' unit='tracker-extract.service' requested by ':1.54' (uid=1000 pid=1792 comm="/usr/lib/tracker/tracker-miner-fs" label="unconfined")
Apr 1 12:57:04 mahan-VirtualBox systemd[1340]: Starting Tracker metadata extractor...
Apr 1 12:57:04 mahan-VirtualBox dbus-daemon[1373]: [session uid=1000 pid=1373] Successfully activated service 'org.freedesktop.Tracker1.Miner.Extract'
Apr 1 12:57:04 mahan-VirtualBox systemd[1340]: Started Tracker metadata extractor.
Apr 1 12:57:14 mahan-VirtualBox systemd[1340]: tracker-extract.service: Succeeded.
Apr 1 12:57:39 mahan-VirtualBox kernel: [52652.410293] Loading Module
Apr 1 12:57:39 mahan-VirtualBox kernel: [52652.410295] DATA: 2/ 8/1995
Apr 1 12:57:39 mahan-VirtualBox kernel: [52652.410296] DATA: 21/ 6/1998
Apr 1 12:57:39 mahan-VirtualBox kernel: [52652.410297] DATA: 12/ 4/1999
Apr 1 12:57:39 mahan-VirtualBox kernel: [52652.410298] DATA: 11/ 3/2000
Apr 1 12:57:39 mahan-VirtualBox kernel: [52652.410298] DATA: 9/ 4/2001
```

```
mahan@mahan-VirtualBox: ~/Desktop
```

```
[43800..301571] DATA: 2/ 8/1995
[43800..301572] DATA: 21/ 6/1998
[43800..301573] DATA: 12/ 4/1999
[43800..301573] DATA: 11/ 3/2000
[43800..301574] DATA: 9/ 4/2001
[44849..094095] Removing Module
[44936..980446] Loading Module
[44936..980448] DATA: 2/ 8/1995
[44936..980449] DATA: 21/ 6/1998
[44936..980449] DATA: 12/ 4/1999
[44936..980450] DATA: 11/ 3/2000
[44936..980450] DATA: 9/ 4/2001
[44965..947324] Removing Module
[45131..523395] Loading Module
[45131..523397] DATA: 2/ 8/1995
[45131..523397] DATA: 21/ 6/1998
[45131..523398] DATA: 12/ 4/1999
[45131..523398] DATA: 11/ 3/2000
[45131..523399] DATA: 9/ 4/2001
[49085..523397] Removing Module
[49152..329805] Loading Module
[49152..329807] DATA: 2/ 8/1995
[49152..329808] DATA: 21/ 6/1998
[49152..329808] DATA: 12/ 4/1999
[49152..329809] DATA: 11/ 3/2000
[49152..329809] DATA: 9/ 4/2001
[49180..758605] Removing Module
[50143..005266] Loading Module
[50143..005268] DATA: 2/ 8/1995
[50143..005269] DATA: 21/ 6/1998
[50143..005269] DATA: 12/ 4/1999
[50143..005270] DATA: 11/ 3/2000
[50143..005270] DATA: 9/ 4/2001
[50165..226340] DATA: 2/ 8/1995
[50165..226342] DATA: 21/ 6/1998
[50165..226343] DATA: 12/ 4/1999
[50165..226344] DATA: 11/ 3/2000
[50165..226345] DATA: 9/ 4/2001
[50165..226346] Removing Module
[52652..410293] Loading Module
[52652..410295] DATA: 2/ 8/1995
[52652..410296] DATA: 21/ 6/1998
[52652..410297] DATA: 12/ 4/1999
[52652..410298] DATA: 11/ 3/2000
[52652..410298] DATA: 9/ 4/2001
mahan@mahan-VirtualBox: ~/Desktop$
```

حال با دستور sudo rmmod simplemod.ko مارژول خود را  
می کنیم و با دستور lsmod اچک می کنیم که  
remove شده باشد.

```
mahan@mahan-VirtualBox:~/Desktop$ sudo rmmod simplemod.ko
mahan@mahan-VirtualBox:~/Desktop$ lsmod
Module           Size Used by
vboxsf          81920  1
vboxvideo        36864  0
snd_intel8x0    45856  2
snd_ac97_codec   135168  1 snd_intel8x0
ac97_bus         16384  1 snd_ac97_codec
snd_pcm          102400  2 snd_intel8x0,snd_ac97_codec
snd_seq_midi     20480  0
snd_seq_midi_event 16384  1 snd_seq_midi
snd_rawmidi      36864  1 snd_seq_midi
crc32_pcmul     16384  1
crc32_pcmul     16384  0
ghash_clmulni_intel 16384  0
snd_seq          69632  2 snd_seq_midi,snd_seq_midi_event
snd_seq_device   16384  3 snd_seq,snd_seq_midi,snd_rawmidi
aesnl_intel     372736  0
snd_timer        36864  2 snd_seq,snd_pcm
aes_x86_64       29488  1 aesnl_intel
joydev          24576  0
crypto_simd     16384  1 aesnl_intel
cryptd          24576  3 crypto_simd,ghash_clmulni_intel,aesnl_intel
glue_helper     16384  1 aesnl_intel
intel_rapl_perf 16384  0
input_leds       16384  0
snd             81920  11 snd_seq,snd_seq_device,snd_intel8x0,snd_timer,snd_ac97_codec,snd_pcm,snd_rawmidi
vmwgfx          290816  2
ttm             102400  2 vmwgfx,vboxvideo
serio_raw       20480  0
drm_kms_helper  180224  2 vmwgfx,vboxvideo
drm            475136  6 vmwgfx,drm_kms_helper,vboxvideo,ttm
fb_sys_fops     16384  1 drm_kms_helper
syscopyarea     16384  1 drm_kms_helper
sysfillrect     16384  1 drm_kms_helper
sysmblt         16384  1 drm_kms_helper
soundcore       16384  1 snd
vboxguest      339968  6 vboxsf
mac_hid          16384  0
sch fq_codel    20480  2
parport_pc      40960  0
ppdev          24576  0
lp              20480  0
parport         53248  3 parport_pc,lp,ppdev
ip_tables       28672  0
x_tables        40960  1 ip_tables
```

حال لاغ سیستم را چک میکنیم که مطمئن شویم لیست به درستی حذف شده است، همچنین برای این کار از دستور dmesg نیز استفاده می کنیم.

Q

mahan@mahan-VirtualBox: ~/Desktop

mahan@mahan-VirtualBox: ~/Desktop

```
mahan@mahan-VirtualBox: ~/Desktop x mahan@mahan-VirtualBox: ~/Desktop x
Apr 1 12:56:38 mahan-VirtualBox dbus-daemon[1373]: [session uid=1000 pid=1373] Activating via systemd: service name='org.freedesktop.Tracker1.Miner.Extract' unit='tracker-extract.service' requested by '':1.54' (uid=1000 pid=1792 comm="/usr/lib/tracker/tracker-miner-fs" "label="unconfined")
Apr 1 12:56:38 mahan-VirtualBox systemd[1340]: Starting Tracker metadata extractor...
Apr 1 12:56:38 mahan-VirtualBox dbus-daemon[1373]: [session uid=1000 pid=1373] Successfully activated service 'org.freedesktop.Tracker1.Miner.Extract'
Apr 1 12:56:38 mahan-VirtualBox systemd[1340]: Started Tracker metadata extractor.
Apr 1 12:56:48 mahan-VirtualBox systemd[1340]: tracker-extract.service: Succeeded.
Apr 1 12:57:04 mahan-VirtualBox dbus-daemon[1373]: [session uid=1000 pid=1373] Activating via systemd: service name='org.freedesktop.Tracker1.Miner.Extract' unit='tracker-extract.service' requested by '':1.54' (uid=1000 pid=1792 comm="/usr/lib/tracker/tracker-miner-fs" "label="unconfined")
Apr 1 12:57:04 mahan-VirtualBox systemd[1340]: Starting Tracker metadata extractor...
Apr 1 12:57:04 mahan-VirtualBox dbus-daemon[1373]: [session uid=1000 pid=1373] Successfully activated service 'org.freedesktop.Tracker1.Miner.Extract'
Apr 1 12:57:04 mahan-VirtualBox systemd[1340]: Started Tracker metadata extractor.
Apr 1 12:57:14 mahan-VirtualBox systemd[1340]: tracker-extract.service: Succeeded.
Apr 1 12:57:39 mahan-VirtualBox kernel: [52652.410293] Loading Module
Apr 1 12:57:39 mahan-VirtualBox kernel: [52652.410295] DATA: 2/ 8/1995
Apr 1 12:57:39 mahan-VirtualBox kernel: [52652.410296] DATA: 21/ 6/1998
Apr 1 12:57:39 mahan-VirtualBox kernel: [52652.410297] DATA: 12/ 4/1999
Apr 1 12:57:39 mahan-VirtualBox kernel: [52652.410298] DATA: 11/ 3/2000
Apr 1 12:57:39 mahan-VirtualBox kernel: [52652.410298] DATA: 9/ 4/2001
Apr 1 13:01:03 mahan-VirtualBox /usr/lib/gdm/gdm-x-session[1358]: (EE) No surface to present from.
Apr 1 13:01:05 mahan-VirtualBox gsd-color[1698]: unable to get EDID for xrandr-Virtual1: unable to get EDID for output
Apr 1 13:01:06 mahan-VirtualBox gsd-color[1698]: message repeated 3 times: [ unable to get EDID for xrandr-Virtual1: unable to get EDID for output]
Apr 1 13:01:06 mahan-VirtualBox org.gnome.Shell.desktop[1575]: Window manager warning: Overwriting existing binding of keysym 31 with keysym 31 (keycode a).
Apr 1 13:01:06 mahan-VirtualBox org.gnome.Shell.desktop[1575]: Window manager warning: Overwriting existing binding of keysym 32 with keysym 32 (keycode b).
Apr 1 13:01:06 mahan-VirtualBox org.gnome.Shell.desktop[1575]: Window manager warning: Overwriting existing binding of keysym 33 with keysym 33 (keycode c).
Apr 1 13:01:06 mahan-VirtualBox org.gnome.Shell.desktop[1575]: Window manager warning: Overwriting existing binding of keysym 34 with keysym 34 (keycode d).
Apr 1 13:01:06 mahan-VirtualBox org.gnome.Shell.desktop[1575]: Window manager warning: Overwriting existing binding of keysym 35 with keysym 35 (keycode e).
Apr 1 13:01:06 mahan-VirtualBox org.gnome.Shell.desktop[1575]: Window manager warning: Overwriting existing binding of keysym 38 with keysym 38 (keycode 11).
Apr 1 13:01:06 mahan-VirtualBox org.gnome.Shell.desktop[1575]: Window manager warning: Overwriting existing binding of keysym 39 with keysym 39 (keycode 12).
Apr 1 13:01:06 mahan-VirtualBox org.gnome.Shell.desktop[1575]: Window manager warning: Overwriting existing binding of keysym 36 with keysym 36 (keycode f).
Apr 1 13:01:06 mahan-VirtualBox org.gnome.Shell.desktop[1575]: Window manager warning: Overwriting existing binding of keysym 37 with keysym 37 (keycode 10).
Apr 1 13:05:00 mahan-VirtualBox kernel: [53092.966736] Removing Module
```

```
mahan@mahan-VirtualBox: ~/Desktop
```

```
[43808.301572] DATA: 21/ 6/1998  
[43808.301573] DATA: 12/ 4/1999  
[43808.301573] DATA: 11/ 3/2000  
[43808.301574] DATA: 9/ 4/2001  
[44849.094895] Removing Module  
[44936.980446] Loading Module  
[44936.980448] DATA: 2/ 8/1995  
[44936.980449] DATA: 21/ 6/1998  
[44936.980449] DATA: 12/ 4/1999  
[44936.980450] DATA: 11/ 3/2000  
[44936.980450] DATA: 9/ 4/2001  
[44965.047327] Removing Module  
[45131.695885] Loading Module  
[45131.695887] DATA: 2/ 8/1995  
[45131.695887] DATA: 21/ 6/1998  
[45131.695888] DATA: 12/ 4/1999  
[45131.695888] DATA: 11/ 3/2000  
[45131.695889] DATA: 9/ 4/2001  
[48085.523397] Removing Module  
[49152.329805] Loading Module  
[49152.329807] DATA: 2/ 8/1995  
[49152.329808] DATA: 21/ 6/1998  
[49152.329808] DATA: 12/ 4/1999  
[49152.329809] DATA: 11/ 3/2000  
[49152.329809] DATA: 9/ 4/2001  
[49180.758605] Removing Module  
[50143.005266] Loading Module  
[50143.005268] DATA: 2/ 8/1995  
[50143.005269] DATA: 21/ 6/1998  
[50143.005269] DATA: 12/ 4/1999  
[50143.005270] DATA: 11/ 3/2000  
[50143.005270] DATA: 9/ 4/2001  
[50165.226340] DATA: 2/ 8/1995  
[50165.226342] DATA: 21/ 6/1998  
[50165.226343] DATA: 12/ 4/1999  
[50165.226344] DATA: 11/ 3/2000  
[50165.226345] DATA: 9/ 4/2001  
[50165.226346] Removing Module  
[52652.410293] Loading Module  
[52652.410295] DATA: 2/ 8/1995  
[52652.410296] DATA: 21/ 6/1998  
[52652.410297] DATA: 12/ 4/1999  
[52652.410298] DATA: 11/ 3/2000  
[52652.410298] DATA: 9/ 4/2001  
[53092.906736] Removing Module
```

```
mahan@mahan-VirtualBox: ~/Desktop$
```

همانطور که می بینیم مرحل به طور کامل و درست انجام شده است.

**THE END**