



گزارش آزمایش ۱۰: آشنایی با درایورها

ملیکا علیزاده ۴۰۱۱۰۶۲۵۵

الینا هژبری ۴۰۱۱۷۰۶۶۱

۳.۱۰ آزمایش ۱

در این بخش باید ماژول سطح هسته‌ی hello را بنویسیم. ولی قبل از نوشتن ماژول، باید دستور زیر را اجرا کنیم تا هدر لینوکس با توجه به نسخه‌ی هسته‌ی فعلی آپدیت شود:

```
sudo apt install build-essential linux-headers-$(uname -r)
```

سپس فایل hello.c را به شکل زیر ساخته.

```
۱ #include <linux/module.h>
۲ #include <linux/kernel.h>
۳ #include <linux/init.h>
۴
۵ MODULE_LICENSE("GPL");
۶
۷ static int __init hello_init(void) {
۸     printk(KERN_INFO "Hello\n");
۹     return 0;
۱۰ }
۱۱
۱۲ module_init(hello_init);
۱۳
۱۴
```

بعد باید فایل Makefile آن را بنویسیم.

```
۱ obj-m += hello.o
۲
۳ all:
۴     make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
۵
```

```
6 clean:
7     make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
8
9
10
```

حال با استفاده از دستورات زیر ماژول را لود و اجرا می‌کنیم.
آماده کردن فایل اجرایی یا .ko. ماژول:

make

لودکردن ماژول در هسته:

sudo insmod hello.ko

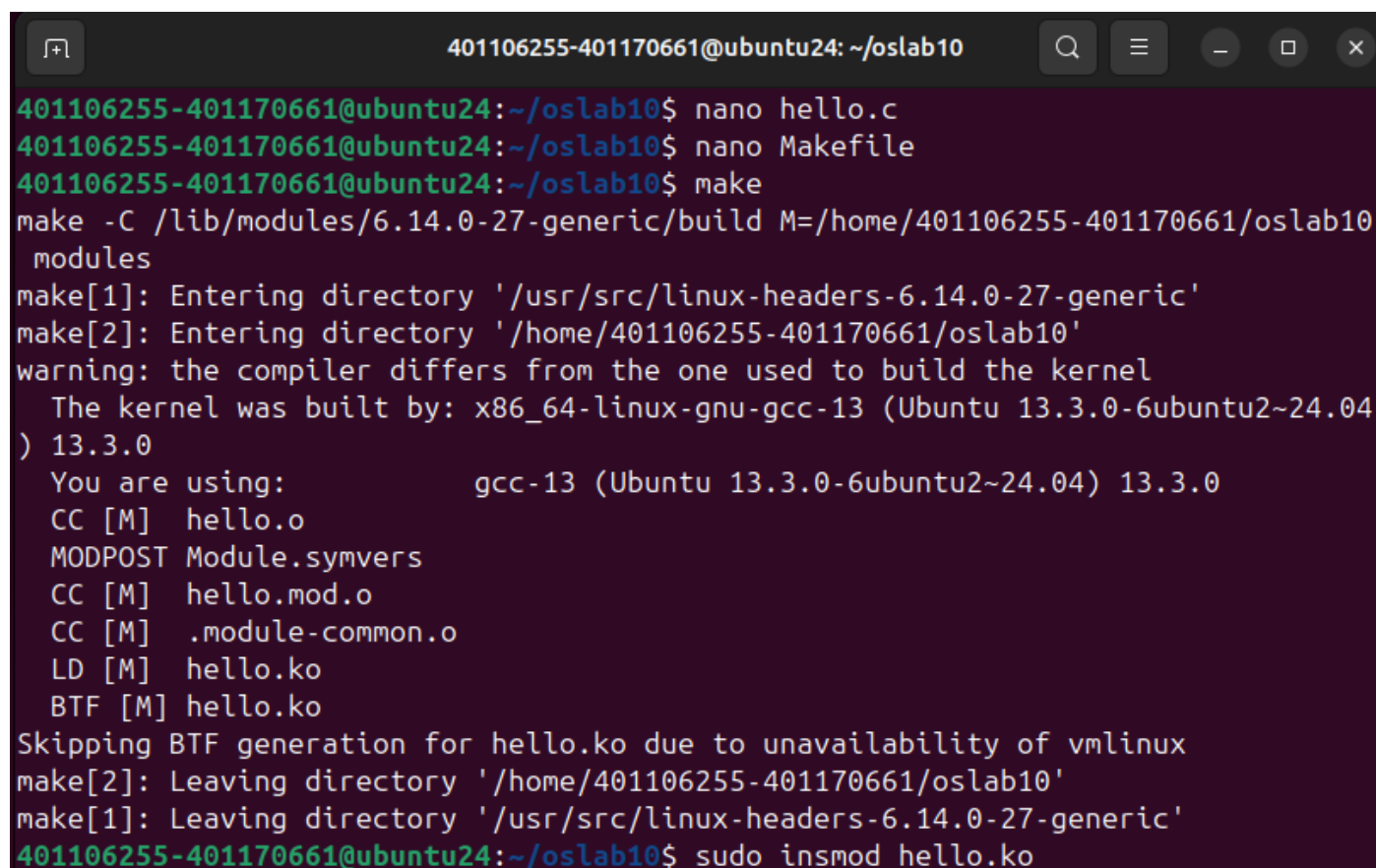
مشاهده‌ی خروجی:

sudo dmesg

و برای آنلود ماژول و پاک کردن فایل اجرایی آن:

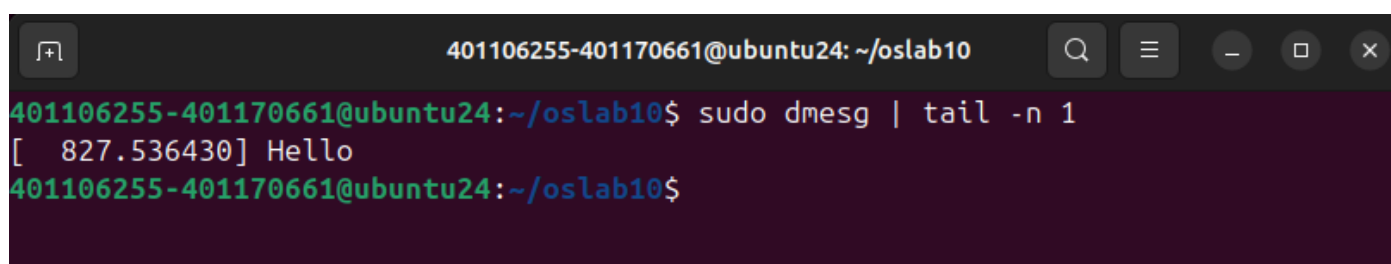
sudo rmmod hello

make clean



```
401106255-401170661@ubuntu24: ~/oslab10
401106255-401170661@ubuntu24:~/oslab10$ nano hello.c
401106255-401170661@ubuntu24:~/oslab10$ nano Makefile
401106255-401170661@ubuntu24:~/oslab10$ make
make -C /lib/modules/6.14.0-27-generic/build M=/home/401106255-401170661/oslab10
modules
make[1]: Entering directory '/usr/src/linux-headers-6.14.0-27-generic'
make[2]: Entering directory '/home/401106255-401170661/oslab10'
warning: the compiler differs from the one used to build the kernel
The kernel was built by: x86_64-linux-gnu-gcc-13 (Ubuntu 13.3.0-6ubuntu2~24.04
) 13.3.0
You are using: gcc-13 (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0
CC [M] hello.o
MODPOST Module.symvers
CC [M] hello.mod.o
CC [M] .module-common.o
LD [M] hello.ko
BTF [M] hello.ko
Skipping BTF generation for hello.ko due to unavailability of vmlinux
make[2]: Leaving directory '/home/401106255-401170661/oslab10'
make[1]: Leaving directory '/usr/src/linux-headers-6.14.0-27-generic'
401106255-401170661@ubuntu24:~/oslab10$ sudo insmod hello.ko
```

شکل ۱: لود ماژول

A terminal window with a dark background. The title bar shows the user and host information: '401106255-401170661@ubuntu24: ~/oslab10'. The terminal contains the following text:

```
401106255-401170661@ubuntu24:~/oslab10$ sudo dmesg | tail -n 1  
[ 827.536430] Hello  
401106255-401170661@ubuntu24:~/oslab10$
```

شکل ۲: نمایش خروجی کد در dmesg

۴.۱۰ آزمایش ۲

در این بخش باید ماژول سطح هسته‌ی nets را بنویسیم. ولی قبل از نوشتن ماژول، باید دستور زیر را اجرا کنیم تا هدر لینوکس با توجه به نسخه‌ی هسته‌ی فعلی آپدیت شود:

```
sudo apt install build-essential linux-headers-$(uname -r)
```

سپس فایل `nets.c` را به شکل زیر ساخته. ابتدا یک ساختار برای تنظیمات Netfilter که سیستم فایروال لینوکس است ساخته شده تا مشخص شود پکت را از کجا و چگونه بگیریم. و در یک تابع داده‌ها توسط کرنل در فایل لاگ به صورت مستقیم نوشته می‌شود. و اگر مشکلی در بازکردن فایل بود در `dmesg` مشخص می‌شود.

```
۱ #include <linux/module.h>
۲ #include <linux/kernel.h>
۳ #include <linux/netfilter.h>
۴ #include <linux/netfilter_ipv4.h>
۵ #include <linux/skbuff.h>
۶ #include <linux/ip.h>
۷ #include <linux/fs.h>
۸ #include <linux/uaccess.h>
۹
۱۰ MODULE_LICENSE("GPL");
۱۱
۱۲ static struct nf_hook_ops netfilter_ops;
۱۳
۱۴ void log_packet(const char *data);
۱۵
۱۶ void log_packet(const char *data) {
۱۷     struct file *file;
۱۸     loff_t pos = 0;
۱۹     file = filp_open("../packet_log.txt", O_WRONLY | O_CREAT | O_APPEND, 0644);
۲۰     if (!IS_ERR(file)) {
۲۱         kernel_write(file, data, strlen(data), &pos);
۲۲         filp_close(file, NULL);
۲۳     } else {
۲۴         printk(KERN_ERR "%ld\n", PTR_ERR(file));
۲۵     }
۲۶ }
۲۷
۲۸
۲۹
```

در ادامه هر بار که پکتی وارد سیستم میشه تابع زیر صدا زده میشه و با توجه به هدر رشته‌ی آن را با `log_packet` در فایل می‌نویسد. سپس در تابع `init` ماژول موارد زیر تعریف شده‌است:

- `hook = packet_handler`: تابع را به عنوان هندلر ثبت می‌کند `pf = PF_INET`: یعنی فقط IPv4 را در نظر بگیرد.
- `hooknum = NF_INET_PRE_ROUTING`: یعنی پکت را در ابتدای مسیر بگیرد.
- `priority = NF_IP_PRI_FIRST`: یعنی بالاترین اولویت را داشته باشد.
- `nf_register_net_hook`: برای ثبت کردن hook در Netfilter است.

```

1 static unsigned int packet_handler(void *priv, struct sk_buff *skb, const struct nf_hook_state *state) {
2     struct iphdr *ip_header;
3     char log_entry[256];
4     if (!skb)
5         return NF_ACCEPT;
6     ip_header = ip_hdr(skb);
7     if (ip_header) {
8         snprintf(log_entry, sizeof(log_entry), "SRC: %pI4, DST: %pI4\n", &ip_header->saddr, &ip_header->da
9         log_packet(log_entry);
10    }
11    return NF_ACCEPT;
12 }
13
14 static int __init nets_init(void) {
15     netfilter_ops.hook = packet_handler;
16     netfilter_ops.pf = PF_INET;
17     netfilter_ops.hooknum = NF_INET_PRE_ROUTING;
18     netfilter_ops.priority = NF_IP_PRI_FIRST;
19     nf_register_net_hook(&init_net, &netfilter_ops);
20     printk(KERN_INFO "driver init\n");
21     return 0;
22 }
23
24 module_init(nets_init);
25
26
27

```

بعد باید فایل Makefile آن را بنویسیم.

```

1 obj-m += nets.o
2
3 all:
4     make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
5
6 clean:
7     make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
8
9
10

```

حال با استفاده از دستورات زیر ماژول را لود و اجرا می‌کنیم.
آماده کردن فایل اجرایی یا .ko. ماژول:

make

لودکردن ماژول در هسته:

sudo insmod nets.ko

مشاهده‌ی خروجی:

```
sudo dmesg
```

و برای آنلود ماژول و پاک کردن فایل اجرایی آن:

```
sudo rmmmod nets
```

```
make clean
```

```
401106255-401170661@ubuntu24:~/oslab10/2$ nano nets.c
401106255-401170661@ubuntu24:~/oslab10/2$ nano Makefile
401106255-401170661@ubuntu24:~/oslab10/2$ make
make -C /lib/modules/6.14.0-27-generic/build M=/home/401106255-401170661/oslab10/2 modules
make[1]: Entering directory '/usr/src/linux-headers-6.14.0-27-generic'
make[2]: Entering directory '/home/401106255-401170661/oslab10/2'
warning: the compiler differs from the one used to build the kernel
The kernel was built by: x86_64-linux-gnu-gcc-13 (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0
You are using: gcc-13 (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0
CC [M] nets.o
MODPOST Module.symvers
CC [M] nets.mod.o
CC [M] .module-common.o
LD [M] nets.ko
BTF [M] nets.ko
Skipping BTF generation for nets.ko due to unavailability of vmlinux
make[2]: Leaving directory '/home/401106255-401170661/oslab10/2'
make[1]: Leaving directory '/usr/src/linux-headers-6.14.0-27-generic'
401106255-401170661@ubuntu24:~/oslab10/2$ sudo dmesg -C
[sudo] password for 401106255-401170661:
401106255-401170661@ubuntu24:~/oslab10/2$ sudo insmod nets.ko
```

شکل ۳: لود ماژول

```
[ 2306.710644] audit: type=1400 audit(1755966976.868:215): apparmor="DENIED" operation="mknod" class="file" profile="snap.firefox.firefox" name="/var/log/packet_log.txt" pid=2802 comm=536F636B657420546872656164 requested_mask="c" denied_mask="c" fsuid=1000 ouid=1000
[ 2306.710664] audit: type=1400 audit(1755966976.868:216): apparmor="DENIED" operation="mknod" class="file" profile="snap.firefox.firefox" name="/var/log/packet_log.txt" pid=2802 comm=536F636B657420546872656164 requested_mask="c" denied_mask="c" fsuid=1000 ouid=1000
[ 2306.712097] -13
[ 2306.712291] audit: type=1400 audit(1755966976.869:217): apparmor="DENIED" operation="mknod" class="file" profile="snap.firefox.firefox" name="/var/log/packet_log.txt" pid=2802 comm=536F636B657420546872656164 requested_mask="c" denied_mask="c" fsuid=1000 ouid=1000
[ 2306.714385] -13
[ 2306.714492] audit: type=1400 audit(1755966976.872:218): apparmor="DENIED" operation="mknod" class="file" profile="snap.firefox.firefox" name="/var/log/packet_log.txt" pid=2802 comm=536F636B657420546872656164 requested_mask="c" denied_mask="c" fsuid=1000 ouid=1000
[ 2306.714639] -13
[ 2306.714678] audit: type=1400 audit(1755966976.872:219): apparmor="DENIED" operation="mknod" class="file" profile="snap.firefox.firefox" name="/var/log/packet_log.txt" pid=2802 comm=536F636B657420546872656164 requested_mask="c" denied_mask="c" fsuid=1000 ouid=1000
```

شکل ۴: نمایش خروجی کد در dmesg

```
401106255-401170661@ubuntu24:~/oslab10/2$ sudo cat packet_log.txt
SRC: 127.0.0.1, DST: 127.0.0.53
SRC: 127.0.0.1, DST: 127.0.0.53
SRC: 188.114.99.0, DST: 10.0.2.15
SRC: 185.125.190.17, DST: 10.0.2.15
SRC: 127.0.0.1, DST: 127.0.0.53
SRC: 127.0.0.1, DST: 127.0.0.53
SRC: 34.36.137.203, DST: 10.0.2.15
SRC: 34.36.137.203, DST: 10.0.2.15
SRC: 104.18.39.21, DST: 10.0.2.15
SRC: 104.18.39.21, DST: 10.0.2.15
SRC: 104.18.39.21, DST: 10.0.2.15
SRC: 104.18.39.21, DST: 10.0.2.15
SRC: 104.18.39.21, DST: 10.0.2.15
SRC: 127.0.0.1, DST: 127.0.0.53
SRC: 127.0.0.1, DST: 127.0.0.53
401106255-401170661@ubuntu24:~/oslab10/2$
```

شکل ۵: نمایش خروجی کد در فایل لاگ packet_log.txt