

بسمه تعالی



دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

# آز سیستم‌های عامل

## گزارش آزمایش هشتم بخش اول

نگارش

**امین منصوری**

استاد

**بیگی**

برای نصب ابزارهای توسعه مورد نیاز دستور زیر را وارد می‌کنیم:

```
root@debian:~# apt install linux-headers-$(uname -r)
Reading package lists... Done
Building dependency tree
Reading state information... Done
linux-headers-3.16.0-6-686-pae is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
root@debian:~#
```

قبل از ایجاد ماژول، لازم است که آدرس شروع جدول فراخوانی‌های سیستمی را در حافظه به دست آوریم. به همین منظور، دستور زیر را در ترمینال وارد می‌کنیم:

```
root@debian:~# grep 'sys_call_table' < /boot/System.map-$(uname -r)
c14b5300 R sys_call_table
root@debian:~#
```

در ادامه، کد ماژول مورد نظر را ایجاد می‌کنیم. این کد در تصویر زیر قابل مشاهده است:

```
GNU nano 2.2.6 File: 1.c

#include <linux/init.h>
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/syscalls.h>
#include <uapi/asm-generic/unistd.h>

MODULE_LICENSE("GPL");
MODULE_AUTHOR("966127035");
MODULE_DESCRIPTION("OSLab-Exp8-Part1");
MODULE_VERSION("0.01");

unsigned long *syscall_table_addr = (unsigned long *)0xffffffffc14b5300;

static int __init init_print_syscalls(void) {
    printk(KERN_INFO "INIT: Print Syscalls\n");
    int i;
    for (i = 0; i < __NR_syscalls; i++)
        printk(KERN_INFO "Syscall %03d: %lx\n", i, (unsigned long)(syscall_table_addr + i));
    return 0;
}

static void __exit exit_print_syscalls(void) {
    printk(KERN_INFO "Exit: Print Syscalls\n");
}

module_init(init_print_syscalls);
module_exit(exit_print_syscalls);
```

در تصویر بالا، بعد از وارد کردن فایل‌های سرآیند، اطلاعات مربوط به ماژول مانند لایسنس، نام مولف، توضیحات را وارد می‌کنیم.

در ادامه دو تابع `init_print_syscalls` و `exit_print_syscalls` را برای ماژول ایجاد می‌کنیم.

در تابع `init`، آدرس هر کدام از فراخوانی‌های سیستمی را چاپ می‌کنیم. برای به دست آوردن آدرس فراخوانی سیستمی `i`، کافی است آدرس ابتدای جدول فراخوانی‌های سیستمی را به اندازه `i` واحد افزایش دهیم. همچنین، توجه کنید که محتوای تابع `exit` تزئینی است و کارکرد خاصی ندارد.

در نهایت، این دو تابع را در ورودی `module_init` و `module_exit` قرار می‌دهیم تا ماژول کامل شود.

بعد از ایجاد کد ماژول، نوبت به ایجاد کردن آن با دستور `make` می‌رسد. به همین منظور، ابتدا فایل `Makefile` را ایجاد می‌کنیم. تصویر کد مربوط به این فایل در زیر آمده است:

GNU nano 2.2.6	File: Makefile	Modified
<pre>obj-m += 1.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  test:     dmesg -C     insmod 1.ko     rmmod 1.ko     dmesg</pre>		

حال، دستور `make` را در ترمینال وارد می‌کنیم:

```
In file included from /root/oslab/8/1/1.c:5:0:
/usr/src/linux-headers-3.16.0-6-common/include/uapi/asm-generic/unistd.h:910:0: warning: "__NR_fadvise64_64" redefined
#define __NR_fadvise64_64 __NR3264_fadvise64
^
In file included from /usr/src/linux-headers-3.16.0-6-common/arch/x86/include/asm/unistd.h:15:0,
                 from /usr/src/linux-headers-3.16.0-6-common/include/uapi/linux/unistd.h:7,
                 from /usr/src/linux-headers-3.16.0-6-common/include/linux/syscalls.h:77,
                 from /root/oslab/8/1/1.c:4:
arch/x86/include/generated/uapi/asm/unistd_32.h:273:0: note: this is the location of the previous definition
#define __NR_fadvise64_64 272
^
/root/oslab/8/1/1.c: In function 'init_print_syscalls':
/root/oslab/8/1/1.c:16:5: warning: ISO C90 forbids mixed declarations and code [-Wdeclaration-after-statement]
    int i;
    ^
Building modules, stage 2.
MODPOST 1 modules
CC      /root/oslab/8/1/1.mod.o
LD [M]  /root/oslab/8/1/1.ko
make[1]: Leaving directory '/usr/src/linux-headers-3.16.0-6-686-pae'
root@debian:~/oslab/8/1# _
```

برای تست کردن ماثول ایجاد شده کافی است دستور make test را در ترمینال وارد کنیم. با وارد کردن این دستور، آدرس فراخوانی‌های سیستمی چاپ خواهد شد:

```
[ 2466.234299] Syscall 245: c14b56d4
[ 2466.234301] Syscall 246: c14b56d8
[ 2466.234302] Syscall 247: c14b56dc
[ 2466.234304] Syscall 248: c14b56e0
[ 2466.234306] Syscall 249: c14b56e4
[ 2466.234307] Syscall 250: c14b56e8
[ 2466.234309] Syscall 251: c14b56ec
[ 2466.234310] Syscall 252: c14b56f0
[ 2466.234312] Syscall 253: c14b56f4
[ 2466.234314] Syscall 254: c14b56f8
[ 2466.234315] Syscall 255: c14b56fc
[ 2466.234317] Syscall 256: c14b5700
[ 2466.234319] Syscall 257: c14b5704
[ 2466.234320] Syscall 258: c14b5708
[ 2466.234322] Syscall 259: c14b570c
[ 2466.234323] Syscall 260: c14b5710
[ 2466.234325] Syscall 261: c14b5714
[ 2466.234327] Syscall 262: c14b5718
[ 2466.234328] Syscall 263: c14b571c
[ 2466.234330] Syscall 264: c14b5720
[ 2466.234331] Syscall 265: c14b5724
[ 2466.234333] Syscall 266: c14b5728
[ 2466.234335] Syscall 267: c14b572c
[ 2466.234336] Syscall 268: c14b5730
[ 2466.234338] Syscall 269: c14b5734
[ 2466.234340] Syscall 270: c14b5738
[ 2466.234341] Syscall 271: c14b573c
[ 2466.234343] Syscall 272: c14b5740
[ 2466.234344] Syscall 273: c14b5744
[ 2466.234346] Syscall 274: c14b5748
[ 2466.234348] Syscall 275: c14b574c
[ 2466.234349] Syscall 276: c14b5750
[ 2466.234351] Syscall 277: c14b5754
[ 2466.234353] Syscall 278: c14b5758
[ 2466.234354] Syscall 279: c14b575c
[ 2466.236496] Exit: Print Syscalls
root@debian:~/oslab/8/1# _
```