

Activity 12: Kernel Module

สมาชิก

ชื่อ-นามสกุล	เลขประจำตัวนิสิต
นายสีปภาส ชวานนท์	6630333721
นายเนติภัทร โพธิพันธ์	6631331621
นายวรลภย์ ศรีชัยนนท์	6632200221

Checkpoint #1

Create a dummy kernel module (dummy.c) and use the given Makefile to build the module. Use insmod, rmmod to install/remove module. You have to demonstrate that the module has been installed and removed (using dmesg and lsmod).

ในขั้นตอนนี้ จะใช้ไฟล์ **dummy.c** ดังกล่าว

```
// dummy.c
#include <linux/module.h> /* Needed by all modules */
#include <linux/kernel.h> /* Needed for KERN_INFO */

MODULE_LICENSE("GPL");
MODULE_AUTHOR("KRERK PIROMSOPA, PH.D. <Krerk.P@chula.ac.th>");
MODULE_DESCRIPTION("\cpmod\ Dummy Kernel Module");

int init_module(void)
{
    printk(KERN_INFO "CPMOD: init\n");

    // non 0 - means init_module failed / module can't be loaded
    return 0;
}

void cleanup_module(void)
{
    printk(KERN_INFO "CPMOD: cleanup\n");
}
```

และ **Makefile** ดังกล่าว

```
obj-m += dummy.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
```

จากนั้นเปิด Ubuntu Virtual Box และใช้งานคำสั่ง **make** เพื่อให้ Makefile ทำงาน และ build ไฟล์ที่จำเป็นต่อการใช้งาน

```
reisenx@reisenx-VirtualBox: ~/Documents/act12/dummy$ make
make -C /lib/modules/6.8.0-51-generic/build M=/home/reisenx/Documents/act12/dummy modules
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-51-generic'
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] /home/reisenx/Documents/act12/dummy/dummy.o
MODPOST /home/reisenx/Documents/act12/dummy/Module.symvers
CC [M] /home/reisenx/Documents/act12/dummy/dummy.mod.o
LD [M] /home/reisenx/Documents/act12/dummy/dummy.ko
BTF [M] /home/reisenx/Documents/act12/dummy/dummy.ko
Skipping BTF generation for /home/reisenx/Documents/act12/dummy/dummy.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-51-generic'
reisenx@reisenx-VirtualBox: ~/Documents/act12/dummy$
```

- จากนั้นใช้งานคำสั่ง **sudo insmod dummy.ko** เพื่อ load kernel module ของ dummy
- เพื่อให้มั่นใจว่า kernel dummy ของเราจะถูก load เข้าไปในระบบปฏิบัติการจริงๆ ให้ตรวจสอบโดยใช้คำสั่ง **lsmod | grep dummy** ซึ่งจะเห็นว่าในที่นี้เรามี kernel module dummy ที่ถูก load เข้าไปในระบบปฏิบัติการจริงๆ
- นอกจากนี้เรายังสามารถตรวจสอบ log messages ที่มาจาก kernel โดยการใช้คำสั่ง **sudo dmesg | grep CPMOD** เพื่อเช็คว่ามี kernel module dummy ถูก load เข้าไปในระบบปฏิบัติการ

```
reisenx@reisenx-VirtualBox: ~/Documents/act12/dummy$ sudo insmod dummy.ko
reisenx@reisenx-VirtualBox: ~/Documents/act12/dummy$ lsmod | grep dummy
dummy                12288  0
snd_seq_dummy        12288  0
snd_seq              118784  9 snd_seq_midi,snd_seq_midi_event,snd_seq_dummy
reisenx@reisenx-VirtualBox: ~/Documents/act12/dummy$ sudo dmesg | grep CPMOD
[ 4733.942572] CPMOD: init
```

- จากนั้นใช้งานคำสั่ง **sudo rmmod dummy** เพื่อนำ kernel module ของ dummy ออกไปจากระบบปฏิบัติการ (unload)
- เพื่อตรวจสอบว่า kernel module dummy ถูกเอาออกไปจริงๆ เราสามารถตรวจสอบ log messages โดยใช้คำสั่ง **sudo dmesg | grep CPMOD**

```
reisenx@reisenx-VirtualBox: ~/Documents/act12/dummy$ sudo rmmod dummy
reisenx@reisenx-VirtualBox: ~/Documents/act12/dummy$ lsmod | grep dummy
snd_seq_dummy        12288  0
snd_seq              118784  9 snd_seq_midi,snd_seq_midi_event,snd_seq_dummy
reisenx@reisenx-VirtualBox: ~/Documents/act12/dummy$ sudo dmesg | grep CPMOD
[ 4733.942572] CPMOD: init
[ 5375.637330] CPMOD: cleanup
```

สุดท้ายให้ใช้งานคำสั่ง **make clean** เพื่อนำไฟล์ที่ไม่จำเป็นออกไปจากเครื่อง (clean)

```
reisenx@reisenx-VirtualBox: ~/Documents/act12/dummy$ make clean
make -C /lib/modules/6.8.0-51-generic/build M=/home/reisenx/Documents/act12/dummy clean
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-51-generic'
CLEAN /home/reisenx/Documents/act12/dummy/Module.symvers
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-51-generic'
reisenx@reisenx-VirtualBox: ~/Documents/act12/dummy$
```

Checkpoint #2

In this exercise, you have to compile the given character driver

Use insmod to install the driver. Once the driver is installed, use the mknod command to create /dev/osinfo device file. You may obtain the major device number from dmesg. Please also show the content of /dev/osinfo.

ในขั้นตอนนี้ จะใช้ไฟล์ **osinfo.c** ดังกล่าว

```
// osinfo.c
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/fs.h>
#include <asm/uaccess.h>
/* Needed by all modules */
/* Needed for KERN_INFO */

MODULE_LICENSE("GPL");
MODULE_AUTHOR("KRERK PIROMSOPA, PH.D. <Krerk.P@chula.ac.th>");
MODULE_DESCRIPTION("\\"osinfo\\" Character Device");

#define DEVICENAME "osinfo"

static int dev_major;
static int dev_open = 0;
static char *f_ptr;

static const char f_data0[] =
    "0:CP ENG CU OS 2022S2 - Instructors\n"
    "1:\tVeera Muangsin, Ph.D.\n"
    "2:\tKrerk Piromsopa, Ph.D.\n"
    "3:\tThongchai Rojkangsadan\n";

// prototypes for device functions
static int device_open(struct inode *, struct file *);
static int device_release(struct inode *inode, struct file *file);
static ssize_t device_read(struct file *, char *, size_t, loff_t *);

// File operations structor
// Only implement those that will be used.
static struct file_operations dev_fops = {
    .read    = device_read,
    .open    = device_open,
    .release = device_release
};
```

```

int init_module(void)
{
    printk(KERN_INFO "CPCHAR: dev osinfo init\n");
    dev_major = register_chrdev(0, DEVICENAME, &dev_fops);

    if (dev_major < 0) {
        printk(KERN_ALERT "Fail register_chrdev osinfo with %d\n", dev_major);
        return dev_major;
    }

    printk(KERN_INFO "Device MajorNumber %d.\n", dev_major);
    printk(KERN_INFO "To create a device file:\n");
    printk(KERN_INFO "\t'mknod /dev/%s c %d 0'.\n", DEVICENAME, dev_major);
    printk(KERN_INFO "Try varying minor numbers.\n");
    printk(KERN_INFO "Please remove the device file and module when done.\n");

    /* * non 0 - means init_module failed */
    return 0;
}

void cleanup_module(void)
{
    printk(KERN_INFO "CPCHAR: dev osinfo cleanup\n");
    unregister_chrdev(dev_major, DEVICENAME);
}

static int device_open(struct inode *inode, struct file *file)
{
    if (dev_open)
        return -EBUSY;

    dev_open++;
    printk(KERN_INFO "dev minor %d\n", MINOR(inode->i_rdev));
    f_ptr = (char *)f_data0;

    // lock module
    try_module_get(THIS_MODULE);

    return 0;
}

static int device_release(struct inode *inode, struct file *file)
{
    dev_open--; /* We're now ready for our next caller */
    // release module
    module_put(THIS_MODULE);
    return 0;
}

```

```

static ssize_t device_read(struct file *filp,
                           char *buffer,
                           /* see include/linux/fs.h */
                           /* buffer to fill with data */
                           /* length of the buffer */
                           size_t length,
                           loff_t *offset)
{
    int bytes_read = 0;

    if (*f_ptr == 0) {
        return 0;
    }

    while (length && *f_ptr) {
        put_user(*(f_ptr++), buffer++);
        length--;
        bytes_read++;
    }

    return bytes_read;
}

```

และสร้าง **Makefile** โดยมีลักษณะดังนี้

```

obj-m += osinfo.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

```

ใช้งานคำสั่ง **make** เพื่อให้ Makefile ทำงาน และ build ไฟล์ที่จำเป็นต่อการใช้งาน

```
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$ make
make -C /lib/modules/6.8.0-51-generic/build M=/home/reisenx/Documents/act12/osinfo modules
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-51-generic'
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] /home/reisenx/Documents/act12/osinfo/osinfo.o
MODPOST /home/reisenx/Documents/act12/osinfo/Module.symvers
CC [M] /home/reisenx/Documents/act12/osinfo/osinfo.mod.o
LD [M] /home/reisenx/Documents/act12/osinfo/osinfo.ko
BTF [M] /home/reisenx/Documents/act12/osinfo/osinfo.ko
Skipping BTF generation for /home/reisenx/Documents/act12/osinfo/osinfo.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-51-generic'
```

- ใช้งานคำสั่ง **sudo insmod osinfo.ko** เพื่อ load kernel module ของ osinfo
- จากนั้นตรวจสอบ MajorNumber โดยการใช้งานคำสั่ง **sudo dmesg | grep "Device MajorNumber"**

```
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$ sudo insmod osinfo.ko
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$ sudo dmesg | grep "Device MajorNumber"
[ 7535.831326] Device MajorNumber 240.
```

- ใช้งานคำสั่ง **sudo mknod /dev/osinfo c 240 0** เพื่อสร้าง Character Device ชื่อว่า osinfo ในโฟลเดอร์ dev
- แสดงผลไฟล์ Character Device ที่ถูกสร้างขึ้นมาด้วยคำสั่ง **cat /dev/osinfo**

```
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$ sudo mknod /dev/osinfo c 240 0
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$ cat /dev/osinfo
0:CP ENG CU OS 2022S2 - Instructors
1:      Veera Muangsinn, Ph.D.
2:      Krerk Piromsopa, Ph.D.
3:      Thongchai Rojksangsan
```

- ใช้งานคำสั่ง **sudo rmmod osinfo** เพื่อนำ kernel module ของ osinfo ออกไปจากระบบปฏิบัติการ (unload)
- ลบโฟลเดอร์และไฟล์ที่ถูกสร้างขึ้นมา (/dev/osinfo) ด้วยคำสั่ง **sudo rm /dev/osinfo**
- สุดท้ายให้ใช้งานคำสั่ง **make clean** เพื่อนำไฟล์ที่ไม่จำเป็นออกไปจากเครื่อง (clean)

```
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$ sudo rmmod osinfo
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$ sudo rm /dev/osinfo
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$ make clean
make -C /lib/modules/6.8.0-51-generic/build M=/home/reisenx/Documents/act12/osinfo clean
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-51-generic'
CLEAN /home/reisenx/Documents/act12/osinfo/Module.symvers
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-51-generic'
```

Checkpoint #3

Modify the osinfo driver to display your name (student name) when the minor device number is 1. (Note that the system must display the instructor information when the minor device number is 0.)

Create a new device file (/dev/osinfo1). Show the expected result to your instructor.

The content of /dev/osinfo1 must display your group information.

เนื่องจากว่าใน checkpoint นี้ จะต้องแสดงชื่อของสมาชิกภายในกลุ่ม จึงต้องมีการแก้ไข **osinfo.c** ดังนี้ จากเดิมที่ส่วนของ code เป็นแบบนี้

```
static const char f_data0[] =
    "0:CP ENG CU OS 2022S2 - Instructors\n"
    "1:\tVeera Muangsin, Ph.D.\n"
    "2:\tKrerk Piromsopa, Ph.D.\n"
    "3:\tThongchai Rojkangsadan\n";
```

จะต้องเติมชื่อสมาชิกในกลุ่มเพิ่มเป็นดังนี้

```
static const char f_data0[] =
    "0:CP ENG CU OS 2022S2 - Instructors\n"
    "1:\tVeera Muangsin, Ph.D.\n"
    "2:\tKrerk Piromsopa, Ph.D.\n"
    "3:\tThongchai Rojkangsadan\n";

/* ----- ADD YOUR GROUP MEMBERS HERE ----- */
static const char f_data1[] =
    "0:CP ENG CU OS 2022S2 - Students, Group Name: LigmaBoy\n"
    "1:\t6630333721 Sippapas Chavanont\n"
    "2:\t6631331621 Naytipat Phothipan\n"
    "3:\t6632200221 Worralop Srichainont\n";
/* ----- */
```

และจากเดิมที่ส่วนของ code เป็นแบบนี้

```
static int device_open(struct inode *inode, struct file *file)
{
    if (dev_open)
        return -EBUSY;
    dev_open++;
    printk(KERN_INFO "dev minor %d\n", MINOR(inode->i_rdev));
    f_ptr = (char *)f_data0;
    // lock module
    try_module_get(THIS_MODULE);
    return 0;
}
```

จะต้องแก้ไขส่วนนี้ให้เป็นแบบนี้ จึงจะสามารถแสดงชื่อของสมาชิกภายในกลุ่มได้

```

static int device_open(struct inode *inode, struct file *file)
{
    if (dev_open)
        return -EBUSY;
    dev_open++;
    /* ----- MODIFY THIS ----- */
    int minor = MINOR(inode->i_rdev);
    printk(KERN_INFO "dev minor %d\n", minor);
    if(minor == 0) {
        f_ptr = (char *)f_data0;
    }
    else if(minor == 1) {
        f_ptr = (char *)f_data1;
    }
    else {
        return -ENODEV;
    }
    /* ----- */
    // lock module
    try_module_get(THIS_MODULE);
    return 0;
}

```

ส่วน **Makefile** ยังคงเหมือนเดิม ไม่ต้องแก้ไขอะไร

หลังจากที่แก้ไฟล์เสร็จสิ้น เราจะต้องทำการ build ใหม่ โดยใช้คำสั่ง **make**

```
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$ make
make -C /lib/modules/6.8.0-51-generic/build M=/home/reisenx/Documents/act12/osinfo modules
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-51-generic'
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] /home/reisenx/Documents/act12/osinfo/osinfo.o
MODPOST /home/reisenx/Documents/act12/osinfo/Module.symvers
CC [M] /home/reisenx/Documents/act12/osinfo/osinfo.mod.o
LD [M] /home/reisenx/Documents/act12/osinfo/osinfo.ko
BTF [M] /home/reisenx/Documents/act12/osinfo/osinfo.ko
Skipping BTF generation for /home/reisenx/Documents/act12/osinfo/osinfo.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-51-generic'
```

- ใช้งานคำสั่ง **sudo insmod osinfo.ko** เพื่อ load kernel module ของ osinfo
- จากนั้นตรวจสอบ MajorNumber โดยการใช้คำสั่ง **sudo dmesg | grep "Device MajorNumber"**

```
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$ sudo insmod osinfo.ko
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$ sudo dmesg | grep "Device MajorNumber"
[ 7535.831326] Device MajorNumber 240.
[ 7650.358495] Device MajorNumber 240.
```

- ใช้งานคำสั่ง **sudo mknod /dev/osinfo c 240 0** เพื่อสร้าง Character Device โดยมี minor device number เท่ากับ 0 ชื่อว่า osinfo ในโฟลเดอร์ dev
- ใช้งานคำสั่ง **sudo mknod /dev/osinfo1 c 240 1** เพื่อสร้าง Character Device โดยมี minor device number เท่ากับ 1 ชื่อว่า osinfo1 ในโฟลเดอร์ dev
- แสดงผลไฟล์ Character Device ที่ถูกสร้างขึ้นมาด้วยคำสั่ง **cat /dev/osinfo** ซึ่งแสดงผลเป็นรายชื่ออาจารย์เนื่องจากมี minor device number เท่ากับ 0
- แสดงผลไฟล์ Character Device ที่ถูกสร้างขึ้นมาด้วยคำสั่ง **cat /dev/osinfo1** ซึ่งแสดงผลเป็นรายชื่อนิสิตภายในกลุ่มเนื่องจากมี minor device number เท่ากับ 1

```
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$ sudo mknod /dev/osinfo c 240 0
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$ sudo mknod /dev/osinfo1 c 240 1
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$ cat /dev/osinfo
0:CP ENG CU OS 2022S2 - Instructors
1: Veera Muangsin, Ph.D.
2: Krerk Piromsopa, Ph.D.
3: Thongchai Rojkangsadan
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$ cat /dev/osinfo1
0:CP ENG CU OS 2022S2 - Students, Group Name: LigmaBoy
1: 6630333721 Sippapas Chavanont
2: 6631331621 Naytipat Phothipan
3: 6632200221 Worralop Srichainont
```

- ใช้งานคำสั่ง `sudo rmmod osinfo` เพื่อนำ kernel module ของ osinfo ออกไปจากระบบปฏิบัติการ (unload)
- ลบโพลเดอร์และไฟล์ที่ถูกสร้างขึ้นมา (/dev/osinfo และ /dev/osinfo1) ด้วยคำสั่ง `sudo rm /dev/osinfo` และ `sudo rm /dev/osinfo1` ตามลำดับ
- สุดท้ายให้ใช้งานคำสั่ง `make clean` เพื่อนำไฟล์ที่ไม่จำเป็นออกไปจากเครื่อง (clean)

```
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$ sudo rmmod osinfo
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$ sudo rm /dev/osinfo
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$ sudo rm /dev/osinfo1
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$ make clean
make -C /lib/modules/6.8.0-51-generic/build M=/home/reisenx/Documents/act12/osinfo clean
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-51-generic'
CLEAN    /home/reisenx/Documents/act12/osinfo/Module.symvers
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-51-generic'
reisenx@reisenx-VirtualBox:~/Documents/act12/osinfo$
```

Checkpoint #4

In this exercise, you are asked to design a character device driver that will show the vendor ID, features, and serial number of your processor.

The function for taking the vendor ID, features and serial number is provided.

Create the /dev/cpuinfo device file. The content must display the CPUID of your processor.

(Note that printf cannot be used within the kernel. You have to convert the code into a character device driver.)

จาก Note ของโจทย์จะต้องทำการเปลี่ยน code ที่โจทย์ให้มาเป็น character device driver ก่อน ซึ่งหลังจากเปลี่ยนแล้วนั้น จะได้ **cpuinfo.c** ที่มีลักษณะดังนี้

```
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/fs.h>
#include <asm/uaccess.h>
#include <linux/uaccess.h>
/* Needed by all modules */
/* Needed for KERN_INFO */

MODULE_LICENSE("GPL");
MODULE_AUTHOR("KRERK PIROMSOPA, PH.D. <Krerk.P@chula.ac.th>");
MODULE_DESCRIPTION("\cpuinfo\" Character Device");

#define DEVICENAME "cpuinfo"

static int dev_major;
static int dev_open = 0;

static char cpuinfo_buffer[256]; // buffer to store output
static char *f_ptr;

static int device_open(struct inode *inode, struct file *file)
{
    if (dev_open) return -EBUSY;
    dev_open++;

    unsigned eax, ebx, ecx, edx;

    // Vendor ID
    eax = 0;
    ecx = 0;
    native_cpuid(&eax, &ebx, &ecx, &edx);
```

```

char vendor[13];
((unsigned int *)vendor)[0] = ebx;
((unsigned int *)vendor)[1] = edx;
((unsigned int *)vendor)[2] = ecx;
vendor[12] = '\0';

// Feature info
eax = 1;
native_cpuid(&eax, &ebx, &ecx, &edx);

int stepping = eax & 0xF;
int model = (eax >> 4) & 0xF;
int family = (eax >> 8) & 0xF;
int proc_type = (eax >> 12) & 0x3;
int ext_model = (eax >> 16) & 0xF;
int ext_family = (eax >> 20) & 0xFF;

// Serial number
eax = 3;
native_cpuid(&eax, &ebx, &ecx, &edx);

// Prepare the output
snprintf(cpuinfo_buffer, sizeof(cpuinfo_buffer),
        "Vendor ID: %s\n"
        "Stepping: %d\n"
        "Model: %d\n"
        "Family: %d\n"
        "Processor Type: %d\n"
        "Extended Model: %d\n"
        "Extended Family: %d\n"
        "Serial Number: 0x%08x%08x\n",
        vendor, stepping, model, family, proc_type, ext_model, ext_family,
edx, ecx);

f_ptr = cpuinfo_buffer;
try_module_get(THIS_MODULE);
return 0;
}

static int device_release(struct inode *inode, struct file *file)
{
    dev_open--;
    module_put(THIS_MODULE);
    return 0;
}

```

```

static ssize_t device_read(struct file *filp, char *buffer, size_t length,
loff_t *offset)
{
    int bytes_read = 0;

    if (*f_ptr == 0)
        return 0;

    while (length && *f_ptr) {
        put_user(*(f_ptr++), buffer++);
        length--;
        bytes_read++;
    }

    return bytes_read;
}

static struct file_operations dev_fops = {
    .read    = device_read,
    .open    = device_open,
    .release = device_release
};

int init_module(void)
{
    printk(KERN_INFO "cpuinfo module loaded.\n");
    dev_major = register_chrdev(0, DEVICENAME, &dev_fops);

    if (dev_major < 0) {
        printk(KERN_ALERT "Registering char device failed with %d\n",
dev_major);
        return dev_major;
    }

    printk(KERN_INFO "Device MajorNumber %d.\n", dev_major);
    printk(KERN_INFO "Use: mknod /dev/cpuinfo c %d 0\n", dev_major);
    return 0;
}

void cleanup_module(void)
{
    unregister_chrdev(dev_major, DEVICENAME);
    printk(KERN_INFO "cpuinfo module unloaded.\n");
}

```

และสร้าง **Makefile** โดยมีลักษณะดังนี้

```
obj-m += cpuinfo.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
```

ใช้งานคำสั่ง **make** เพื่อให้ Makefile ทำงาน และ build ไฟล์ที่จำเป็นต่อการใช้งาน

```
reisenx@reisenx-VirtualBox:~/Documents/act12/cpuinfo$ make
make -C /lib/modules/6.8.0-51-generic/build M=/home/reisenx/Documents/act12/cpuinfo modules
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-51-generic'
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] /home/reisenx/Documents/act12/cpuinfo/cpuinfo.o
MODPOST /home/reisenx/Documents/act12/cpuinfo/Module.symvers
CC [M] /home/reisenx/Documents/act12/cpuinfo/cpuinfo.mod.o
LD [M] /home/reisenx/Documents/act12/cpuinfo/cpuinfo.ko
BTF [M] /home/reisenx/Documents/act12/cpuinfo/cpuinfo.ko
Skipping BTF generation for /home/reisenx/Documents/act12/cpuinfo/cpuinfo.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-51-generic'
```

- ใช้งานคำสั่ง **sudo insmod cpuinfo.ko** เพื่อ load kernel module ของ cpuinfo
- จากนั้นตรวจสอบ MajorNumber โดยการใช้งานคำสั่ง **sudo dmesg | grep "Device MajorNumber"**
- ใช้งานคำสั่ง **sudo mknod /dev/cpuinfo c 240 0** เพื่อสร้าง Character Device ชื่อว่า cpuinfo ในโฟลเดอร์ dev
- แสดงผลไฟล์ Character Device ที่ถูกสร้างขึ้นมาด้วยคำสั่ง **cat /dev/cpuinfo**

```
reisenx@reisenx-VirtualBox:~/Documents/act12/cpuinfo$ sudo insmod cpuinfo.ko
reisenx@reisenx-VirtualBox:~/Documents/act12/cpuinfo$ sudo dmesg | grep "Device MajorNumber"
[ 1897.912095] Device MajorNumber 240.
[ 2087.264827] Device MajorNumber 240.
reisenx@reisenx-VirtualBox:~/Documents/act12/cpuinfo$ sudo mknod /dev/cpuinfo c 240 0
reisenx@reisenx-VirtualBox:~/Documents/act12/cpuinfo$ cat /dev/cpuinfo
Vendor ID: GenuineIntel
Stepping: 5
Model: 14
Family: 6
Processor Type: 0
Extended Model: 7
Extended Family: 0
Serial Number: 0x0000000000000000
```

- ใช้งานคำสั่ง **sudo rmmod cpuinfo** เพื่อนำ kernel module ของ cpuinfo ออกจากระบบปฏิบัติการ (unload)
- ลบโฟลเดอร์และไฟล์ที่ถูกสร้างขึ้นมา (/dev/cpuinfo) ด้วยคำสั่ง **sudo rm /dev/cpuinfo**
- สุดท้ายให้ใช้งานคำสั่ง **make clean** เพื่อนำไฟล์ที่ไม่จำเป็นออกไปจากเครื่อง (clean)

```
reisenx@reisenx-VirtualBox:~/Documents/act12/cpuinfo$ sudo rmmod cpuinfo
reisenx@reisenx-VirtualBox:~/Documents/act12/cpuinfo$ sudo rm /dev/cpuinfo
reisenx@reisenx-VirtualBox:~/Documents/act12/cpuinfo$ make clean
make -C /lib/modules/6.8.0-51-generic/build M=/home/reisenx/Documents/act12/cpuinfo clean
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-51-generic'
CLEAN /home/reisenx/Documents/act12/cpuinfo/Module.symvers
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-51-generic'
reisenx@reisenx-VirtualBox:~/Documents/act12/cpuinfo$
```

Checkpoint #5

Create a character device (cpsysinfo) as a kernel module. This character device must provide information about your operating systems as follows

Minor 0: active processes on your operating system.

Minor 1: Amount of memory

Create the /dev/cp-psinfo device file (Minor 0). The file must provide pid, command of all processes.

ในขั้นตอนนี้ จะต้องสร้างไฟล์ **cpsysinfo.c** ที่มีลักษณะดังนี้

```
#include <linux/init.h>
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/fs.h>
#include <linux/uaccess.h>
#include <linux/device.h>
#include <linux/sched/signal.h> // for_each_process
#include <linux/mm.h>           // for si_meminfo
#include <linux/slab.h>         // for kzalloc

#define DEVICE_NAME "cpsysinfo"
#define CLASS_NAME "cpclass"

MODULE_LICENSE("GPL");
MODULE_AUTHOR("KRERK PIROMSOPA, PH.D. <Krerk.P@chula.ac.th>");
MODULE_DESCRIPTION("\\"cpsysinfo\\" Character Device");

static int majorNumber;
static struct class* cpClass = NULL;
static struct device* cpDevice0 = NULL;
static struct device* cpDevice1 = NULL;

static char *info_buffer;
static int info_size;
static int already_read = 0;

#define TO_KB(pages) ((pages) * (PAGE_SIZE / 1024))

static ssize_t dev_read(struct file *file, char *buffer, size_t len, loff_t
*offset) {
    int bytes_to_copy;
    if (already_read) {
        already_read = 0;
        return 0;
    }
}
```



```

bytes_to_copy = min(len, (size_t)info_size);

if (copy_to_user(buffer, info_buffer, bytes_to_copy)) {
    return -EFAULT;
}

already_read = 1;
return bytes_to_copy;
}

static int dev_open(struct inode *inodep, struct file *filep) {
    int minor = MINOR(inodep->i_rdev);
    struct task_struct *task;

    if (info_buffer)
        kfree(info_buffer);

    info_buffer = kzalloc(8192, GFP_KERNEL);
    if (!info_buffer)
        return -ENOMEM;

    info_size = 0;
    already_read = 0;

    if (minor == 0) {
        // Process list
        for_each_process(task) {
            info_size += snprintf(info_buffer + info_size, 8192 - info_size,
                                  "%d,%s\n", task->pid, task->comm);
            if (info_size >= 8192 - 100)
                break;
        }
    } else if (minor == 1) {
        // Memory info
        struct sysinfo i;
        si_meminfo(&i);

        unsigned long mem_total = TO_KB(i.totalram);
        unsigned long mem_free = TO_KB(i.freeram);
        unsigned long mem_available = TO_KB(i.freeram + i.bufferram);

        info_size = snprintf(info_buffer, 8192,
                              "MemTotal: %lu kB\n"
                              "MemFree: %lu kB\n"
                              "MemAvailable: %lu kB\n",
                              mem_total, mem_free, mem_available);
    }
}

```

```

    } else {
        snprintf(info_buffer, 8192, "Invalid minor number: %d\n", minor);
        info_size = strlen(info_buffer);
    }

    return 0;
}

static int dev_release(struct inode *inodep, struct file *filep) {
    return 0;
}

static struct file_operations fops = {
    .open = dev_open,
    .read = dev_read,
    .release = dev_release,
};

static int __init cpsysinfo_init(void) {
    majorNumber = register_chrdev(0, DEVICE_NAME, &fops);
    if (majorNumber < 0) {
        printk(KERN_ALERT "Failed to register char device\n");
        return majorNumber;
    }

    cpClass = class_create(CLASS_NAME); // For Linux 6.8+
    if (IS_ERR(cpClass)) {
        unregister_chrdev(majorNumber, DEVICE_NAME);
        return PTR_ERR(cpClass);
    }

    cpDevice0 = device_create(cpClass, NULL, MKDEV(majorNumber, 0), NULL,
DEVICE_NAME);
    cpDevice1 = device_create(cpClass, NULL, MKDEV(majorNumber, 1), NULL,
DEVICE_NAME "1");

    if (IS_ERR(cpDevice0) || IS_ERR(cpDevice1)) {
        device_destroy(cpClass, MKDEV(majorNumber, 0));
        device_destroy(cpClass, MKDEV(majorNumber, 1));
        class_destroy(cpClass);
        unregister_chrdev(majorNumber, DEVICE_NAME);
        return -1;
    }
}

```

```

    printk(KERN_INFO "cpsysinfo loaded. Major number: %d\n", majorNumber);
    printk(KERN_INFO "Use:\n");
    printk(KERN_INFO "    sudo mknod /dev/cp-psinfo c %d 0\n", majorNumber);
    printk(KERN_INFO "    sudo mknod /dev/cp-meminfo c %d 1\n", majorNumber);
    return 0;
}

static void __exit cpsysinfo_exit(void) {
    if (info_buffer)
        kfree(info_buffer);
    device_destroy(cpClass, MKDEV(majorNumber, 0));
    device_destroy(cpClass, MKDEV(majorNumber, 1));
    class_unregister(cpClass);
    class_destroy(cpClass);
    unregister_chrdev(majorNumber, DEVICE_NAME);
    printk(KERN_INFO "cpsysinfo unloaded\n");
}

module_init(cpsysinfo_init);
module_exit(cpsysinfo_exit);

```

และสร้าง **Makefile** โดยมีลักษณะดังนี้

```

obj-m += cpsysinfo.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

```

ใช้งานคำสั่ง **make** เพื่อให้ Makefile ทำงาน และ build ไฟล์ที่จำเป็นต่อการใช้งาน

```
reisenx@reisenx-VirtualBox:~/Documents/act12/cpsysinfo$ make
make -C /lib/modules/6.8.0-51-generic/build M=/home/reisenx/Documents/act12/cpsysinfo modules
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-51-generic'
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] /home/reisenx/Documents/act12/cpsysinfo/cpsysinfo.o
MODPOST /home/reisenx/Documents/act12/cpsysinfo/Module.symvers
CC [M] /home/reisenx/Documents/act12/cpsysinfo/cpsysinfo.mod.o
LD [M] /home/reisenx/Documents/act12/cpsysinfo/cpsysinfo.ko
BTF [M] /home/reisenx/Documents/act12/cpsysinfo/cpsysinfo.ko
Skipping BTF generation for /home/reisenx/Documents/act12/cpsysinfo/cpsysinfo.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-51-generic'
reisenx@reisenx-VirtualBox:~/Documents/act12/cpsysinfo$
```

- ใช้งานคำสั่ง **sudo insmod cpsysinfo.ko** เพื่อ load kernel module ของ cpsysinfo
- จากนั้นตรวจสอบ MajorNumber โดยการใช้งานคำสั่ง **sudo dmesg | grep cpsysinfo**

```
reisenx@reisenx-VirtualBox:~/Documents/act12/cpsysinfo$ sudo insmod cpsysinfo.ko
[sudo] password for reisenx:
reisenx@reisenx-VirtualBox:~/Documents/act12/cpsysinfo$ sudo dmesg | grep cpsysinfo
[ 683.565821] cpsysinfo: loading out-of-tree module taints kernel.
[ 683.565830] cpsysinfo: module verification failed: signature and/or required key missing - tainting kernel
[ 683.570726] cpsysinfo loaded. Major number: 240
reisenx@reisenx-VirtualBox:~/Documents/act12/cpsysinfo$
```

- ใช้งานคำสั่ง **sudo mknod /dev/cp-psinfo c 240 0** เพื่อสร้าง Character Device ชื่อว่า cp-psinfo ในโฟลเดอร์ dev
- แสดงผลไฟล์ Character Device ที่ถูกสร้างขึ้นมาด้วยคำสั่ง **cat /dev/cp-psinfo**
- เนื่องจากว่า device ที่สร้างมามี minor device number เท่ากับ 0 ดังนั้นมันจะแสดงผลรายการ process ที่ active ในระบบปฏิบัติการ ณ ขณะนั้น

```
reisenx@reisenx-VirtualBox:~/Documents/act12/cpsysinfo$ sudo mknod /dev/cp-psinfo c 240 0
reisenx@reisenx-VirtualBox:~/Documents/act12/cpsysinfo$ cat /dev/cp-psinfo
1,systemd
2,kthreadd
3,pool_workqueue_
4,kworker/R-rcu_g
5,kworker/R-rcu_p
6,kworker/R-slub_
7,kworker/R-netns
10,kworker/0:0H
11,kworker/u4:0
12,kworker/R-mm_pe
13,rcu_tasks_kthre
14,rcu_tasks_rude_
15,rcu_tasks_trace
16,ksoftirqd/0
17,rcu_preempt
18,migration/0
19,idle_inject/0
20,cpuhp/0
21,cpuhp/1
22,idle_inject/1
23,migration/1
24,ksoftirqd/1
28,kworker/u6:0
29,kdevtmpfs
30,kworker/R-inet_
31,kworker/u5:1
32,kauditd
33,khungtaskd
34,oom_reaper
35,kworker/R-write
36,kworker/u5:2
37,kcompactd0
38,ksmd
```

Checkpoint #6

Create the /dev/cp-meminfo device file (Minor 1). The file must provide the total amount of available memory. (The result should look similar to /proc/meminfo.)

ใน checkpoint นี้สามารถทำต่อจาก checkpoint ที่แล้วได้เลย เพียงใช้คำสั่งข้างล่างนี้ก็เป็นอันเสร็จสิ้น

- ใช้งานคำสั่ง `sudo mknod /dev/cp-meminfo c 240 1` เพื่อสร้าง Character Device ชื่อว่า cp-meminfo ในโฟลเดอร์ dev
- แสดงผลไฟล์ Character Device ที่ถูกสร้างขึ้นมาด้วยคำสั่ง `cat /dev/cp-meminfo`
- เนื่องจากว่า device ที่สร้างมามี minor device number เท่ากับ 1 ดังนั้นมันจะแสดงผลขนาดหน่วยความจำของระบบปฏิบัติการ

```
reisenx@reisenx-VirtualBox:~/Documents/act12/cpsysinfo$ sudo mknod /dev/cp-meminfo c 240 1
reisenx@reisenx-VirtualBox:~/Documents/act12/cpsysinfo$ cat /dev/cp-meminfo
MemTotal: 4010076 kB
MemFree: 1772260 kB
MemAvailable: 1839148 kB
reisenx@reisenx-VirtualBox:~/Documents/act12/cpsysinfo$
```

- ใช้งานคำสั่ง `sudo rmmod cpsysinfo` เพื่อนำ kernel module ของ cpsysinfo ออกไปจากระบบปฏิบัติการ (unload)
- ลบโฟลเดอร์และไฟล์ที่ถูกสร้างขึ้นมา (/dev/cp-psinfo และ /dev/cp-meminfo) ด้วยคำสั่ง `sudo rm /dev/cp-psinfo` และ `sudo rm /dev/cp-meminfo` ตามลำดับ
- สุดท้ายให้ใช้งานคำสั่ง `make clean` เพื่อนำไฟล์ที่ไม่จำเป็นออกไปจากเครื่อง (clean)

```
reisenx@reisenx-VirtualBox:~/Documents/act12/cpsysinfo$ sudo rmmod cpsysinfo
reisenx@reisenx-VirtualBox:~/Documents/act12/cpsysinfo$ sudo rm /dev/ps-info
reisenx@reisenx-VirtualBox:~/Documents/act12/cpsysinfo$ sudo rm /dev/mem-info
reisenx@reisenx-VirtualBox:~/Documents/act12/cpsysinfo$ make clean
make -C /lib/modules/6.8.0-51-generic/build M=/home/reisenx/Documents/act12/cpsysinfo clean
make[1]: Entering directory '/usr/src/linux-headers-6.8.0-51-generic'
CLEAN /home/reisenx/Documents/act12/cpsysinfo/Module.symvers
make[1]: Leaving directory '/usr/src/linux-headers-6.8.0-51-generic'
reisenx@reisenx-VirtualBox:~/Documents/act12/cpsysinfo$
```