

CS5250 – Advanced Operating Systems

Assignment 3

AY2018/2019 Semester 2

Name: SHI Jingli

Student ID: A0163341N

[Assignment Answer]

Part A:

1. Build and run modules

- a. module_init is called/loading when the module is loaded into the kernel(e.g. using insmod).
module_exit is called/loading when the module is being unloaded from kernel(e.g. using rmmod).

b.

- Building module command:
\$ make -C /lib/modules/`uname -r`/build M=\$pwd
- Installing module command:
\$ sudo make -C /lib/modules/`uname -r`/build M=\$pwd modules_install
- Loading module command:
\$ sudo insmod hello.ko
- Removing module command:
\$ sudo rmmod hello.ko

c.

- Building module

```
shi@e0146905: ~/advOS/ca3
File Edit View Search Terminal Help
shi@e0146905:~/advOS/ca3$ make -C /lib/modules/`uname -r`/build M=$pwd
make: Entering directory '/home/shi/Downloads/linux-5.0-rc7'
  CALL  scripts/checksyscalls.sh
  DESCEND objtool
  CHK   include/generated/compile.h
Kernel: arch/x86/boot/bzImage is ready  (#3)
Building modules, stage 2.
MODPOST 5003 modules
make: Leaving directory '/home/shi/Downloads/linux-5.0-rc7'
shi@e0146905:~/advOS/ca3$
```

```
shi@e0146905: ~/advOS/ca3
File Edit View Search Terminal Help
shi@e0146905:~/advOS/ca3$ ls
hello.c  hello.ko  hello.mod.c  hello.mod.o  hello.o  modules.order  Module.symvers
```

- Installing module

```
shi@e0146905: ~/advOS/ca3
File Edit View Search Terminal Help
shi@e0146905:~/advOS/ca3$ sudo make -C /lib/modules/`uname -r`/build M=$pwd modules_install
[sudo] password for shi:
make: Entering directory '/home/shi/Downloads/linux-5.0-rc7'
```

- Loading Module

```
shi@e0146905: ~/advOS/ca3
File Edit View Search Terminal Help
shi@e0146905:~/advOS/ca3$ sudo insmod hello.ko
shi@e0146905:~/advOS/ca3$ lsmod
Module           Size Used by
hello            16384  0
snd_intel8x0     45056  4
snd_ac97_codec   135168  1 snd_intel8x0
ac97_bus          16384  1 snd_ac97_codec
snd_pcm           102400  2 snd_intel8x0,snd_ac97_codec
```

- Removing Module

```
shi@e0146905: ~/advOS/ca3
File Edit View Search Terminal Help
shi@e0146905:~/advOS/ca3$ sudo rmmod hello.ko
shi@e0146905:~/advOS/ca3$
```

- Printk message

```
shi@e0146905: ~/advOS/ca3
File Edit View Search Terminal Help
shi@e0146905:~/advOS/ca3$ dmesg | tail -2
[ 6285.993483] Hello, world
[ 6294.295783] Goodbye, cruel world
shi@e0146905:~/advOS/ca3$
```

d. Add <who> parameter

The new module code is as below.

```
shi@e0146905: ~/advOS/ca3
```

```
File Edit View Search Terminal Help
1 #include <linux/kernel.h>
2 #include <linux/init.h>
3 #include <linux/module.h>
4
5 // define the module argument and default value
6 static char *name = "world";
7 // charp = char ptr, S_IRUGO can be readonly
8 module_param(name, charp, S_IRUGO);
9 MODULE_PARM_DESC(name,"pass value to name.");
10 MODULE_LICENSE("GPL");
11 static int hello_init(void)
12 {
13     printk(KERN_ALERT "Hello %s\n",name);
14     return 0;
15 }
16
17 static void hello_exit(void)
18 {
19     printk(KERN_ALERT "Goodbye, cruel world\n");
20 }
21
22 module_init(hello_init);
23 module_exit(hello_exit);
```

after build and install the new module, we load the module by passing ‘A0163341N’ to parameter *name*.

```
shi@e0146905: ~/advOS/ca3
```

```
File Edit View Search Terminal Help
shi@e0146905:~/advOS/ca3$ sudo insmod hello.ko name=A0163341N
shi@e0146905:~/advOS/ca3$
```

Then check the printk message.

```
shi@e0146905: ~/advOS/ca3
```

```
File Edit View Search Terminal Help
shi@e0146905:~/advOS/ca3$ sudo insmod hello.ko name=A0163341N
shi@e0146905:~/advOS/ca3$ dmesg | tail -2
[10443.138230] Goodbye, cruel world
[10447.939732] Hello A0163341N
shi@e0146905:~/advOS/ca3$
```

2. Build a device model

a.

```
shi@e0146905: ~
```

```
File Edit View Search Terminal Help
shi@e0146905:~$ sudo mknod /dev/onebyte c 61 0
[sudo] password for shi:
shi@e0146905:~$
```

device name is **onebyte**, the type c means character device, and 61 and 0 are major and minor of the device respectively.

b. \$ ls -l /dev

File Edit View Search Terminal Help						
drwxr-xr-x	2	root	root	60	Mar 27 19:37	mapper
crw----	1	root	root	10,	227 Mar 27 19:37	mcelog
crw-r----	1	root	kmem	1,	1 Mar 27 19:37	mem
crw----	1	root	root	10,	56 Mar 27 19:37	memory_bandwidth
drwxrwxrwt	2	root	root	40	Mar 27 19:37	mqueue
drwxr-xr-x	2	root	root	60	Mar 27 19:37	net
crw----	1	root	root	10,	58 Mar 27 19:37	network_latency
crw----	1	root	root	10,	57 Mar 27 19:37	network_throughput
crw-rw-rw-	1	root	root	1,	3 Mar 27 19:37	null
crw-r--r--	1	root	root	61,	0 Mar 27 19:40	onebyte
crw-r----	1	root	kmem	1,	4 Mar 27 19:37	port
crw----	1	root	root	108,	0 Mar 27 19:37	ppp
crw----	1	root	root	10,	1 Mar 27 19:37	psaux
crw-rw-rw-	1	root	tty	5,	2 Mar 27 19:41	ptmx
drwxr-xr-x	2	root	root	0	Mar 27 19:37	pts

c.

The completed read and write functions.

```
File Edit View Search Terminal Help
39
40 ssize_t onebyte_read(struct file *filep, char *buf, size_t count, loff_t *f_pos)
41 {
42     /*please complete the function on your own*/
43     int copy_err = 0;
44     // check if onebyte_data is null
45     if(!onebyte_data)
46     {
47         printk(KERN_WARNING "ONEBYTE is null.\n");
48         return -EINVAL;
49     }
50
51     // copy data
52     copy_err = copy_to_user(buf, onebyte_data,1);
53     if(copy_err)
54     {
55         printk(KERN_ERR "Fail to copy data.\n");
56         return copy_err;
57     }
58     // set reading position of where to start reading the file
59     if(*f_pos == 0)
60     {
61         *f_pos += 1;
62         return 1;
63     }
64     else
65         return 0;
66 }
```

```
shi@e0146905: ~/advOS/ca3/CS5250-CA3
File Edit View Search Terminal Help
67
68 ssize_t onebyte_write(struct file *filep, const char *buf, size_t count, loff_t *f_pos)
69 {
70     /*please complete the function on your own*/
71     // check if user data is null
72     if(!buf)
73     {
74         printk(KERN_WARNING "buf is null.\n");
75         return -EINVAL;
76     }
77     // check data length to be written
78     if(count <=0)
79     {
80         printk(KERN_ERR "The length of data to be written is invalid.\n");
81         return -EINVAL;
82     }
83     copy_from_user(onebyte_data, buf, 1);
84     if(count > 1)
85     {
86         printk(KERN_ERR "The length of data to be written is more than one byte.\n");
87         return -ENOSPC;
88     }
89     // set write position of where to start write the file
90     if(*f_pos == 0)
91     {
92         *f_pos += 1;
93         return 1;
94     }
95     else
96         return 0;
97 }
```

The Makefile is as below

```
shi@e0146905: ~/advOS/ca3/CS5250-CA3
File Edit View Search Terminal Help
1 obj-m+=onebyte.o
2
3 all:
4     make -C /lib/modules/`uname -r`/build/ M=$(PWD) modules
5 clean:
6     make -C /lib/modules/`uname -r`/build/ M=$(PWD) clean
```

The onebyte.c and Makefile is uploaded to github and the link is

<https://github.com/SHIJINGLI0206/CS5250-CA3.git>

The commits history is as below.

```

shi@e0146905: ~/advOS/ca3/CS5250-CA3

File Edit View Search Terminal Help
commit 65da40e5099c66c75bdd5346a3605fba4fd5fcbb (HEAD -> master, origin/master, origin/HEAD)
Author: SHI Jingli <jinglishi0206@gmail.com>
Date:   Thu Mar 28 11:34:48 2019 +0800

    fix issue that no output from cat

commit f8d7a80911ae98c691770335d9a0ba6c4d0445b2
Author: jinglishi <JINGLISHI@sjl.lan>
Date:   Thu Mar 28 00:38:30 2019 +0800

    replace copy_to_user with put_fs_byte, copy_from_user with get_fs_byte.

commit 54f39e55011b3feade7d72a28c0a68784ffc551a
Author: jinglishi <JINGLISHI@sjl.lan>
Date:   Wed Mar 27 23:48:38 2019 +0800

    update pwd path in Makefile.

commit 3467900f2ccb3b1b5b8fe9ce4ba6d8261cefde3d
Author: jinglishi <JINGLISHI@sjl.lan>
Date:   Wed Mar 27 23:41:40 2019 +0800

    update Makefile.

commit 935867801ac2ddd377d017f9ccb4cc28dd4b8055
Author: jinglishi <JINGLISHI@sjl.local>
Date:   Wed Mar 27 21:17:12 2019 +0800

    update write function to return EFBIG.

commit 52d09011c36d714f15decf6595c8ef42a4ad9129
Author: jinglishi <JINGLISHI@sjl.local>
Date:   Wed Mar 27 20:56:19 2019 +0800

    add makefile

commit acc1cd91c08db222bba62136a258df13b82745d9
Author: jinglishi <JINGLISHI@sjl.local>
Date:   Wed Mar 27 20:40:58 2019 +0800

    implement read and write functions.

commit 09c276c26963d3acf2f6eb3c3ecc8c2f021b5ee7
Author: jinglishi <JINGLISHI@sjl.local>
Date:   Wed Mar 27 19:38:22 2019 +0800

    change name to onebyte.c

commit 4e69f85d63914764197959b10d7d218ca72ddc9f
Author: jinglishi <JINGLISHI@sjl.local>
Date:   Wed Mar 27 19:20:58 2019 +0800

    change file name to devca3.c

commit 791f1a88876a2a7ad796d95ae869073375551a57
Author: jinglishi <JINGLISHI@sjl.lan>
Date:   Wed Mar 27 00:23:24 2019 +0800

    first commit

commit 834d3dadf33b515b8a5b635681409d841f31988f
Author: jinglishi <JINGLISHI@sjl.lan>
Date:   Wed Mar 27 00:21:18 2019 +0800

    first commit
(END)

```

Then build compile the module, and load the module into kernel.

```
shi@e0146905: ~/advOS/ca3/CS5250-CA3
File Edit View Search Terminal Help
shi@e0146905:~/advOS/ca3/CS5250-CA3$ ls
Makefile onebyte.c README.md
shi@e0146905:~/advOS/ca3/CS5250-CA3$ make
make -C /lib/modules/`uname -r`/build/ M=/home/shi/advOS/ca3/CS5250-CA3 modules
make[1]: Entering directory '/home/shi/Downloads/linux-5.0-rc7'
  CC [M]  /home/shi/advOS/ca3/CS5250-CA3/onebyte.o
  Building modules, stage 2.
MODPOST 1 modules
  CC      /home/shi/advOS/ca3/CS5250-CA3/onebyte.mod.o
  LD [M]  /home/shi/advOS/ca3/CS5250-CA3/onebyte.ko
make[1]: Leaving directory '/home/shi/Downloads/linux-5.0-rc7'
shi@e0146905:~/advOS/ca3/CS5250-CA3$ sudo insmod onebyte.ko
[sudo] password for shi:
shi@e0146905:~/advOS/ca3/CS5250-CA3$
```

Finally, the testing cases are as below.

```
root@e0146905: ~
File Edit View Search Terminal Help
root@e0146905:~# cat /dev/onebyte
Xroot@e0146905:~# printf a>/dev/onebyte
root@e0146905:~# cat /dev/onebyte
aroot@e0146905:~# printf b>/dev/onebyte
root@e0146905:~# cat /dev/onebyte
broot@e0146905:~# printf abc>/dev/onebyte
-bash: printf: write error: No space left on device
root@e0146905:~# cat /dev/onebyte
aroot@e0146905:~#
```

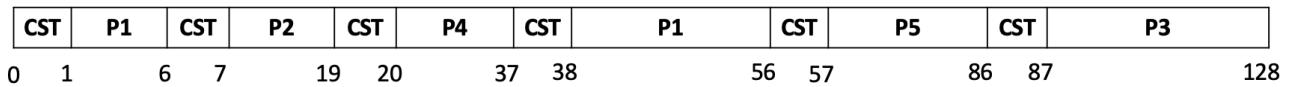
Part B.

Process	Burst CPU Time	Arrival Time
P1	23	0
P2	12	5
P3	41	10
P4	17	15
P5	29	40

1. (a) Pre-emptive Shortest Remaining Time First

The execution schedule is as below.

CST: Context Switching Time (1 time unit)



The completion time:

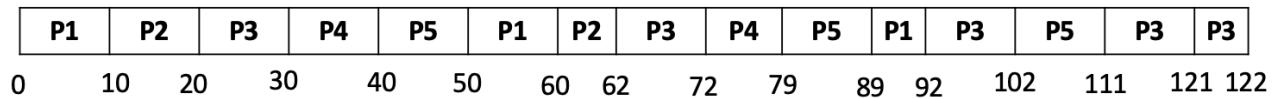
Process	Completion Time
P1	56
P2	19

P3	128
P4	37
P5	86

1. (b) Round Robin

The execution schedule is as below.

Quantum = 10



The completion time:

Process	Completion Time
P1	92
P2	62
P3	122
P4	79
P5	111

2. Draw Red-Black Tree

After nodes are inserted, the final red-black tree is as below.

