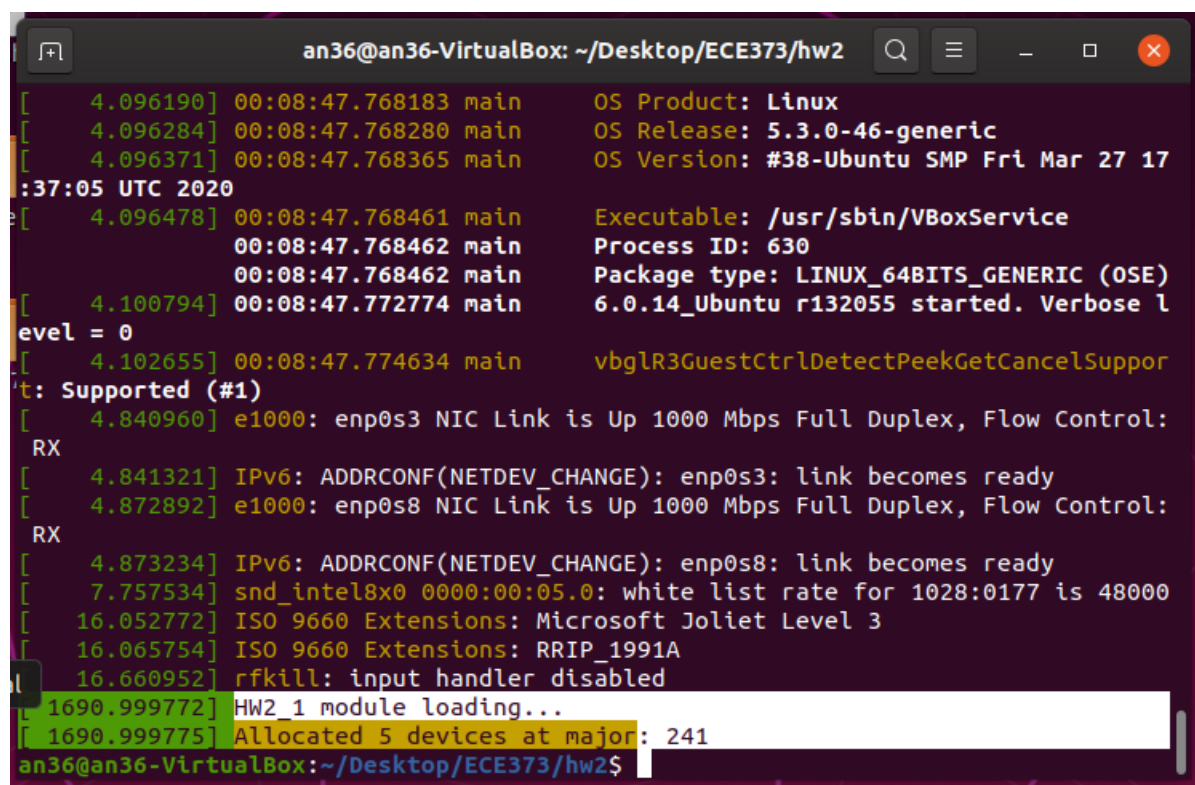


Abdullah Almarzouq
4/22/20

Assignment #2

1. Setting up the char device

In this first step, I will be creating a kernel module that creates and register a single character device. The character device will be dynamically allocated. I will also create a device node file in /dev using the 'mknod' command. The kernel module will load and do nothing at this point.



```

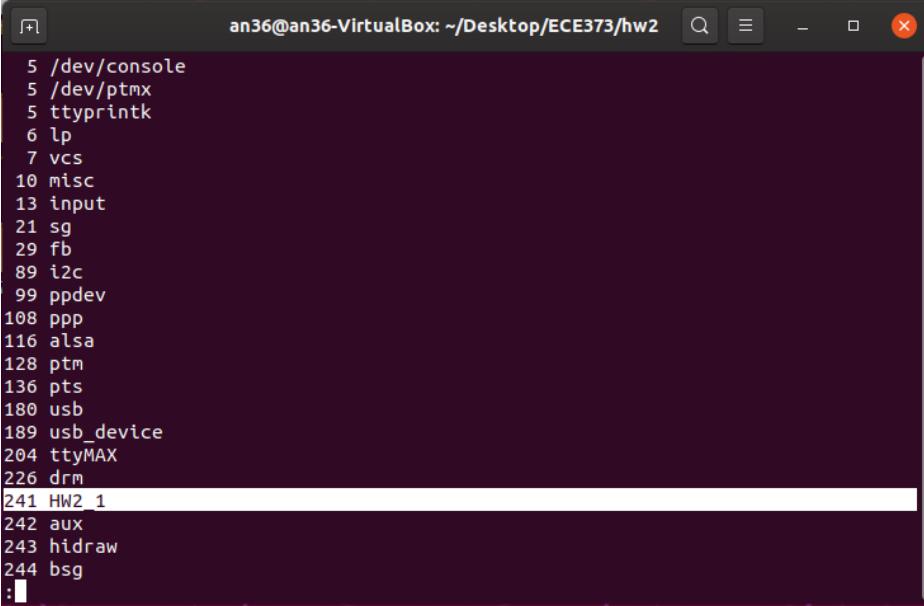
an36@an36-VirtualBox: ~/Desktop/ECE373/hw2
[ 4.096190] 00:08:47.768183 main OS Product: Linux
[ 4.096284] 00:08:47.768280 main OS Release: 5.3.0-46-generic
[ 4.096371] 00:08:47.768365 main OS Version: #38-Ubuntu SMP Fri Mar 27 17
:37:05 UTC 2020
[ 4.096478] 00:08:47.768461 main Executable: /usr/sbin/VBoxService
00:08:47.768462 main Process ID: 630
00:08:47.768462 main Package type: LINUX_64BITS_GENERIC (OSE)
[ 4.100794] 00:08:47.772774 main 6.0.14_Ubuntu r132055 started. Verbose l
evel = 0
[ 4.102655] 00:08:47.774634 main vbglR3GuestCtrlDetectPeekGetCancelSuppor
t: Supported (#1)
[ 4.840960] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control:
RX
[ 4.841321] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s3: link becomes ready
[ 4.872892] e1000: enp0s8 NIC Link is Up 1000 Mbps Full Duplex, Flow Control:
RX
[ 4.873234] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s8: link becomes ready
[ 7.757534] snd_intel8x0 0000:00:05.0: white list rate for 1028:0177 is 48000
[ 16.052772] ISO 9660 Extensions: Microsoft Joliet Level 3
[ 16.065754] ISO 9660 Extensions: RRIP_1991A
[ 16.660952] rfkill: input handler disabled
[ 1690.999772] HW2_1 module loading...
[ 1690.999775] Allocated 5 devices at major: 241
an36@an36-VirtualBox:~/Desktop/ECE373/hw2$

```

Figure1: 'dmesg' shows message printed to kernel

In the figure above, we can see that the module (named HW2_1) printed two messages:

- "HW2_1 module loading..."
- "Allocate 5 devices at major: 241"



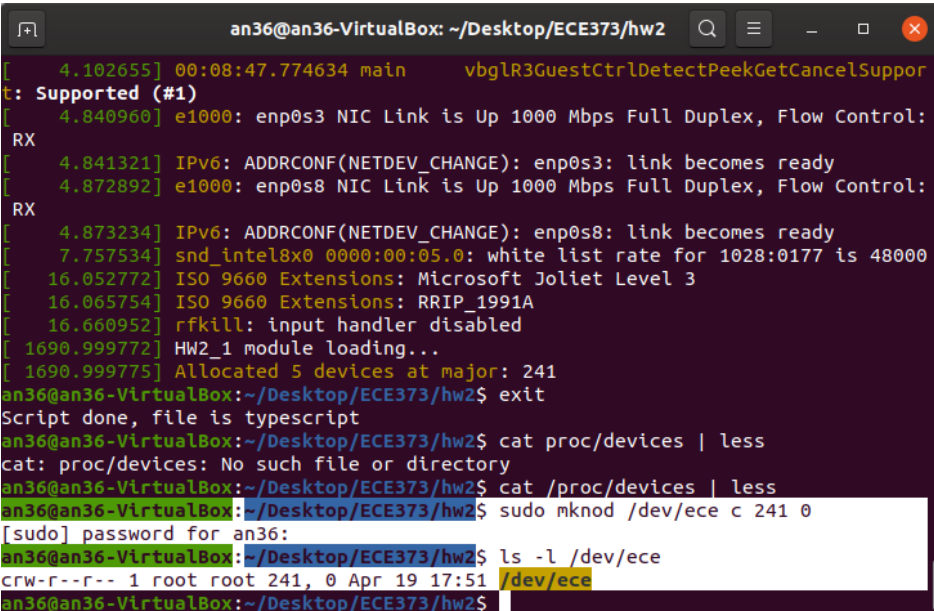
```

an36@an36-VirtualBox: ~/Desktop/ECE373/hw2
5 /dev/console
5 /dev/ptmx
5 ttyprintk
6 lp
7 vcs
10 misc
13 input
21 sg
29 fb
89 i2c
99 ppdev
108 ppp
116 alsa
128 ptm
136 pts
180 usb
189 usb_device
204 ttyMAX
226 drm
241 HW2_1
242 aux
243 hidraw
244 bsg
:

```

Figure2: HW2_1 device major number

The figure above shows out device major number (“241”) under “/proc/devices”.



```

an36@an36-VirtualBox: ~/Desktop/ECE373/hw2
[ 4.102655] 00:08:47.774634 main vbglR3GuestCtrlDetectPeekGetCancelSupport: Supported (#1)
[ 4.840960] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[ 4.841321] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s3: link becomes ready
[ 4.872892] e1000: enp0s8 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[ 4.873234] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s8: link becomes ready
[ 7.757534] snd_intel8x0 0000:00:05.0: white list rate for 1028:0177 is 48000
[ 16.052772] ISO 9660 Extensions: Microsoft Joliet Level 3
[ 16.065754] ISO 9660 Extensions: RRIP_1991A
[ 16.660952] rfskill: input handler disabled
[ 1690.999772] HW2_1 module loading...
[ 1690.999775] Allocated 5 devices at major: 241
an36@an36-VirtualBox:~/Desktop/ECE373/hw2$ exit
Script done, file is typescript
an36@an36-VirtualBox:~/Desktop/ECE373/hw2$ cat /proc/devices | less
cat: /proc/devices: No such file or directory
an36@an36-VirtualBox:~/Desktop/ECE373/hw2$ cat /proc/devices | less
an36@an36-VirtualBox:~/Desktop/ECE373/hw2$ sudo mknod /dev/ece c 241 0
[sudo] password for an36:
an36@an36-VirtualBox:~/Desktop/ECE373/hw2$ ls -l /dev/ece
crw-r--r-- 1 root root 241, 0 Apr 19 17:51 /dev/ece
an36@an36-VirtualBox:~/Desktop/ECE373/hw2$

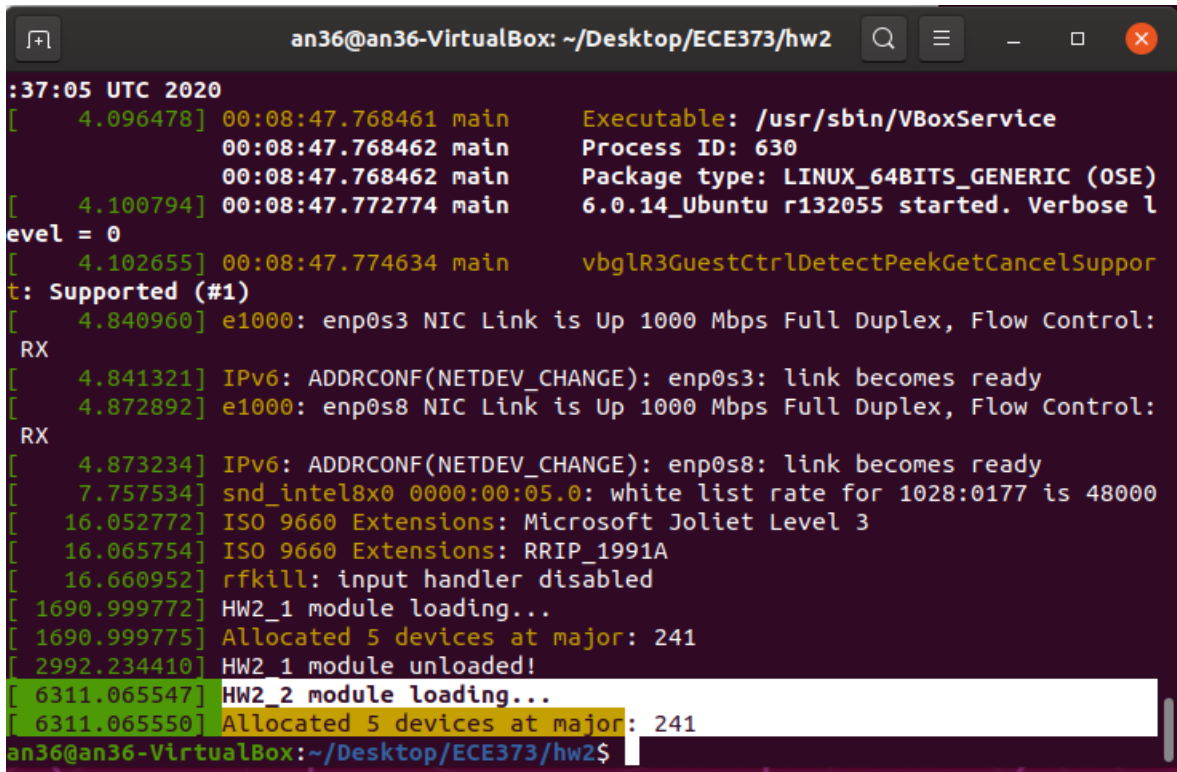
```

Figure3: making a node using “mknod” command

Now, the figure above shows that we created a new node under “/dev” using the command “mknod /dev/ece c 241 0”. Also, as it can be seen from the figure above, the node has been created as the “ls -l /dev/ece” command shows that a character device node has been created successfully

2. Hooking up file operations

At this point I've created a new .c file called "HW2_2.c" which adds an integer variable called "syscall_val" to the char device struct. The "syscall_val" will be initialized at 40. Also, this new "HW2_2.c" file will have a read() callback function which will read "syscall_val" and return it to userspace. And it'll also have a write() callback function which will write a new value to "syscall_val" and print it. This new .c file will have a new node file which will be created using "mknod" command as it might have a new major device number.



```

an36@an36-VirtualBox: ~/Desktop/ECE373/hw2
:37:05 UTC 2020
[ 4.096478] 00:08:47.768461 main Executable: /usr/sbin/VBoxService
[ 4.096478] 00:08:47.768462 main Process ID: 630
[ 4.096478] 00:08:47.768462 main Package type: LINUX_64BITS_GENERIC (0SE)
[ 4.100794] 00:08:47.772774 main 6.0.14_Ubuntu r132055 started. Verbose l
level = 0
[ 4.102655] 00:08:47.774634 main vbglR3GuestCtrlDetectPeekGetCancelSupport: Supported (#1)
[ 4.840960] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control:
RX
[ 4.841321] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s3: link becomes ready
[ 4.872892] e1000: enp0s8 NIC Link is Up 1000 Mbps Full Duplex, Flow Control:
RX
[ 4.873234] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s8: link becomes ready
[ 7.757534] snd_intel8x0 0000:00:05.0: white list rate for 1028:0177 is 48000
[ 16.052772] ISO 9660 Extensions: Microsoft Joliet Level 3
[ 16.065754] ISO 9660 Extensions: RRIP_1991A
[ 16.660952] rfkill: input handler disabled
[ 1690.999772] HW2_1 module loading...
[ 1690.999775] Allocated 5 devices at major: 241
[ 2992.234410] HW2_1 module unloaded!
[ 6311.065547] HW2_2 module loading...
[ 6311.065550] Allocated 5 devices at major: 241
an36@an36-VirtualBox:~/Desktop/ECE373/hw2$

```

Figure4: loading HW2_2 module

As shown in figure4, the HW2_2 module has been loaded into the kernel with the same major number (241) as the HW2_1 module. The HW2_1 module has been unloaded prior to loading the new HW2_2 module. Also, the typescript which is called "typescript1", included in the same directory/file as this pdf, shows the list of devices under "/proc/devices".

```

an36@an36-VirtualBox: ~/Desktop/ECE373/hw2
[ 4.100794] 00:08:47.768462 main Package type: LINUX_64BITS_GENERIC (0SE)
[ 4.102655] 00:08:47.772774 main 6.0.14_Ubuntu r132055 started. Verbose l
level = 0
[ 4.102655] 00:08:47.774634 main vbglR3GuestCtrlDetectPeekGetCancelSuppor
t: Supported (#1)
[ 4.840960] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control:
RX
[ 4.841321] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s3: link becomes ready
[ 4.872892] e1000: enp0s8 NIC Link is Up 1000 Mbps Full Duplex, Flow Control:
RX
[ 4.873234] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s8: link becomes ready
[ 7.757534] snd_intel8x0 0000:00:05.0: white list rate for 1028:0177 is 48000
[ 16.052772] ISO 9660 Extensions: Microsoft Joliet Level 3
[ 16.065754] ISO 9660 Extensions: RRIP_1991A
[ 16.660952] rfkill: input handler disabled
[ 1690.999772] HW2_1 module loading...
[ 1690.999775] Allocated 5 devices at major: 241
[ 2992.234410] HW2_1 module unloaded!
[ 6311.065547] HW2_2 module loading...
[ 6311.065550] Allocated 5 devices at major: 241
an36@an36-VirtualBox:~/Desktop/ECE373/hw2$ sudo mknod /dev/ece c 241 0
an36@an36-VirtualBox:~/Desktop/ECE373/hw2$ ls -l /dev/ece
crw-r--r-- 1 root root 241, 0 Apr 19 19:01 /dev/ece
an36@an36-VirtualBox:~/Desktop/ECE373/hw2$

```

Figure5: Creating new node file under /dev

The figure above shows that we created a new “/dev/ece” node for “HW2_2” module which has the same major number as the previously created node (which has been removed prior to creating this new node file).

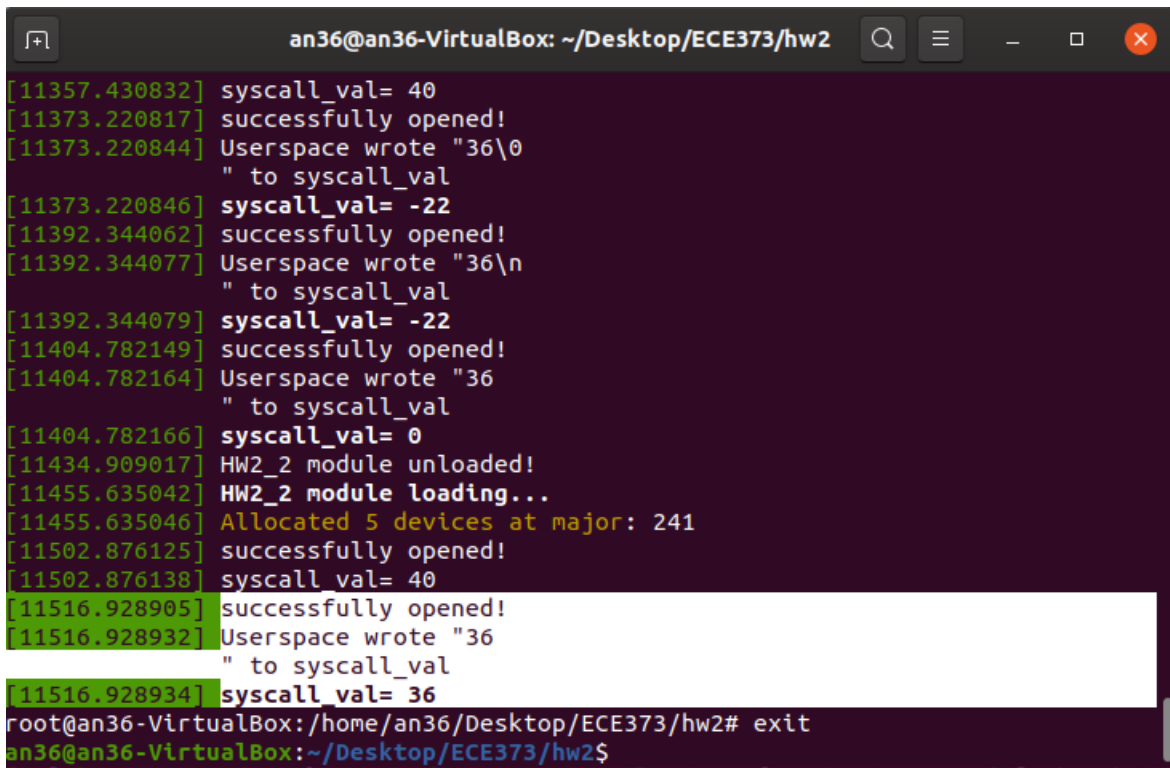
```

an36@an36-VirtualBox: ~/Desktop/ECE373/hw2
level = 0
[ 4.102655] 00:08:47.774634 main vbglR3GuestCtrlDetectPeekGetCancelSuppor
t: Supported (#1)
[ 4.840960] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control:
RX
[ 4.841321] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s3: link becomes ready
[ 4.872892] e1000: enp0s8 NIC Link is Up 1000 Mbps Full Duplex, Flow Control:
RX
[ 4.873234] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s8: link becomes ready
[ 7.757534] snd_intel8x0 0000:00:05.0: white list rate for 1028:0177 is 48000
[ 16.052772] ISO 9660 Extensions: Microsoft Joliet Level 3
[ 16.065754] ISO 9660 Extensions: RRIP_1991A
[ 16.660952] rfkill: input handler disabled
[ 1690.999772] HW2_1 module loading...
[ 1690.999775] Allocated 5 devices at major: 241
[ 2992.234410] HW2_1 module unloaded!
[ 6311.065547] HW2_2 module loading...
[ 6311.065550] Allocated 5 devices at major: 241
[ 7874.814759] HW2_2 module unloaded!
[ 9879.536623] HW2_2 module loading...
[ 9879.536624] Allocated 5 devices at major: 241
[ 9989.142008] successfully opened!
[ 9989.142016] syscall_val= 40
an36@an36-VirtualBox:~/Desktop/ECE373/hw2$

```

Figure6: printing syscall_val value to kernel

Figure6 shows that we successfully read (using “cat”) and printed the value of “syscall_val” which is 40.



```

an36@an36-VirtualBox: ~/Desktop/ECE373/hw2
[11357.430832] syscall_val= 40
[11373.220817] successfully opened!
[11373.220844] Userspace wrote "36\0"
               " to syscall_val
[11373.220846] syscall_val= -22
[11392.344062] successfully opened!
[11392.344077] Userspace wrote "36\n"
               " to syscall_val
[11392.344079] syscall_val= -22
[11404.782149] successfully opened!
[11404.782164] Userspace wrote "36"
               " to syscall_val
[11404.782166] syscall_val= 0
[11434.909017] HW2_2 module unloaded!
[11455.635042] HW2_2 module loading...
[11455.635046] Allocated 5 devices at major: 241
[11502.876125] successfully opened!
[11502.876138] syscall_val= 40
[11516.928905] successfully opened!
[11516.928932] Userspace wrote "36"
               " to syscall_val
[11516.928934] syscall_val= 36
root@an36-VirtualBox:/home/an36/Desktop/ECE373/hw2# exit
an36@an36-VirtualBox:~/Desktop/ECE373/hw2$

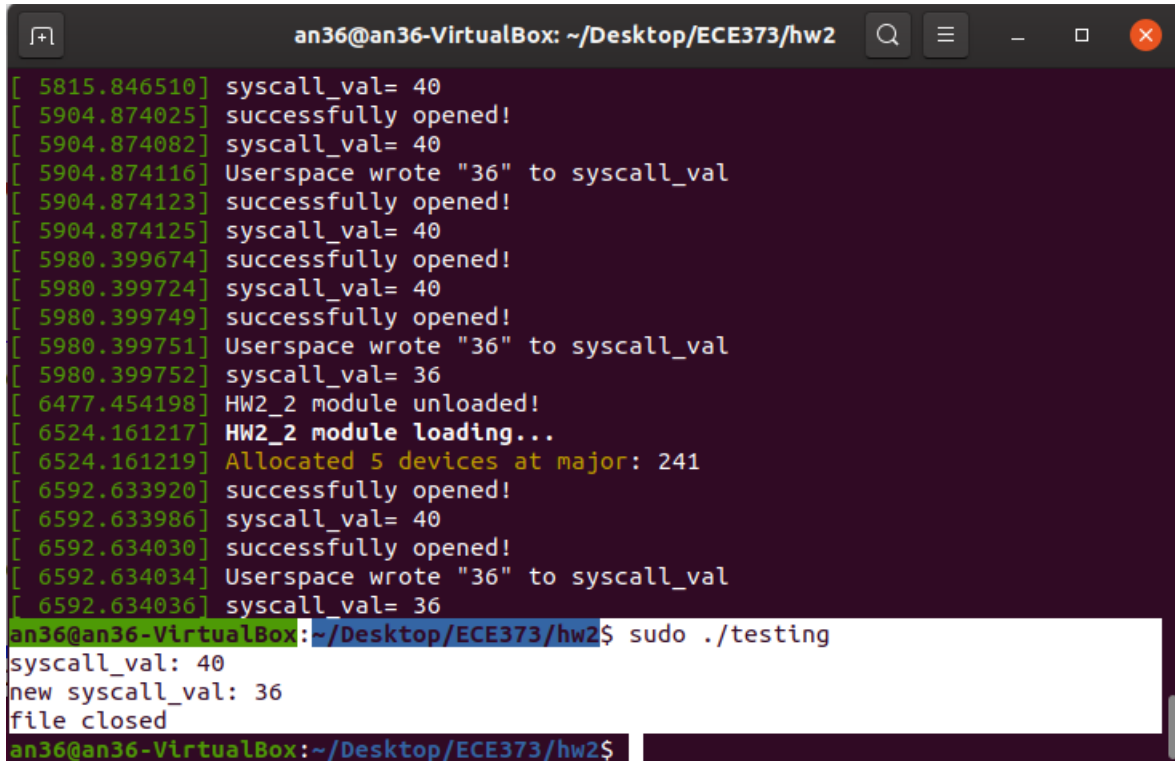
```

Figure7: writing to HW2_2 module

Figure7 shows that we can successfully write/echo to our kernel module. When we enter the command “echo “36” > /dev/ece”, the value of “syscall_val” changes from 40 to 36 as shown.

3. Testing, testing...

To test the HW2_2.c kernel module, I've created a userspace program called "testing.c". This userspace program will open the node file "/dev/ece", read from it, print the read value and then close it. After that, the userspace program will open the node file again, write to it (to modify syscall_val) then read from it, print the new syscall_val and close the node file again.



```

an36@an36-VirtualBox: ~/Desktop/ECE373/hw2
[ 5815.846510] syscall_val= 40
[ 5904.874025] successfully opened!
[ 5904.874082] syscall_val= 40
[ 5904.874116] Userspace wrote "36" to syscall_val
[ 5904.874123] successfully opened!
[ 5904.874125] syscall_val= 40
[ 5980.399674] successfully opened!
[ 5980.399724] syscall_val= 40
[ 5980.399749] successfully opened!
[ 5980.399751] Userspace wrote "36" to syscall_val
[ 5980.399752] syscall_val= 36
[ 6477.454198] HW2_2 module unloaded!
[ 6524.161217] HW2_2 module loading...
[ 6524.161219] Allocated 5 devices at major: 241
[ 6592.633920] successfully opened!
[ 6592.633986] syscall_val= 40
[ 6592.634030] successfully opened!
[ 6592.634034] Userspace wrote "36" to syscall_val
[ 6592.634036] syscall_val= 36
an36@an36-VirtualBox: ~/Desktop/ECE373/hw2$ sudo ./testing
syscall_val: 40
new syscall_val: 36
file closed
an36@an36-VirtualBox: ~/Desktop/ECE373/hw2$

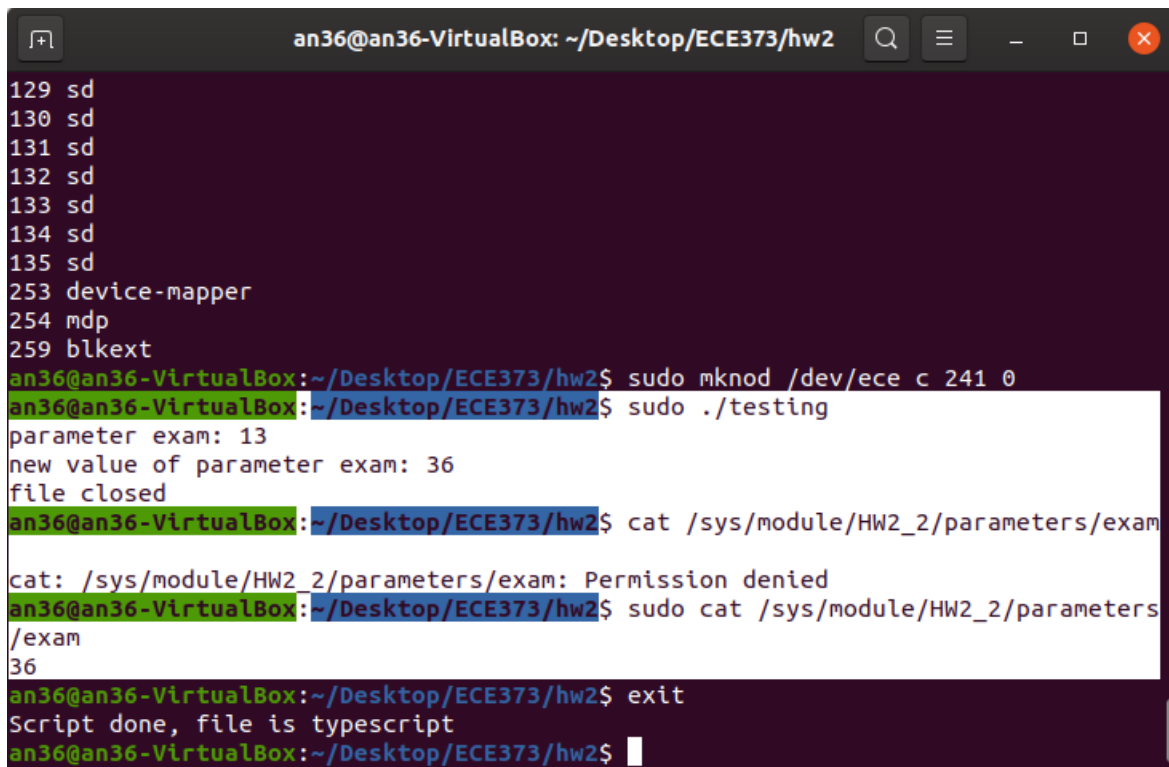
```

Figure8: executing testing.c

As it can be seen from the figure above, the value of syscall_val gets printed, modified then printed again. Also, a typescript file, included in the same directory as this pdf, called "typescript2" which shows the module being loaded then tested with the userspace program "testing.c".

4. Changes

At this point, I added a module parameter called “exam” to the kernel module “HW2_2.c” and I modified the “syscall_val” value to be equal to the value of “exam”. This parameter (“exam”) will be initialized while reloading the kernel module. Moreover, the userspace program “testing.c” will be updated so that it opens “/sys/module/HW2_2/parameters/exam”, reads the value of “exam” then modifies it so that the “syscall_val” value gets modified.



```

an36@an36-VirtualBox: ~/Desktop/ECE373/hw2
129 sd
130 sd
131 sd
132 sd
133 sd
134 sd
135 sd
253 device-mapper
254 mdp
259 blkext
an36@an36-VirtualBox:~/Desktop/ECE373/hw2$ sudo mknod /dev/ece c 241 0
an36@an36-VirtualBox:~/Desktop/ECE373/hw2$ sudo ./testing
parameter exam: 13
new value of parameter exam: 36
file closed
an36@an36-VirtualBox:~/Desktop/ECE373/hw2$ cat /sys/module/HW2_2/parameters/exam
cat: /sys/module/HW2_2/parameters/exam: Permission denied
an36@an36-VirtualBox:~/Desktop/ECE373/hw2$ sudo cat /sys/module/HW2_2/parameters
/exam
36
an36@an36-VirtualBox:~/Desktop/ECE373/hw2$ exit
Script done, file is typescript
an36@an36-VirtualBox:~/Desktop/ECE373/hw2$

```

Figure9: testing with parameter

In figure9, the value of parameter “exam” was initialized to 13 while reloading the module. Then, the userspace program (testing.c) was executed to read, print and modify the value of “exam”. As it can be seen, the userspace program successfully read, printed, modified and printed again the value of parameter “exam”. Lastly, the “cat” command was used to show the current value of parameter “exam”, which is equal to 36 after been modified. The typescript which includes all the results of this step of the homework is included in the same directory as this pdf and it is called “typescript3”.