Assignment 4 : Linux Kernel Module

Name: Bhanuj Gandhi
Roll no.: 2022201068

In this assignment, our task was to download a module, and modify it to the total number, number of running, number of interruptable, and number of uninterruptible tasks.

Steps Followed

1. Create a directory named say `assignment_4`

```
$ mkdir assignment_4
$ cd assignment_4
```

2. Install all the packages that are required in the process of kernel installation.

```
$ sudo apt update
$ sudo apt install binutils gcc make
```

3. Download the boilerplate code for the kernel module

```
$ wget http://faculty.washington.edu/wlloyd/courses/tcss422/assignments/hello_module.tar.gz
```

4. Unzip the tar.gz file using tar command utility

```
$ tar xzf hello_module.tar.gz
```

5. Change the directory to newly created module folder.

```
$ cd hello_module
```

6. hello_module contains "**helloModule.c**", which has a starter code for the Linux kernel module which says *the module is initialized* upon installing the module and says *cleaning up the module* upon removing.
7. **module_init:** This is the function that runs when the module is installed in the kernel. All the initializations happen here.
8. **module_exit:** This function runs when the module is removed using *rmmod* command.
9. Run the ***make*** file command to compile and explore how kernel messages are printed in ***dmesg*** utility.

Counting the number of tasks running in the Linux

1. Upon exploring the boilerplate module, I created my own module which counts the number of tasks running in the Linux.
2. For this activity, we make use of **task_struct** Linux kernel data structure. It is used for the inspection of running threads and processes.
3. There is a helper function **for_each_process** which takes the task_struct pointer and iterates over all the processes currently in the process table.
4. I have used __state member variable of *task_struct* which depicts the type of the process.
5. Type of the process can be found from __state variable which is defined as volatile long
   0 means Running
   1 means Interruptable
   2 means Uninterruptable

countprocessmodule.c

```
#include <linux/cdev.h>
#include <linux/module.h>
#include <linux/pid_namespace.h>
#include <linux/proc_fs.h>
#include <linux/sched/signal.h>
#include <linux/slab.h>


/*
Method to count number of processes
Type of the process can be found from __state variable which is defined as volatile long
0 -> Running
1 -> Interruptable
2 -> Uninterruptable
*/
void count_proc(void) {
  int total = 0, running = 0, interruptable = 0, uninterruptible = 0;
    struct task_struct *proc;

    for_each_process(proc) {
        total++;
        if(proc->__state == TASK_RUNNING)
          running++;
        else if(proc->__state == TASK_INTERRUPTIBLE)
          interruptable++;
        else if(proc->__state == TASK_UNINTERRUPTIBLE)
          uninterruptible++;
    }
    printk(KERN_INFO "countprocessmodule: Total number of processes: %d\n", total);
    printk(KERN_INFO "countprocessmodule: Number of running processes: %d\n", running);
    printk(KERN_INFO "countprocessmodule: Number of interruptible processes: %d\n", interruptable);
    printk(KERN_INFO "countprocessmodule: Number of uninterruptible processes: %d\n", uninterruptible);

}

int proc_init(void) {
    printk(KERN_INFO "countprocessmodule: Initialising count process module\n");
    count_proc();
    return 0;
}

void proc_cleanup(void) {
    printk(KERN_INFO "countprocessmodule: performing cleanup of module\n");
}

MODULE_LICENSE("MIT");
MODULE_AUTHOR("Bhanuj Gandhi");
MODULE_DESCRIPTION("A module that counts number of tasks running, interrupt-able, and uniterruptible.");

module_init(proc_init);
module_exit(proc_cleanup);
```

Makefile

```
CONFIG_MODULE_SIG=n

obj-m += countprocessmodule.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
```

Steps to execute:

1. Compile the code using MakeFile

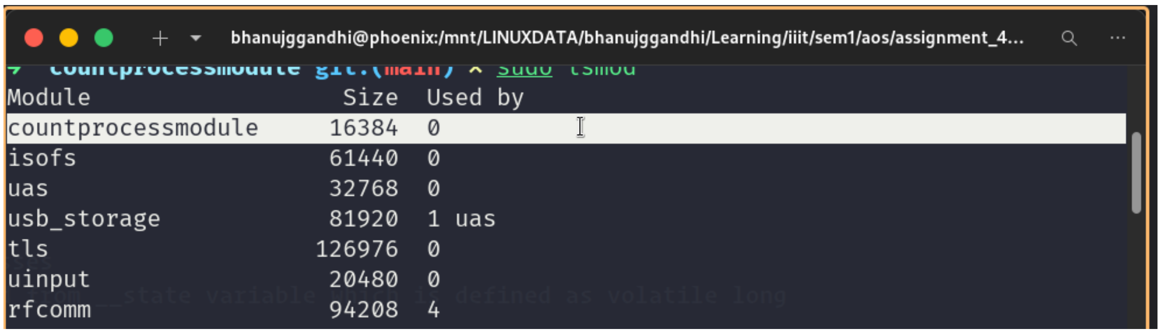
   ```
   $ make
   ```

2. Install the module using *`insmod`* utility.

   ```
   $ sudo insmod countprocessmodule.ko
   ```

3. Check if the module is successfully installed using `lsmod` command

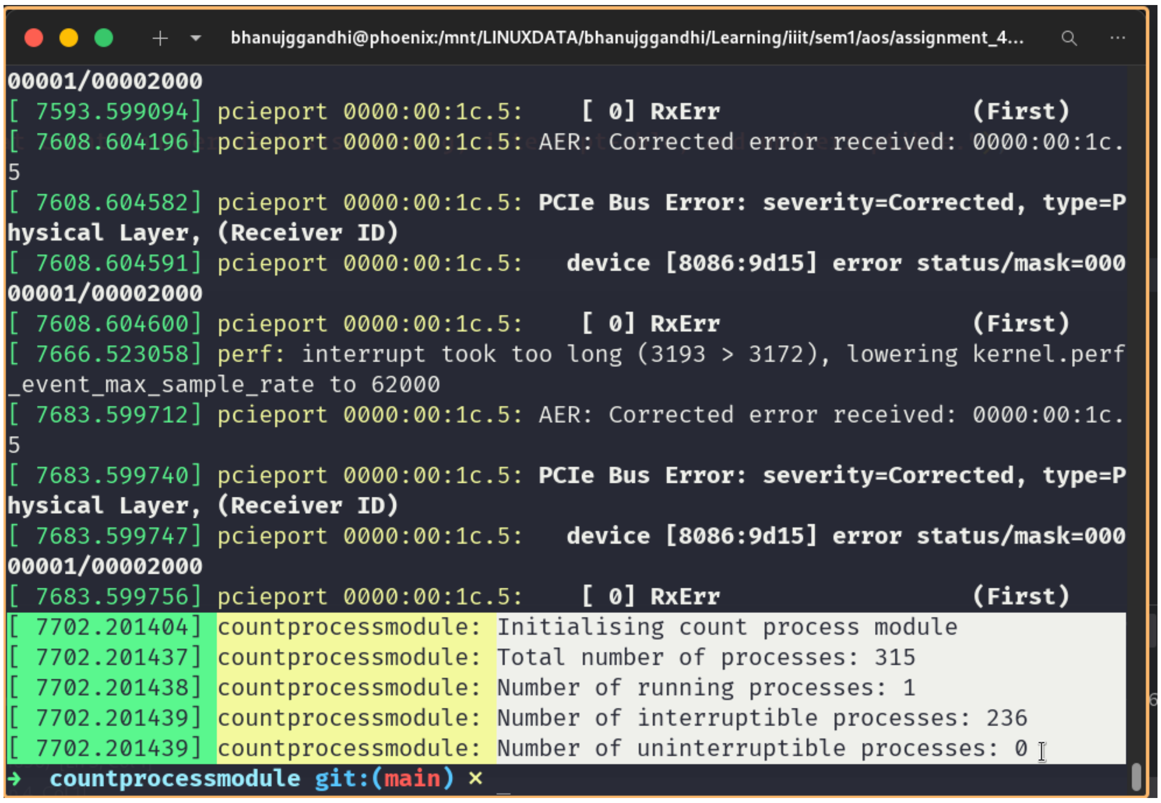
   ```
   $ sudo lsmod
   ```


bhanujggandhi@phoenix:/mnt/LINUXDATA/bhanujggandhi/Learning/iiit/sem1/aos/assignment_4...

```
→ countprocessmodule git:(main) ✗ sudo lsmod
Module                  Size  Used by
countprocessmodule     16384  0
isofs                  61440  0
uas                    32768  0
usb_storage            81920  1 uas
tls                   126976  0
uinput                 20480  0
rfcomm                 94208  4
```

4. Check if the messages are printed in the system log using `dmesg` utiltity.

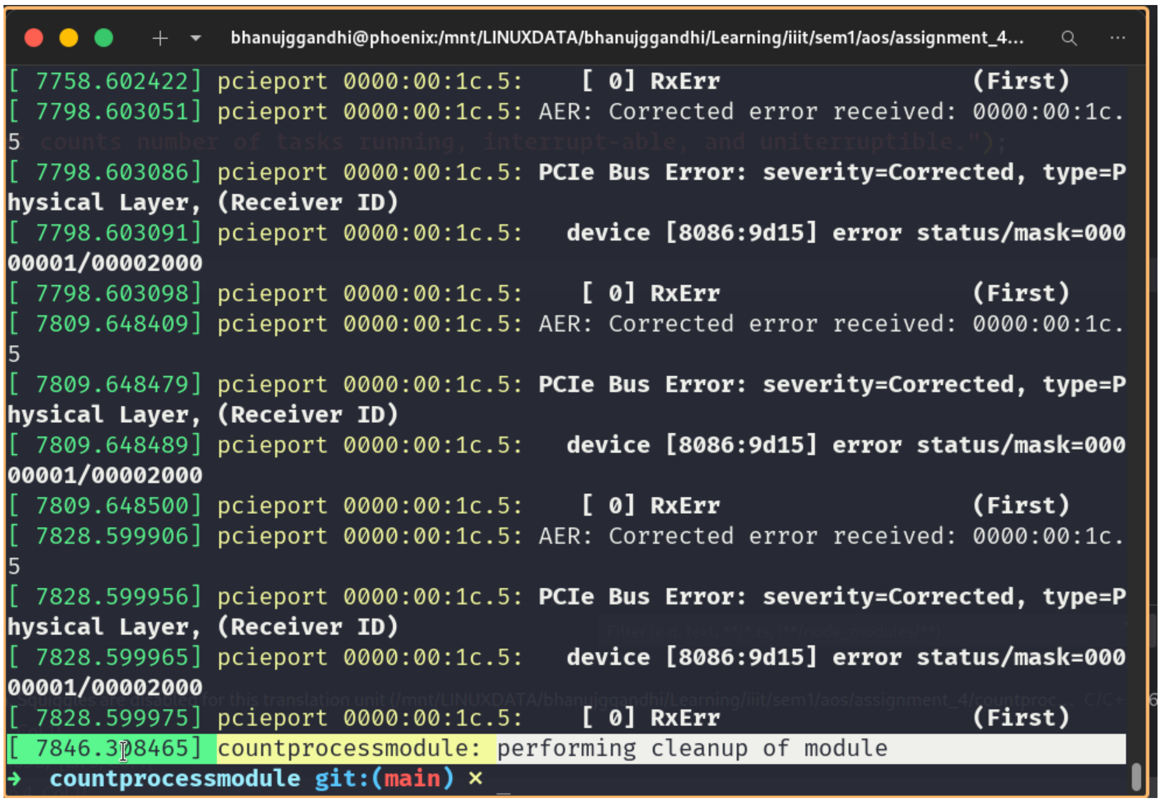
```
$ sudo dmesg
```


bhanujggandhi@phoenix:/mnt/LINUXDATA/bhanujggandhi/Learning/iiit/sem1/aos/assignment_4...

```
00001/00002000
[ 7593.599094] pcieport 0000:00:1c.5:    [ 0] RxErr                  (First)
[ 7608.604196] pcieport 0000:00:1c.5: AER: Corrected error received: 0000:00:1c.
5
[ 7608.604582] pcieport 0000:00:1c.5: PCIe Bus Error: severity=Corrected, type=P
hysical Layer, (Receiver ID)
[ 7608.604591] pcieport 0000:00:1c.5:   device [8086:9d15] error status/mask=000
00001/00002000
[ 7608.604600] pcieport 0000:00:1c.5:    [ 0] RxErr                  (First)
[ 7666.523058] perf: interrupt took too long (3193 > 3172), lowering kernel.perf
_event_max_sample_rate to 62000
[ 7683.599712] pcieport 0000:00:1c.5: AER: Corrected error received: 0000:00:1c.
5
[ 7683.599740] pcieport 0000:00:1c.5: PCIe Bus Error: severity=Corrected, type=P
hysical Layer, (Receiver ID)
[ 7683.599747] pcieport 0000:00:1c.5:   device [8086:9d15] error status/mask=000
00001/00002000
[ 7683.599756] pcieport 0000:00:1c.5:    [ 0] RxErr                  (First)
[ 7702.201404] countprocessmodule: Initialising count process module
[ 7702.201437] countprocessmodule: Total number of processes: 315
[ 7702.201438] countprocessmodule: Number of running processes: 1
[ 7702.201439] countprocessmodule: Number of interruptible processes: 236
[ 7702.201439] countprocessmodule: Number of uninterruptible processes: 0
→ countprocessmodule git:(main) ✗ _
```

5. Remove the module using `rmmod` utility.

```
$ sudo rmmod countprocessmodule
```



bhanujggandhi@phoenix:/mnt/LINUXDATA/bhanujggandhi/Learning/iiit/sem1/aos/assignment_4...

```
[ 7758.602422] pcieport 0000:00:1c.5:    [ 0] RxErr                  (First)
[ 7798.603051] pcieport 0000:00:1c.5: AER: Corrected error received: 0000:00:1c.
5
[ 7798.603086] pcieport 0000:00:1c.5: PCIe Bus Error: severity=Corrected, type=P
hysical Layer, (Receiver ID)
[ 7798.603091] pcieport 0000:00:1c.5:   device [8086:9d15] error status/mask=000
00001/00002000
[ 7798.603098] pcieport 0000:00:1c.5:    [ 0] RxErr                  (First)
[ 7809.648409] pcieport 0000:00:1c.5: AER: Corrected error received: 0000:00:1c.
5
[ 7809.648479] pcieport 0000:00:1c.5: PCIe Bus Error: severity=Corrected, type=P
hysical Layer, (Receiver ID)
[ 7809.648489] pcieport 0000:00:1c.5:   device [8086:9d15] error status/mask=000
00001/00002000
[ 7809.648500] pcieport 0000:00:1c.5:    [ 0] RxErr                  (First)
[ 7828.599906] pcieport 0000:00:1c.5: AER: Corrected error received: 0000:00:1c.
5
[ 7828.599956] pcieport 0000:00:1c.5: PCIe Bus Error: severity=Corrected, type=P
hysical Layer, (Receiver ID)
[ 7828.599965] pcieport 0000:00:1c.5:   device [8086:9d15] error status/mask=000
00001/00002000
[ 7828.599975] pcieport 0000:00:1c.5:    [ 0] RxErr                  (First)
[ 7846.308465] countprocessmodule: performing cleanup of module
➜  countprocessmodule git:(main) ✗ _
```