# Embedded Systems Programming
# Assignment 4.2
Linux Kernel Module

Steinarr Hrafn Höskuldsson

October 6, 2022

## Part 1

The kernel headers were installed without issues. The `hello` kernel example from L4.4 was used and the Makefile from L4.3 was used to compile it.

```
pi@raspberrypi:~/EmbeddedAssignments/Assignment4_2LinuxKernelModule $ sudo
    modinfo hello.ko
filename:       /home/pi/EmbeddedAssignments/Assignment4_2LinuxKernelModule/hello
    .ko
version:        0.1
description:    A simple Linux LKM that accepts characters (bytes) from the user.
author:         Steinarr Hrafn
license:        GPL
srcversion:     61AB8CCEEFA7BB47532F8F3
depends:
name:           hello
vermagic:       5.15.61-v7+ SMP mod_unload modversions ARMv7 p2v8
parm:           name:The name to display in /var/log/kern.log (charp)
```

Listing 1: The output of modinfo

The kernel module was loaded and unloaded while the kernel output log was being monitored.

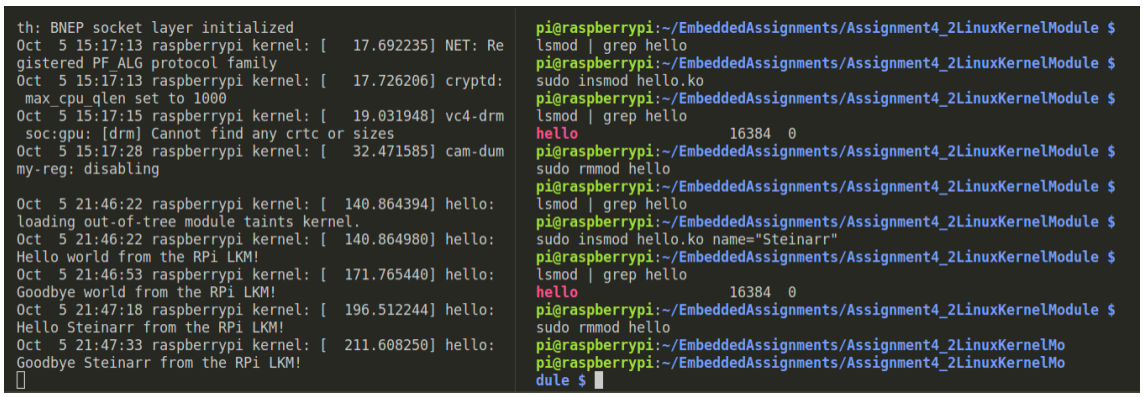


Figure 1: Screenshot of the terminals used to test the loading and unloading of the hello kernel module.

# Part 2

The `mydev.c`example given in the assignment was used, it already has the sysfs function `mydev_write()` implemented. `mydev.ko` was compiled with the same Makefile as before and inserted into the kernel. The device node was then created and tested before cleaning up the kernel module and device node.


Figure 2: Screenshot of the terminals used to test the loading and unloading of the mydev device node

# Part 3

A simple application was written that writes an increasing number of characters to `/dev/mydev1`.

```c
#include<fcntl.h>
#include<unistd.h>
#include <stdio.h>

int main(){
    int fd = open("/dev/mydev1", O_RDWR);
    int count = 0;
    int max_count = 25;
    char buffer[30] = "AAAAAAAAAAAAAAAAAAAAAAAAAAAA";
    while (1){
        count = (count % max_count) + 1;
        write(fd, buffer, count);
        sleep(1);
    }
}
```

Listing 2: A simple program that writes to the Device Node

Running the program while monitoring the Kernel Log resulted in the following being observed in the Kernel Log:

```
[caption={Output of the Kernel Log when use_mydev is run.}]
Oct  6 23:26:25 raspberrypi kernel: [ 2548.997027] mydev:
Device has been opened 4 time(s)
Oct  6 23:26:25 raspberrypi kernel: [ 2548.997070] mydev
write: accepting 1 bytes from the user
Oct  6 23:26:26 raspberrypi kernel: [ 2549.997275] mydev
write: accepting 2 bytes from the user
Oct  6 23:26:27 raspberrypi kernel: [ 2550.997449] mydev
write: accepting 3 bytes from the user
Oct  6 23:26:28 raspberrypi kernel: [ 2551.997596] mydev
write: accepting 4 bytes from the user
Oct  6 23:26:29 raspberrypi kernel: [ 2552.997778] mydev
write: accepting 5 bytes from the user
Oct  6 23:26:29 raspberrypi kernel: [ 2553.118693] mydev:
Device successfully closed
```