

Embedded Systems Programming

Assignment 4.2

Linux Kernel Module

Steinarr Hrafn Höskuldsson

October 6, 2022

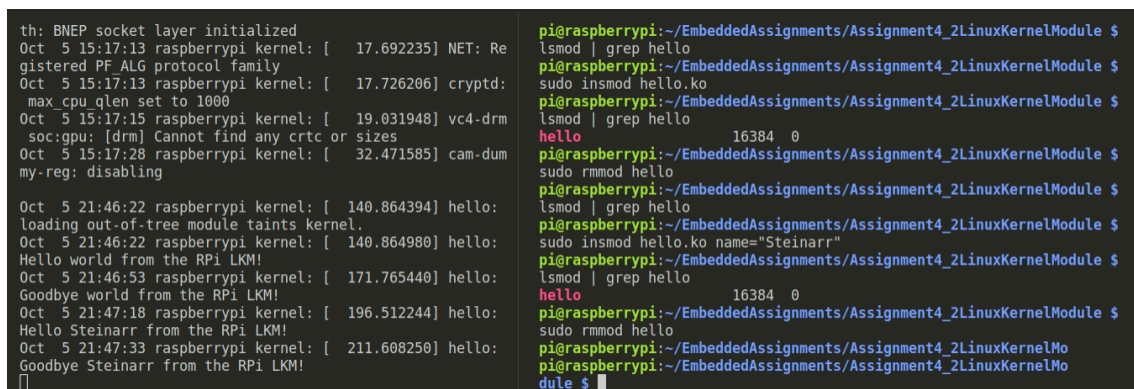
Part 1

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:        61AB8CCEFA7BB47532F8F3
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.



The screenshot shows two terminal windows. The left window displays kernel logs with timestamps and messages from the RPi LKM, including 'Hello world from the RPi LKM!', 'Goodbye world from the RPi LKM!', 'Hello Steinarr from the RPi LKM!', and 'Goodbye Steinarr from the RPi LKM!'. The right window shows the user performing several commands: `lsmod | grep hello`, `sudo insmod hello.ko`, `lsmod | grep hello`, `sudo rmmod hello`, `lsmod | grep hello`, `sudo insmod hello.ko name="Steinarr"`, `lsmod | grep hello`, and `sudo rmmod hello`. The output of these commands shows the module being loaded and then unloaded, with the name 'Steinarr' appearing in the logs.

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.

```

Oct  5 23:08:42 raspberrypi kernel: [ 5080.312049] mydev: Hello world from the RPi LKM!
Oct  5 23:08:42 raspberrypi kernel: [ 5080.312090] mydev: Device registered correctly with major number 238
Oct  5 23:09:25 raspberrypi kernel: [ 5123.779546] mydev: Device has been opened 1 time(s)
Oct  5 23:09:25 raspberrypi kernel: [ 5123.828729] mydev write: accepting 16 bytes from the user
Oct  5 23:09:25 raspberrypi kernel: [ 5123.835069] mydev: Device successfully closed
Oct  5 23:10:05 raspberrypi kernel: [ 5164.263014] mydev: Goodbye world from the RPi LKM!

pi@raspberrypi:~/EmbeddedAssignments/Assignment4_2LinuxKernelModule
$ sudo insmod mydev.ko
pi@raspberrypi:~/EmbeddedAssignments/Assignment4_2LinuxKernelModule
$ sudo mknod /dev/mydev1 c 238 0
pi@raspberrypi:~/EmbeddedAssignments/Assignment4_2LinuxKernelModule
$ sudo chmod a+w /dev/mydev1
pi@raspberrypi:~/EmbeddedAssignments/Assignment4_2LinuxKernelModule
$ sudo echo "what's up, doc?" > /dev/mydev1
pi@raspberrypi:~/EmbeddedAssignments/Assignment4_2LinuxKernelModule
$ sudo rm /dev/mydev1
pi@raspberrypi:~/EmbeddedAssignments/Assignment4_2LinuxKernelModule
$ sudo rmmod mydev
pi@raspberrypi:~/EmbeddedAssignments/Assignment4_2LinuxKernelModule
$

```

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`.

```

#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

```