Rishabh Tewari

2020, December 08

CS 4352: Operating Systems Final Project

Project Description

In this final project we were supposed to implement a simple ramdisk module that can be formatted with `mkfs.ext2`, and can be mounted and unmounted.

Requirements for the project:

- Print out my name
- Implement a ramdisk as a kernel module.
- Allow an ext2 filesystem to be installed onto the ramdisk.
- Allow the filesystem to be mounted and unmounted with “mount” and “unmount”.

Deliverables:

- Code for the project
- Makefile
- Project Report

Instructions to run the project

1. Compile the program by running `make`
2. After make is successfully run, run `make test`
3. This might ask you for the password since it’s using superuser privileges
4. Check dmesg output

Description of my attempts at the project

My attempt at the project partially meets the requirements needed for the project.

The Makefile has instructions to compile and create the modules correctly. My `Makefile` also has a test component that you can test after running `make`, by running `make test`.

My code is in available in the `CS4352Adv.c` file.

Currently here's what my program does:

- Correctly compile the file
- Add information about the module: name, description, etc.
- Print out my name, and other information about the module when inserted
- Print out if there's any errors in loading the module
- Register a Block device
- Print out a message when removing the module
- Unregister the block device when `cleanup_module()` is called

Code for the project:

CS4352Adv.c

```
/*
* Rishabh Tewari
* CS 4352: Final Project
*
* */
#define BLKDEV_NAME                  "os_block_dev"
#define BLOCK_MAJOR                   240

#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/fs.h>

MODULE_AUTHOR("Rishabh Tewari");
MODULE_DESCRIPTION("CS 4352 Final project");
MODULE_VERSION("0.02");
MODULE_LICENSE("GPL");

/*Function called when the module is inserted into the kernel*/
int init_module()
{
    printk(KERN_INFO "This program was written by\nRishabh Tewari\nCS 4352 Final
Project Module Loaded. The more adv version");
    int status;
    status = register_blkdev(BLOCK_MAJOR, BLKDEV_NAME);

    printk(KERN_INFO "CS 4352 Block device REGISTERED successfully - Rishabh Tewari");

    if(status < 0)
    {
            printk(KERN_ERR "unable to register mybdev block device\n");
            return -EBUSY;
    }
```

```
    return 0; //0 shows the module was loaded in successfully
}

/*Function called when the module is removed*/
void cleanup_module()
{
    printk(KERN_INFO "CS 4352 Final Project Module Removed\n");
    unregister_blkdev(BLOCK_MAJOR, BLKDEV_NAME);
}
```

Makefile

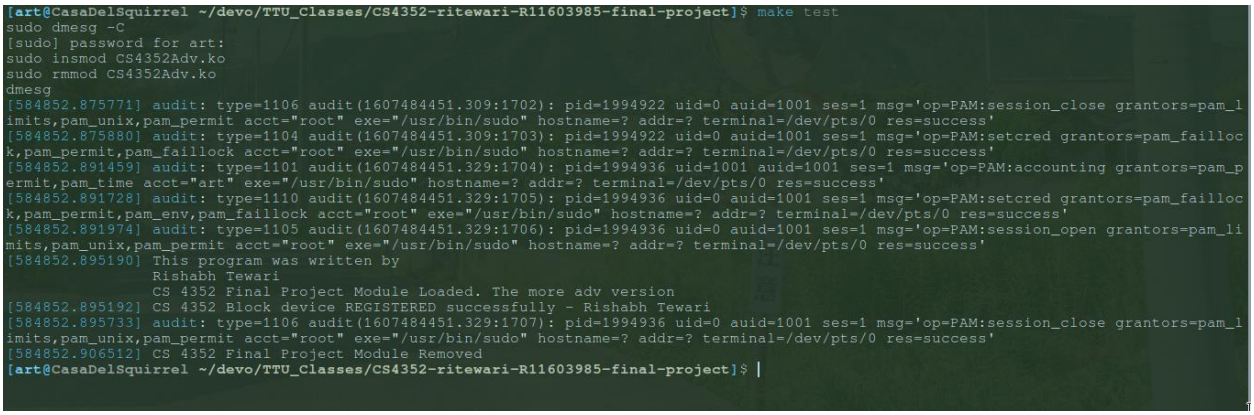
```
obj-m += CS4352Adv.o

all:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean

test:
    sudo dmesg -C
    sudo insmod CS4352Adv.ko
    sudo rmmod CS4352Adv.ko
    dmesg
```

The output:



Resources Used

https://tldp.org/LDP/lkmpg/2.6/html/lkmpg.html

http://www.cs.uni.edu/~diesburg/courses/cop4610_fall10/week06/week6.pdf

https://lwn.net/Kernel/LDD3/

https://opensourceforu.com/2012/02/device-drivers-disk-on-ram-block-drivers/

https://lwn.net/Articles/58719/

https://blog.sourcerer.io/writing-a-simple-linux-kernel-module-d9dc3762c234?gi=8dc1852445e5

https://tldp.org/LDP/lkmpg/2.6/html/lkmpg.html

http://derekmolloy.ie/writing-a-linux-kernel-module-part-1-introduction/