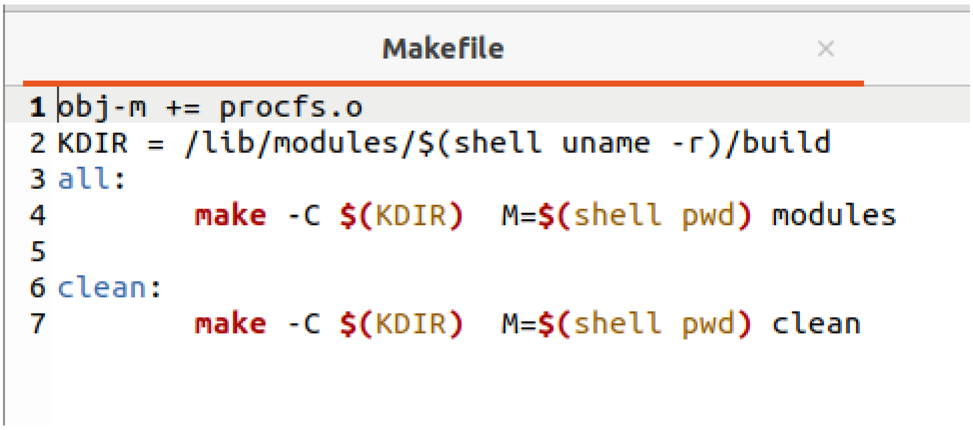
DEVICE DRIVERS LAB – 5

DONE BY
A S V DHANUSH
CS20B1057

Makefile



```
obj-m += procfs.o
KDIR = /lib/modules/$(shell uname -r)/build
all:
	make -C $(KDIR)  M=$(shell pwd) modules

clean:
	make -C $(KDIR)  M=$(shell pwd) clean
```

procfs.c

(NOTE : Ignore the warnings)

```
#include<linux/kernel.h>
#include<linux/init.h>
#include<linux/module.h>
#include<linux/kdev_t.h>
#include<linux/fs.h>
#include<linux/cdev.h>
#include<linux/device.h>
#include<linux/slab.h>
#include<linux/uaccess.h>
#include<linux/ioctl.h>
#include<linux/proc_fs.h>

#define mem_size 1024  // Macro for memory size
char chr_array[40]="Welcome to Device Drivers Lab\n";
static int length = 0;
// Define the ioctl code
#define WR_DATA _IOW('a','a',int32_t*)
#define RD_DATA _IOR('a','b',int32_t*)

int32_t val=0;

dev_t dev = 0;
static struct class *dev_class;
static struct cdev my_cdev;

uint8_t *kernel_buffer;

static int __init chr_driver_init(void);
static void __exit chr_driver_exit(void);

static int my_open(struct inode *inode, struct file *file);
static int my_release(struct inode *inode, struct file *file);
static ssize_t my_read(struct file *filp, char __user *buf, size_t len, loff_t *off);
static ssize_t my_write(struct file *filp, const char *buf, size_t len, loff_t *off);
static long my_ioctl(struct file *file,unsigned int cmd,unsigned long arg);

static int open_proc(struct inode *inode, struct file *file);
static int release_proc(struct inode *inode, struct file *file);
static ssize_t read_proc(struct file *filp, char __user *buf, size_t len, loff_t *off);
static ssize_t write_proc(struct file *filp, const char *buf, size_t len, loff_t *off);

static struct file_operations fops=
{
        .owner          =       THIS_MODULE,
        .read           =       my_read,
        .write          =       my_write,
        .open           =       my_open,
        .unlocked_ioctl =my_ioctl,
        .release  =     my_release,
};
static struct proc_ops proc_fops = {
    .proc_open = open_proc,
    .proc_read = read_proc,
    .proc_write = write_proc,
    .proc_release = release_proc
};
// procfs functions
static int open_proc(struct inode *inode, struct file *file){
        printk(KERN_INFO" Procfs File is opened\n");
        return 0;
}
static ssize_t read_proc(struct file *filp, char __user *buf, size_t len, loff_t *off){
        printk(KERN_INFO "Procfs file reading.....\n");
        if(length)
                length=0;
        else{
                length=1;
                return 0;
        }
        copy_to_user(buf,chr_array,40);
        return len;
}
static ssize_t write_proc(struct file *filp, const char *buf, size_t len, loff_t *off){
        printk(KERN_INFO"Procfs file writing.......\n");
        copy_from_user(chr_array,buf,40);
        return len;
}
static int release_proc(struct inode *inode, struct file *file){
        printk(KERN_INFO" Procfs File is opened\n");
        return 0;
}


// char_driver device functions
static int my_open(struct inode *inode, struct file *file)
{
        // Creating physical Memory
        if((kernel_buffer = kmalloc(mem_size, GFP_KERNEL))==0)
        {
                printk(KERN_INFO"Can NOT allocate the memory to kernel ...\n");
                return -1;
        }
        printk(KERN_INFO "Device File Opened...\n");
        return 0;
}
```

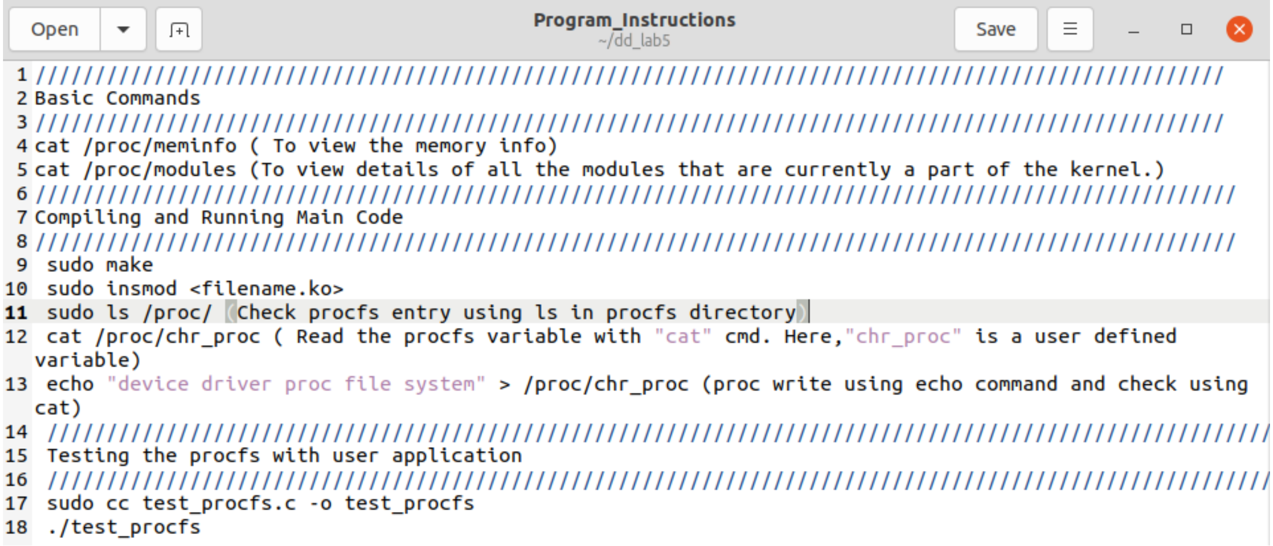
```
static int my_release(struct inode *inode, struct file *file)
{
        kfree(kernel_buffer);
        printk(KERN_INFO"Device File Closed...\n");
        return 0;
}

static ssize_t my_read(struct file *filp, char __user *buf, size_t  len, loff_t *loff)
{
        copy_to_user(buf, kernel_buffer,mem_size);
        printk(KERN_INFO "Data Read: DONE....\n");
        return mem_size;
}

static ssize_t my_write(struct file *filp, const char __user *buf, size_t len, loff_t  *loff)
{
        copy_from_user(kernel_buffer, buf, len);
        printk(KERN_INFO "Data is written Successfully...\n");
        return len;
}

//IOCTL functions

static long my_ioctl(struct file *file,unsigned int cmd, unsigned long arg)
{
        switch(cmd){
                case WR_DATA:
                        copy_from_user(&val,(int32_t*)arg,sizeof(val));
                        printk(KERN_INFO " val=%d\n",val);
                        break;
                case RD_DATA:
                        copy_to_user((int32_t*)arg,&val,sizeof(val));
                        break;
        }
return 0;
}


static int __init chr_driver_init(void)
{
        // Allocating Major Number Dynamically
        if((alloc_chrdev_region(&dev, 0, 1, "my_Dev")) <0)
        {
                printk(KERN_INFO"Can NOT allocate the Major Number..\n");
                return -1;
        }
        printk(KERN_INFO"Major = %d and Minor = %d..\n", MAJOR(dev),MINOR(dev));

        // Creating  cdev structure

        cdev_init(&my_cdev, &fops);

        // Adding Character device to the system
        if ((cdev_add(&my_cdev, dev, 1)) < 0)
        {
                printk(KERN_INFO"Can NOT add the device to the system...\n");
                goto r_class;
        }

        // Creating Struct Class
        if((dev_class = class_create(THIS_MODULE,"my_class")) == NULL)
        {
                printk(KERN_INFO"Unable to create the struct class ...\n");
                goto r_class;    // Un recognize the character device
        }

        // Creating Device
        if((device_create(dev_class, NULL, dev, NULL,"my_device"))== NULL)
        {
                printk(KERN_INFO"Can NOT create the device...\n");
                goto r_device;
        }

        /* Create  the Proc entry*/
        proc_create("qwerty",0666,NULL,&proc_fops);

        printk(KERN_INFO"Device Driver is inserted properly DONE...\n");
        return 0;


r_device:
        class_destroy(dev_class);

r_class:
        unregister_chrdev_region(dev,1);
        return -1;
}

void __exit chr_driver_exit(void)
{
        device_destroy(dev_class, dev);
        class_destroy(dev_class);
        cdev_del(&my_cdev);
        unregister_chrdev_region(dev,1);
        printk(KERN_INFO"Device Driver is Removed Successfully...\n");
}

module_init(chr_driver_init);
module_exit(chr_driver_exit);

MODULE_LICENSE("GPL");
MODULE_AUTHOR("IIITDM KANCHEEPURAM");
MODULE_DESCRIPTION("The Character Device Driver");
```

test_procfs.c

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<fcntl.h>
#include<unistd.h>

int8_t write_buf[1024];
int8_t read_buf[1024];

int main()
{

        int fd;
        char option;

        printf("Welcome to the Procfs DEMO..\n");
        fd = open("/proc/chr_proc",O_RDWR);
        if(fd < 0)
```

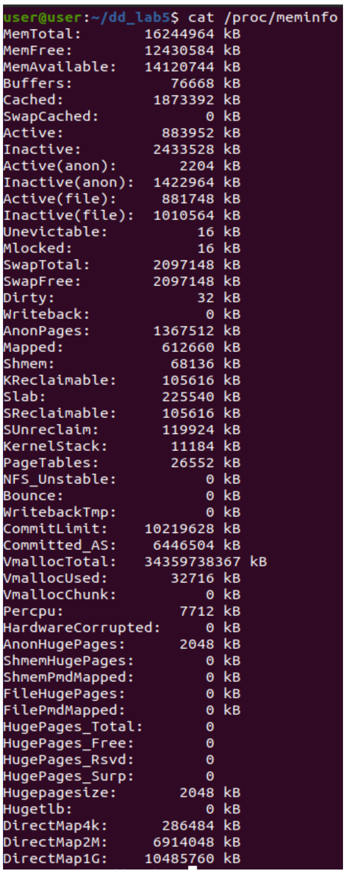
```
	{
		printf("Unable to Open the Device File...\n");
		return 0;
	}

	while(1)
	{
		printf("************** Please Enter Your Options**************\n");
		printf("                    1. Write                        \n");
		printf("                    2. Read                         \n");
		printf("                    3. Exit                              \n");
		scanf("%c", &option);
		printf(" Your Options are = %c \n", option);

		switch(option)
		{
			case '1':
				printf("Enter the String to Write in to the Driver...\n");
				scanf(" %[^\t\n]s", write_buf);
				printf("Data Writtern...\n");
				write(fd, write_buf, strlen(write_buf)+1);
				printf("Write Operation Completed ... DONE...\n");
				break;
			case '2':
				printf("Data is Reading....\n");
				read(fd, read_buf, 1024);
				printf("Done....\n\n");
				printf("Data = %s\n\n", read_buf);
				break;
			case '3':
				close(fd);
				exit(1);
				break;
			default:
				printf("Enter the Valid Option = %c\n",option);
				break;
			}
	}
	close(fd);
}
```

Program_Instructions



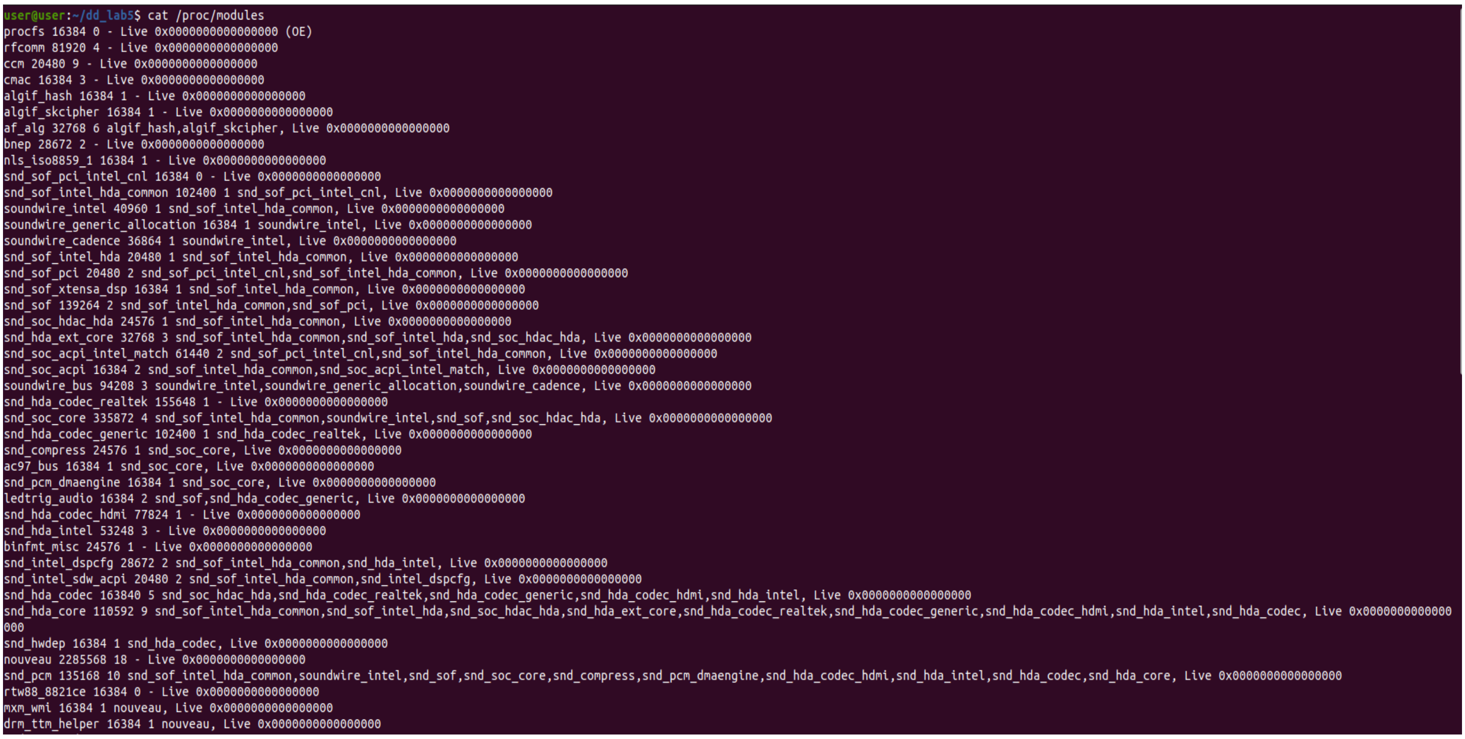
```
////////////////////////////////////////////////////////////////////////////////////////////////////
Basic Commands
////////////////////////////////////////////////////////////////////////////////////////////////////
cat /proc/meminfo ( To view the memory info)
cat /proc/modules (To view details of all the modules that are currently a part of the kernel.)
/////////////////////////////////////////////////////////////////////////////////////////////////////
Compiling and Running Main Code
/////////////////////////////////////////////////////////////////////////////////////////////////////
sudo make
sudo insmod <filename.ko>
sudo ls /proc/ (Check procfs entry using ls in procfs directory)
cat /proc/chr_proc ( Read the procfs variable with "cat" cmd. Here,"chr_proc" is a user defined
variable)
echo "device driver proc file system" > /proc/chr_proc (proc write using echo command and check using
cat)
//////////////////////////////////////////////////////////////////////////////////////////////////////
Testing the procfs with user application
//////////////////////////////////////////////////////////////////////////////////////////////////////
sudo cc test_procfs.c -o test_procfs
./test_procfs
```

cat /proc/meminfo (To view the memory info)



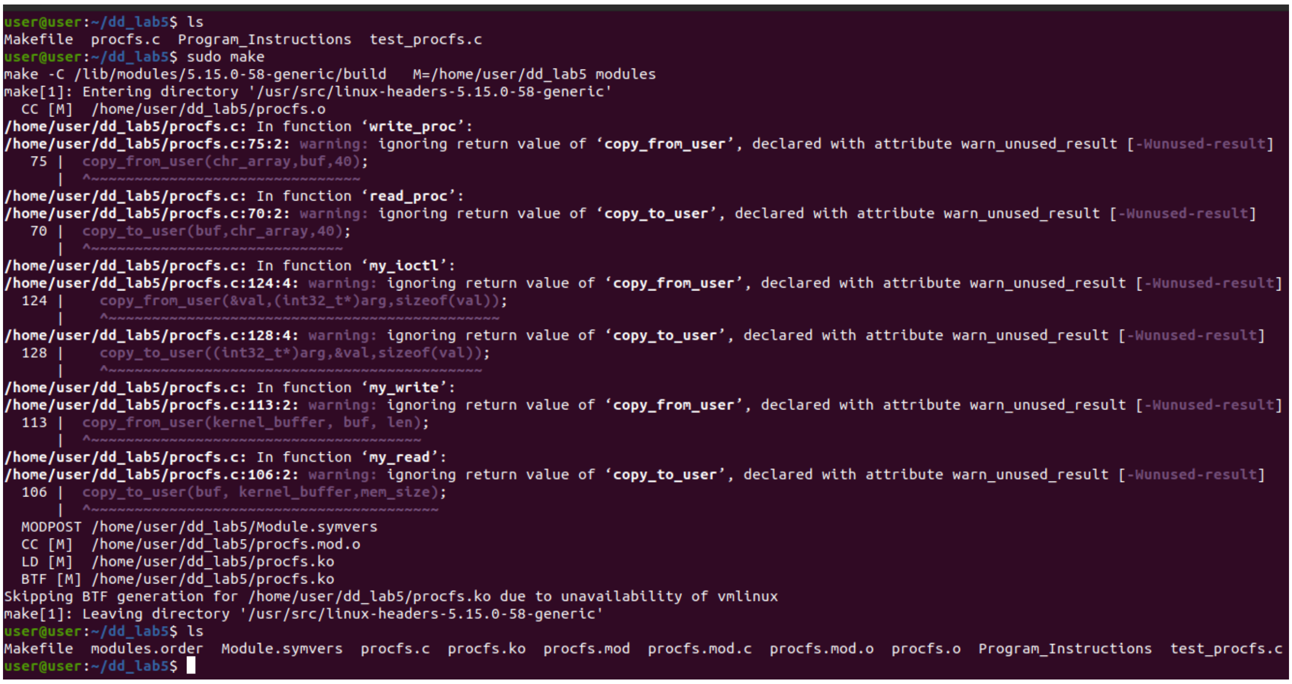
```
user@user:~/dd_lab5$ cat /proc/meminfo
MemTotal:       16244964 kB
MemFree:        12430584 kB
MemAvailable:   14120744 kB
Buffers:           76668 kB
Cached:          1873392 kB
SwapCached:            0 kB
Active:           883952 kB
Inactive:        2433528 kB
Active(anon):       2204 kB
Inactive(anon):  1422964 kB
Active(file):     881748 kB
Inactive(file):  1010564 kB
Unevictable:          16 kB
Mlocked:              16 kB
SwapTotal:       2097148 kB
SwapFree:        2097148 kB
Dirty:                32 kB
Writeback:             0 kB
AnonPages:       1367512 kB
Mapped:           612660 kB
Shmem:             68136 kB
KReclaimable:     105616 kB
Slab:             225540 kB
SReclaimable:     105616 kB
SUnreclaim:       119924 kB
KernelStack:       11184 kB
PageTables:        26552 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:    10219628 kB
Committed_AS:    6446504 kB
VmallocTotal:   34359738367 kB
VmallocUsed:       32716 kB
VmallocChunk:          0 kB
Percpu:             7712 kB
HardwareCorrupted:     0 kB
AnonHugePages:      2048 kB
ShmemHugePages:        0 kB
ShmemPmdMapped:        0 kB
FileHugePages:         0 kB
FilePmdMapped:         0 kB
HugePages_Total:       0
HugePages_Free:        0
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
Hugetlb:               0 kB
DirectMap4k:      286484 kB
DirectMap2M:     6914048 kB
DirectMap1G:    10485760 kB
```

cat /proc/modules (To view details of all the modules that are currently a part of the kernel.)



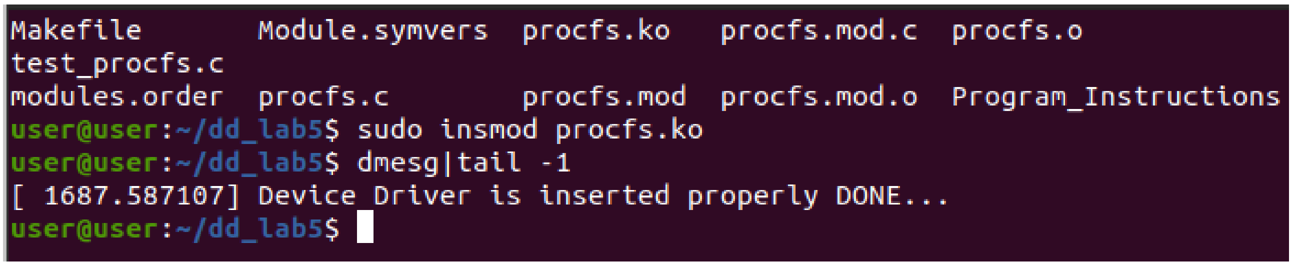
```
user@user:~/dd_lab5$ cat /proc/modules
procfs 16384 0 - Live 0x0000000000000000 (OE)
rfcomm 81920 4 - Live 0x0000000000000000
ccm 20480 9 - Live 0x0000000000000000
cmac 16384 3 - Live 0x0000000000000000
algif_hash 16384 1 - Live 0x0000000000000000
algif_skcipher 16384 1 - Live 0x0000000000000000
af_alg 32768 6 algif_hash,algif_skcipher, Live 0x0000000000000000
bnep 28672 2 - Live 0x0000000000000000
nls_iso8859_1 16384 1 - Live 0x0000000000000000
snd_sof_pci_intel_cnl 16384 0 - Live 0x0000000000000000
snd_sof_intel_hda_common 102400 1 snd_sof_pci_intel_cnl, Live 0x0000000000000000
soundwire_intel 40960 1 snd_sof_intel_hda_common, Live 0x0000000000000000
soundwire_generic_allocation 16384 1 soundwire_intel, Live 0x0000000000000000
soundwire_cadence 36864 1 soundwire_intel, Live 0x0000000000000000
snd_sof_intel_hda 20480 1 snd_sof_intel_hda_common, Live 0x0000000000000000
snd_sof_pci 20480 2 snd_sof_pci_intel_cnl,snd_sof_intel_hda_common, Live 0x0000000000000000
snd_sof_xtensa_dsp 16384 1 snd_sof_intel_hda_common, Live 0x0000000000000000
snd_sof 139264 2 snd_sof_intel_hda_common,snd_sof_pci, Live 0x0000000000000000
snd_soc_hdac_hda 24576 1 snd_sof_intel_hda_common, Live 0x0000000000000000
snd_hda_ext_core 32768 3 snd_sof_intel_hda_common,snd_sof_intel_hda,snd_soc_hdac_hda, Live 0x0000000000000000
snd_soc_acpi_intel_match 61440 2 snd_sof_pci_intel_cnl,snd_sof_intel_hda_common, Live 0x0000000000000000
snd_soc_acpi 16384 2 snd_sof_intel_hda_common,snd_soc_acpi_intel_match, Live 0x0000000000000000
soundwire_bus 94208 3 soundwire_intel,soundwire_generic_allocation,soundwire_cadence, Live 0x0000000000000000
snd_hda_codec_realtek 155648 1 - Live 0x0000000000000000
snd_soc_core 335872 4 snd_sof_intel_hda_common,soundwire_intel,snd_sof,snd_soc_hdac_hda, Live 0x0000000000000000
snd_hda_codec_generic 102400 1 snd_hda_codec_realtek, Live 0x0000000000000000
snd_compress 24576 1 snd_soc_core, Live 0x0000000000000000
ac97_bus 16384 1 snd_soc_core, Live 0x0000000000000000
snd_pcm_dmaengine 16384 1 snd_soc_core, Live 0x0000000000000000
ledtrig_audio 16384 2 snd_sof,snd_hda_codec_generic, Live 0x0000000000000000
snd_hda_codec_hdmi 77824 1 - Live 0x0000000000000000
snd_hda_intel 53248 3 - Live 0x0000000000000000
binfmt_misc 24576 1 - Live 0x0000000000000000
snd_intel_dspcfg 28672 2 snd_sof_intel_hda_common,snd_hda_intel, Live 0x0000000000000000
snd_intel_sdw_acpi 20480 2 snd_sof_intel_hda_common,snd_intel_dspcfg, Live 0x0000000000000000
snd_hda_codec 163840 5 snd_soc_hdac_hda,snd_hda_codec_realtek,snd_hda_codec_generic,snd_hda_codec_hdmi,snd_hda_intel, Live 0x0000000000000000
snd_hda_core 110592 9 snd_sof_intel_hda_common,snd_sof_intel_hda,snd_soc_hdac_hda,snd_hda_ext_core,snd_hda_codec_realtek,snd_hda_codec_generic,snd_hda_codec_hdmi,snd_hda_intel,snd_hda_codec, Live 0x0000000000000
000
snd_hwdep 16384 1 snd_hda_codec, Live 0x0000000000000000
nouveau 2285568 18 - Live 0x0000000000000000
snd_pcm 135168 10 snd_sof_intel_hda_common,soundwire_intel,snd_sof,snd_soc_core,snd_compress,snd_pcm_dmaengine,snd_hda_codec_hdmi,snd_hda_intel,snd_hda_codec,snd_hda_core, Live 0x0000000000000000
rtw88_8821ce 16384 0 - Live 0x0000000000000000
mxm_wmi 16384 1 nouveau, Live 0x0000000000000000
drm_ttm_helper 16384 1 nouveau, Live 0x0000000000000000
```

Creating the kernel object using make



```
user@user:~/dd_lab5$ ls
Makefile  procfs.c  Program_Instructions  test_procfs.c
user@user:~/dd_lab5$ sudo make
make -C /lib/modules/5.15.0-58-generic/build   M=/home/user/dd_lab5 modules
make[1]: Entering directory '/usr/src/linux-headers-5.15.0-58-generic'
  CC [M]  /home/user/dd_lab5/procfs.o
/home/user/dd_lab5/procfs.c: In function ‘write_proc’:
/home/user/dd_lab5/procfs.c:75:2: warning: ignoring return value of ‘copy_from_user’, declared with attribute warn_unused_result [-Wunused-result]
   75 |  copy_from_user(chr_array,buf,40);
      |  ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/home/user/dd_lab5/procfs.c: In function ‘read_proc’:
/home/user/dd_lab5/procfs.c:70:2: warning: ignoring return value of ‘copy_to_user’, declared with attribute warn_unused_result [-Wunused-result]
   70 |  copy_to_user(buf,chr_array,40);
      |  ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/home/user/dd_lab5/procfs.c: In function ‘my_ioctl’:
/home/user/dd_lab5/procfs.c:124:4: warning: ignoring return value of ‘copy_from_user’, declared with attribute warn_unused_result [-Wunused-result]
  124 |    copy_from_user(&val,(int32_t*)arg,sizeof(val));
      |    ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/home/user/dd_lab5/procfs.c:128:4: warning: ignoring return value of ‘copy_to_user’, declared with attribute warn_unused_result [-Wunused-result]
  128 |    copy_to_user((int32_t*)arg,&val,sizeof(val));
      |    ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/home/user/dd_lab5/procfs.c: In function ‘my_write’:
/home/user/dd_lab5/procfs.c:113:2: warning: ignoring return value of ‘copy_from_user’, declared with attribute warn_unused_result [-Wunused-result]
  113 |  copy_from_user(kernel_buffer, buf, len);
      |  ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/home/user/dd_lab5/procfs.c: In function ‘my_read’:
/home/user/dd_lab5/procfs.c:106:2: warning: ignoring return value of ‘copy_to_user’, declared with attribute warn_unused_result [-Wunused-result]
  106 |  copy_to_user(buf, kernel_buffer,mem_size);
      |  ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  MODPOST /home/user/dd_lab5/Module.symvers
  CC [M]  /home/user/dd_lab5/procfs.mod.o
  LD [M]  /home/user/dd_lab5/procfs.ko
  BTF [M] /home/user/dd_lab5/procfs.ko
Skipping BTF generation for /home/user/dd_lab5/procfs.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-5.15.0-58-generic'
user@user:~/dd_lab5$ ls
Makefile  modules.order  Module.symvers  procfs.c  procfs.ko  procfs.mod  procfs.mod.c  procfs.mod.o  procfs.o  Program_Instructions  test_procfs.c
user@user:~/dd_lab5$ 
```

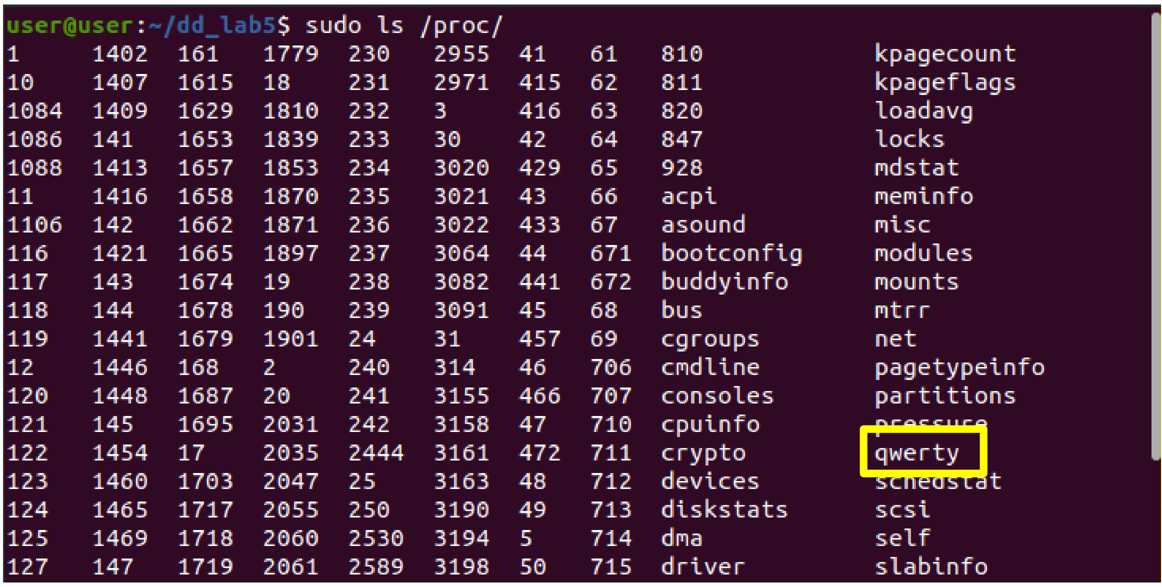
Inserting procfs.ko into the kernel



```
Makefile       Module.symvers  procfs.ko    procfs.mod.c  procfs.o
test_procfs.c
modules.order  procfs.c        procfs.mod   procfs.mod.o  Program_Instructions
user@user:~/dd_lab5$ sudo insmod procfs.ko
user@user:~/dd_lab5$ dmesg|tail -1
[ 1687.587107] Device Driver is inserted properly DONE...
user@user:~/dd_lab5$ 
```

sudo ls /proc/ (Checking if our kernel module name is there or not)

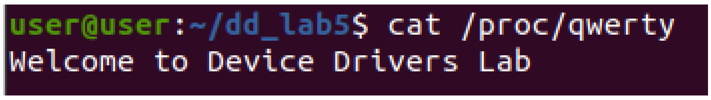
(Note : “chr_proc” has been changed to “qwerty’ as user defined variable)



```
user@user:~/dd_lab5$ sudo ls /proc/
1     1402  161   1779  230   2955  41   61   810           kpagecount
10    1407  1615  18    231   2971  415  62   811           kpageflags
1084  1409  1629  1810  232   3     416  63   820           loadavg
1086  141   1653  1839  233   30    42   64   847           locks
1088  1413  1657  1853  234   3020  429  65   928           mdstat
11    1416  1658  1870  235   3021  43   66   acpi          meminfo
1106  142   1662  1871  236   3022  433  67   asound        misc
116   1421  1665  1897  237   3064  44   671  bootconfig    modules
117   143   1674  19    238   3082  441  672  buddyinfo     mounts
118   144   1678  190   239   3091  45   68   bus           mtrr
119   1441  1679  1901  24    31    457  69   cgroups       net
12    1446  168   2     240   314   46   706  cmdline       pagetypeinfo
120   1448  1687  20    241   3155  466  707  consoles      partitions
121   145   1695  2031  242   3158  47   710  cpuinfo       pressure
122   1454  17    2035  2444  3161  472  711  crypto        qwerty
123   1460  1703  2047  25    3163  48   712  devices       schedstat
124   1465  1717  2055  250   3190  49   713  diskstats     scsi
125   1469  1718  2060  2530  3194  5    714  dma           self
127   147   1719  2061  2589  3198  50   715  driver        slabinfo
```

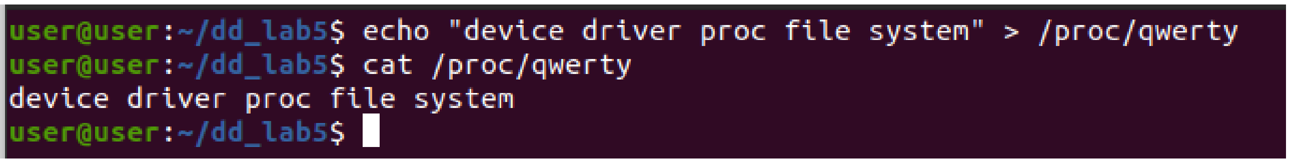
(qwerty is present)

cat /proc/qwerty (To read the text written by user in module file with user defined variable “qwerty”)



```
user@user:~/dd_lab5$ cat /proc/qwerty
Welcome to Device Drivers Lab
```

echo "device driver proc file system" > /proc/chr_proc (proc overwrite using echo command and check using cat)



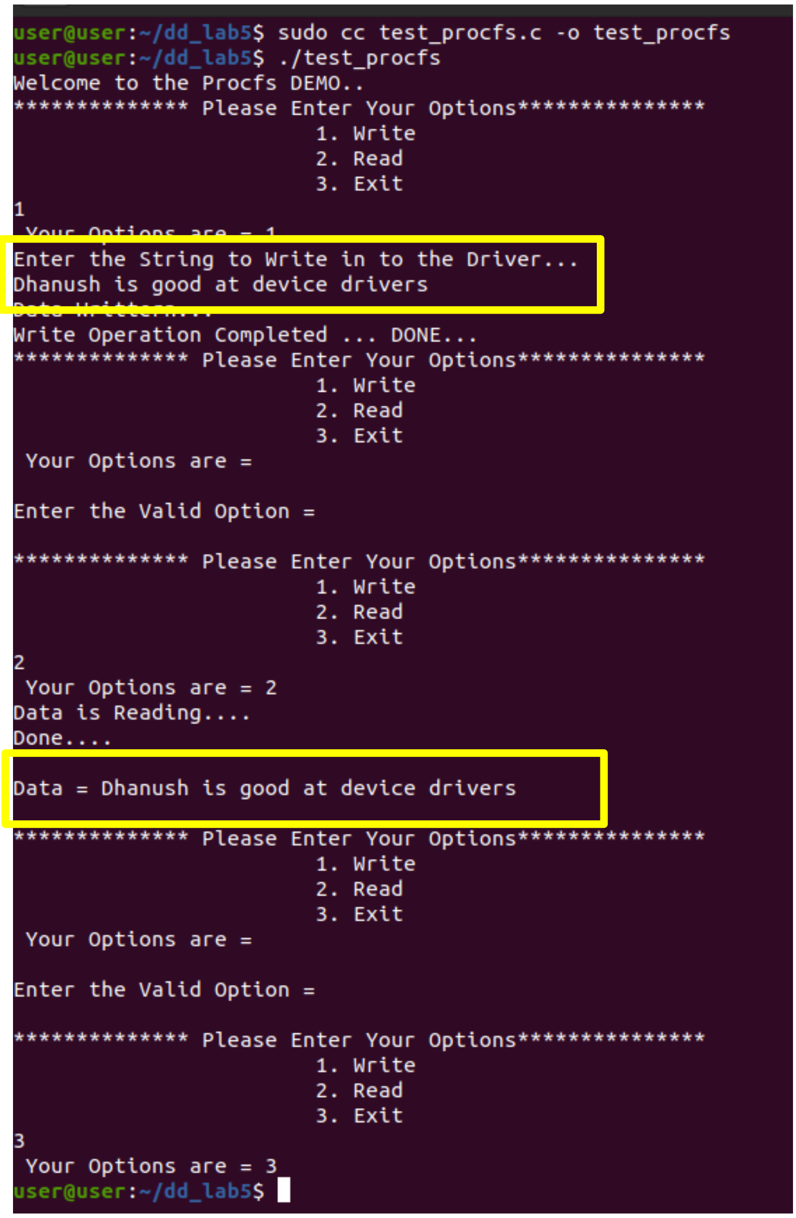
```
user@user:~/dd_lab5$ echo "device driver proc file system" > /proc/qwerty
user@user:~/dd_lab5$ cat /proc/qwerty
device driver proc file system
user@user:~/dd_lab5$ 
```

Note : The text “Welcome to Device Drivers Lab” has been overwritten to “device driver proc file system”
and is been displayed by cat command

Running the test_procfs using

sudo cc test_procfs.c -o test_procfs

./test_procfs



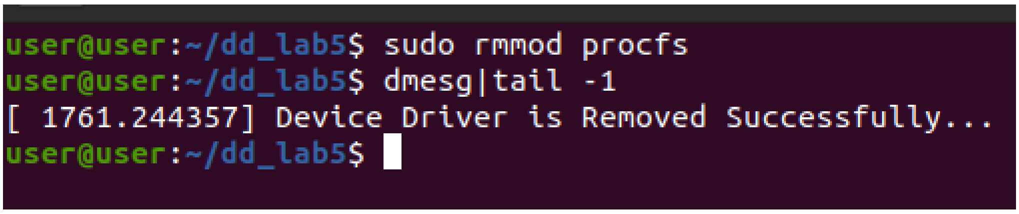
```
user@user:~/dd_lab5$ sudo cc test_procfs.c -o test_procfs
user@user:~/dd_lab5$ ./test_procfs
Welcome to the Procfs DEMO..
************** Please Enter Your Options***************
                        1. Write
                        2. Read
                        3. Exit
1
 Your Options are = 1
Enter the String to Write in to the Driver...
Dhanush is good at device drivers
Data Writtern....
Write Operation Completed ... DONE...
************** Please Enter Your Options***************
                        1. Write
                        2. Read
                        3. Exit
 Your Options are = 

Enter the Valid Option = 

************** Please Enter Your Options***************
                        1. Write
                        2. Read
                        3. Exit
2
 Your Options are = 2
Data is Reading....
Done....

Data = Dhanush is good at device drivers

************** Please Enter Your Options***************
                        1. Write
                        2. Read
                        3. Exit
 Your Options are = 

Enter the Valid Option = 

************** Please Enter Your Options***************
                        1. Write
                        2. Read
                        3. Exit
3
 Your Options are = 3
user@user:~/dd_lab5$ 
```

Data Written

Data Read
And Printed

Exitted

Removing the module from the kernel



```
user@user:~/dd_lab5$ sudo rmmod procfs
user@user:~/dd_lab5$ dmesg|tail -1
[ 1761.244357] Device Driver is Removed Successfully...
user@user:~/dd_lab5$ 
```