CS 746 : Linux Kernel Programming

Assignment 2
Making my own Character Device Driver

Submitted by

Aurobindo Mondal
143050082

Under the guidance of

Prof. Purushottam Kulkarni

Department of Computer Science and Engineering
IIT Bombay

March 16, 2016

Algorithm for my Character Device Driver:

- Register the character device

```
register_chrdev(279, "myCharDev", &myFileOps
```

- Allocate Kernel Memory and initialise it with some header

```
buffer = get_zeroed_page(GFP_KERNEL);
memcpy((char *)buffer, "Hi, I am Kernel!", 16);
```

- Override mmap of the file_operation struct

```
static const struct file_operations myFileOps = {
.owner      = THIS_MODULE,
.mmap       = myMmap,
};
```

- Write your own mmap where you override the vma_ops

```
static int myMmap(struct file *myFile, struct vm_area_struct *myVma)    {
        myVma->vm_ops = &mmap_vm_ops;
        myVma->vm_flags |= VM_RESERVED;
        return 0;
}
struct vm_operations_struct mmap_vm_ops = {
    .fault =    mmap_fault,
};
```

- Change the mmap_fault

```
static int mmap_fault(struct vm_area_struct *vma, struct vm_fault *vmf)  {
    struct page *page;
    page = virt_to_page(buffer);
    get_page(page);
    vmf->page = page;
    return 0;
}
```

Working of mmap_fault :

Whenever the user process tries to write to the virtual memory, a page fault occurs (as the pag is not mapped to physical memory) which is handled by mmap_fault.

So my mmap_fault give the address of the kernel page as the reply to the page fault.

Results and Observations :

Output :

```
$ sudo insmod part2-charDevDrvr.ko
$ sudo ./part2-testerProgram
Hi, I am Kernel!
Hi, I am Kernel!Hi, I am the User

$ sudo rmmod part2_charDevDrvr
[ 1002.592889] Device successfully registered!!!
[ 1002.592908] Hi, I am Kernel!
[ 1041.827902] Hi, I am Kernel!Hi, I am the User
[ 1041.827918] Removed device!!!
[ 1041.827918] Good Bye!!!!
```

Applying Patch :

```
Command: diff -uNr original_kernel modified kernel > patchfile
```