



**TÜRKİYE
CUMHURİYETİ**
**İSTANBUL
ÜNİVERSİTESİ**
**MÜHENDİSLİK
FAKÜLTESİ**
**BİLGİSAYAR
MÜHENDİSLİĞİ**



ÖDEV #2 | 4 İŞLEM YAPABİLEN PROC UYGULAMASI

GRUP ADI: **FSOCIETY**

- | | |
|-------------------------|--------------------------|
| • KADIR KIVANC | 1306130027 - Grup Lideri |
| • HALİL DEMİR | 1306130019 |
| • MUHAMMED FATİH CAMBEK | 1306130030 |
| • OMER ALİ OZBEK | 1306130071 |
| • ADEM ENES ULUTAS | 1306140097 |

ÖDEV 2 | 4 İŞLEM YAPABİLEN PROC UYGULAMASI

1. YÜKLÜ LINUX VE KERNEL VERSİYONU

Kullanılan Linux sürümü “13.04”, Kernel versiyonu ise “3.8.0-19-generic”tir.

2. PROC UYGULAMASI

```
//---- Kullarılan Kütüphaneler ----//
#include <linux/module.h>
#include <linux/init.h>
#include <linux/proc_fs.h>
#include <asm/uaccess.h>
//-----//

//----- Global Değişkenler -----//
static unsigned long val1;
static char op;
static unsigned long val2;
static unsigned long result;
//-----//

/***** Proc Sanal Dosyasından Okuma Fonksiyonu *****/

static int procfile_read(char *buffer, char **start, off_t offset, int length)
{
    int size;
    size = sprintf(buffer, "%ld\n", result);
    *start = buffer + offset;
    size = size - offset;

    if(size>length)
    {
        return length;
    }

    else if (size>0)
    {
        return size;
    }

    else return 0;
}

/*****
```

```

/***** Proc Sanal Dosyasına Yazma Fonksiyonu *****/
static int procfile_write(struct file *file, const char __user *buff, unsigned long len, void *data)
{
    char buf[23]; // vals1(10) + boşluk(1) + işlem(1) + boşluk(1) + vals2(10) = buf(23)
    char vals1[10], vals2[10]; // Maksimum 10 elemanlı sayılar kabul edildi.
    char *endp1, *endp2;

    if (len > sizeof(buf)) { return -EINVAL; }
    if (copy_from_user(buf, buff, len)) { return -EFAULT; } // Data, "user katmanı"ndan katmanından
    "kernel katmanı"na katmanına kopyalandı.

    //----- karakterleri çözümleyerek sayıları ve işlem operatörünü belirler -----//
    int i;
    for(i=0; i<23; i++)
    {
        if(buf[i]==' ')
        {
            i++;
            op = buf[i];
            i++;
            i++;
            int j, k=0;
            for(j=i; buf[j]!=NULL; j++)
            {
                vals2[k]=buf[j];
                k++;
            }
            break;
        }

        else vals1[i]=buf[i];
    }

    //-----//
    val1= simple_strtoul(vals1, &endp1, 10); // karakterler tanıdıktan sonra, "char" tipinden "long"
    val2= simple_strtoul(vals2, &endp2, 10); // tipine dönüştürülerek uygun değişkenlere aktarılır.
    //----- Operatöre Göre Yapılacak İşlemi Belirleme -----//
    switch(op){
        case '+' : result = val1 + val2; break;
        case '-' : result = val1 - val2; break;
        case 'x' : result = val1 * val2; break;
        case '/' : result = val1 / val2; break;}

    //-----//
    printk (KERN_INFO "%ld %c %ld = %ld\n", val1, op, val2, result); // Yapılan işlemi "log"a yazar.
    return result;
}

/*****/

```

```
/****** "insmod" komutu ile çalışacak olan init fonksiyonu *****/
```

```
static int __init dortislem_init(void)
```

```
{
```

```
    struct proc_dir_entry *proc_odevi;
```

```
    struct proc_dir_entry *proc_odevi_dortislem;
```

```
    proc_odevi = proc_mkdir("proc_odevi", 0); // "proc_odevi" adlı proc dizinini oluşturduk.
```

```
    if (!proc_odevi) {
```

```
        printk (KERN_ERR "Dizin oluşturulamadı!: /proc/proc_odevi\n");
```

```
        return -ENOMEM;
```

```
    }
```

```
    proc_odevi_dortislem = create_proc_read_entry("proc_odevi/dortislem", 0, NULL, procfile_read, &proc_odevi); // proc_odevi klasörünün içine dortislem sanal dosyasını oluşturduk.
```

```
    if (!proc_odevi_dortislem) {
```

```
        printk (KERN_ERR "Sanal dosya oluşturulamadı!: /proc/proc_odevi/dortislem\n");
```

```
        remove_proc_entry("proc_odevi", 0);
```

```
        return -ENOMEM;
```

```
    }
```

```
    proc_odevi_dortislem->write_proc = procfile_write; // dortislem dosyasına yazma işleminin yapılması "procfile_write" fonksiyonu ile belirleniyor.
```

```
    return 0;
```

```
}
```

```
/******
```

```
/****** "rmmod" komutu ile çalışacak olan cleanup (exit) fonksiyonu *****/
```

```
static void __exit dortislem_exit(void)
```

```
{
```

```
    remove_proc_entry("proc_odevi/dortislem", 0); // proc sanal dosyası
```

```
    remove_proc_entry("proc_odevi", 0); // ve dizini silinir.
```

```
}
```

```
/******
```

```
module_init(dortislem_init);
```

```
module_exit(dortislem_exit);
```

3. MAKEFILE DOSYASI

```
obj-m += four.o

all:

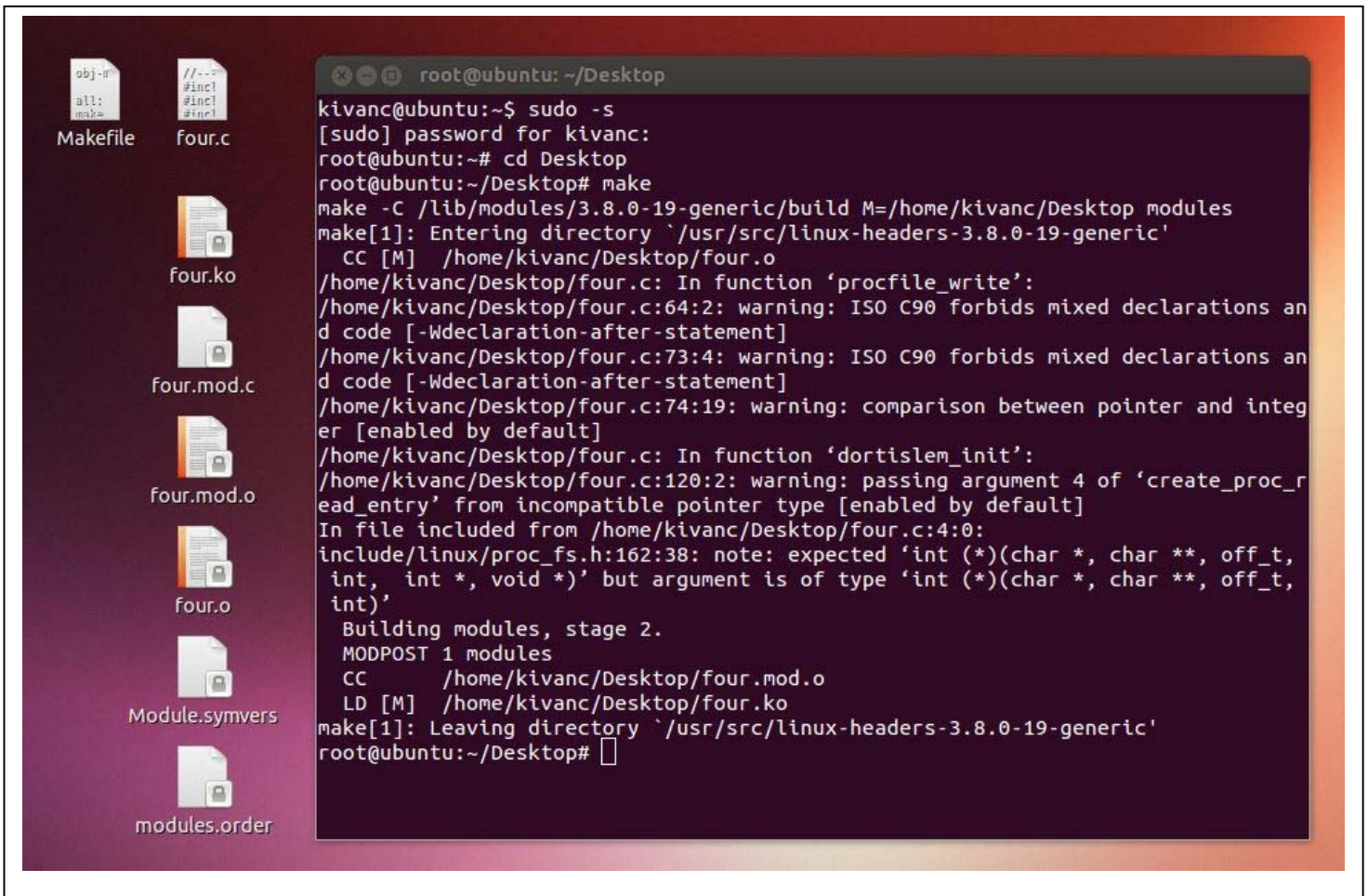
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

clean:

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

4. DERLEME

make komutuyla Makefile dosyası çağırılır ve kod derlenir. Derleme işleminden sonra içinde four.ko dosyası da olan bir takım dosya oluşur.



5. MODÜLÜ KERNEL'A YÜKLEME - INSMOD

```
root@ubuntu:~/Desktop# insmod four.ko
root@ubuntu:~/Desktop#
```

6. MODÜLÜ ÇALIŞTIRMA - ECHO, CAT VE LOG KONTROLÜ

Toplama

```
root@ubuntu:~/Desktop# echo 724 + 26 > /proc/proc_odevi/dortislem
root@ubuntu:~/Desktop# cat /proc/proc_odevi/dortislem
750
root@ubuntu:~/Desktop# grep "724 + 26 = 750" /var/log/syslog
Dec  7 15:20:28 ubuntu kernel: [ 7581.693195] 724 + 26 = 750
root@ubuntu:~/Desktop#
```

Çıkarma

```
root@ubuntu:~/Desktop# echo 2016 - 1881 > /proc/proc_odevi/dortislem
root@ubuntu:~/Desktop# cat /proc/proc_odevi/dortislem
135
root@ubuntu:~/Desktop# grep "2016 - 1881 = 135" /var/log/syslog
Dec  9 07:59:39 ubuntu kernel: [ 1336.542225] 2016 - 1881 = 135
root@ubuntu:~/Desktop#
```

Çarpma

```
root@ubuntu:~/Desktop# echo 11 x 41 > /proc/proc_odevi/dortislem
root@ubuntu:~/Desktop# cat /proc/proc_odevi/dortislem
451
root@ubuntu:~/Desktop# grep "11 x 41 = 451" /var/log/syslog
Dec  9 10:26:19 ubuntu kernel: [10131.168191] 11 x 41 = 451
root@ubuntu:~/Desktop#
```

Bölme

```
root@ubuntu:~/Desktop# echo "8 / 2" > /proc/proc_odevi/dortislem
root@ubuntu:~/Desktop# cat /proc/proc_odevi/dortislem
4
root@ubuntu:~/Desktop# grep "8 / 2 = 4" /var/log/syslog
Dec  9 11:14:53 ubuntu kernel: [ 118.955917] 8 / 2 = 4
root@ubuntu:~/Desktop#
```

Not: Bölme işleminde, sistem '/' işaretini regular expression olarak algıladığı için çift tırnak arasında kullanılmıştır.

7. MODÜLÜ KERNEL’DAN SİLME – RMMOD

```
root@ubuntu:~/Desktop# rmmmod four.ko
root@ubuntu:~/Desktop#
```

8. KARŞILAŞILAN SORUNLAR

`create_proc_entry()`; kullanımının, 3.9 kernel sürümünden sonra `proc_create()`; olarak değişmesinden kaynaklı `proc` oluşturmama sorunu yaşandı. 3.8 kernel sürümü kullanılarak sorun giderildi.

9. ÜYE NOTLARI

- | | | |
|-------------------------|--------------------------|-----|
| • KADIR KIVANC | 1306130027 - Grup Lideri | 100 |
| • HALİL DEMİR | 1306130019 | 100 |
| • MUHAMMED FATİH CAMBEK | 1306130030 | 100 |
| • OMER ALI OZBEK | 1306130071 | 100 |
| • ADEM ENES ULUTAS | 1306140097 | 100 |