Sistem Programlama

Final Ödevi:

Character Device uygulaması-arduino ile iletişim

1.Kodlar

CharacterDev

```
#include<linux/kernel.h>
#include<linux/init.h>
#include<linux/module.h>
#include<linux/syscalls.h>
#include<linux/string.h>
#include<linux/file.h>
#include<linux/fs.h>
#include<linux/fcntl.h>
#include<asm/uaccess.h>
#include<linux/delay.h>
#define FILE_DIR "/dev/ttyUSB0"  //Arduino cihazının bağlı olduğu port

//Module nitelikleri
MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("Character Device Program");
MODULE_AUTHOR("ALFA");


static char msg[50];   //Argüman metnini tutar
static short readPos=0;  //Okuma pozisyonu
static int times=0;   //Character device erişim sayısını tutar


static int dev_open(struct inode *,struct file *);
static int dev_rls(struct inode *,struct file *);
static ssize_t dev_read(struct file *,char *,size_t,loff_t *);
static ssize_t dev_write(struct file *,const char*,size_t,loff_t *);

static void write_file(const char*,const char*);

static struct file_operations cmd_file_ops={
      .read=dev_read,
      .open=dev_open,
      .write=dev_write,
      .release=dev_rls
};

int init_module(){
   printk(KERNEL_INFO "Inıt module girdi\n");
      int MajorNum=register_chrdev(70,"CharacterDev",&cmd_file_ops);
      if(MajorNum<0)printk(KERN_ALERT "Yüklenirken hata oldu..\n");

      else printk(KERN_INFO "Yüklendi\n");
```

```
	return MajorNum;
}

void cleanup_module(){
    unregister_chrdev(70,"CharacterDev");
	printk(KERN_INFO"Silindi\n");

}

static int dev_open(struct inode *inod,struct file *fil){
	times++;
	printk(KERN_INFO "Cihaz %d kere açıldı\n",times);
	return 0;
}

static ssize_t dev_read(struct file*filp,char *buff,size_t len,loff_t *off)
{
   printk(KERN_INFO "dev_read a girdi\n");
	short count =0;   //Okunan karakter sayısını tutan değişken
	while(len && (msg[readPos]!=0))
{
	put_user(msg[readPos],buff++);
	count++;
	len--;
	readPos++;
}
	return count;
}
static ssize_t dev_write(struct file *filp,const char *buff,size_t len,loff_t *off)
{
   printk(KERN_INFO "dev_write a girdi\n");
	short ind =0;
	short count=0;
	memset(msg,0,50);   //Metin için hafıza tahsis edilir
	readPos=0;

	while(len>0){
		msg[count++]=buff[ind++];
	len--;
}
	write_file(FILE_DIR,msg);
	return count;
}
static int dev_rls(struct inode *inod,struct file *fil){
	printk(KERN_INFO"Cihaz kapandı\n");
	return 0;
}
```

```
static void write_file(const char*filename,const char* data){
   printk(KERN_INFO "write file a girdi");
      struct file *filp;
mm_segment_t oldfs;
      int ret;
      int count;
      filp=filp_open(filename,O_RDWR|O_APPEND,0644);

      if(IS_ERR(filp))
      {
      printk("Açılma hatası...\n");
      return;
      }
      for(count=0;data[count]!='\0';count++);

      msleep(5000);

      printk(KERN_INFO"Mesaj: %s",data);

      oldfs=get_fs();
      set_fs(get_ds());
      ret=kernel_write(filp,data,count,&filp->f_pos);

      set_fs(oldfs);
      filp_close(filp,NULL);
}
```

CKod

```
#include<stdio.h>
#include<fcntl.h>
#include<assert.h>
#include<string.h>
#include<ctype.h>
#include<stdlib.h>
#include<unistd.h>

int main(int argc,char *argv[]){

   assert(argc>1);            //Terminalden argüman girilip girilmediği kontrol edilir

   FILE *arduino;
   int index;                 //Argüman indislerini tutan değişken

   char buf[20];              //Argüman olarak alınan metni tutan değişken
```

```
   memset(buf,0,20);           //metin için hafıza tahsis edilir

   strcpy(buf,argv[1]);


   for(index=2;index<argc;index++){
      strcat(buf," ");
      strcat(buf,argv[index]);
   }

   int fd=open("/dev/CharacterDev",O_RDWR);

   write(fd,buf,strlen(buf));

   close(fd);
   return 0;
}
```

Arduino Kodu

```
#include <LiquidCrystal.h>


LiquidCrystal lcd(12, 11, 5, 4, 3, 2);


void setup() {
  lcd.begin(16, 2);
  lcd.print("Grup Adi : Alfa..");
  Serial.begin(9600);
}



  if(Serial.available()>0){
    lcd.setCursor(0,0);
    for (int i = 0; i < 16; ++i){
      lcd.print(' ');
    qz}
    lcd.setCursor(0,0);
    lcd.print("Grup uyeleri : ");
```

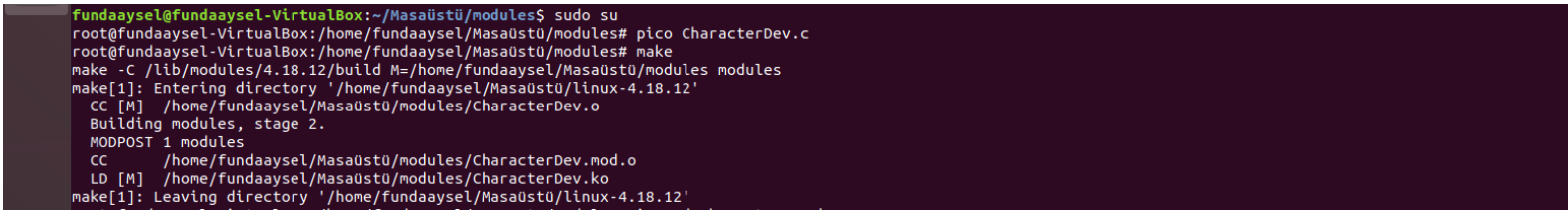
```
  lcd.setCursor(0,1);
  String input = Serial.readString();
  Serial.println(input);
  lcd.print(input);
  }
 }
```

Makefile

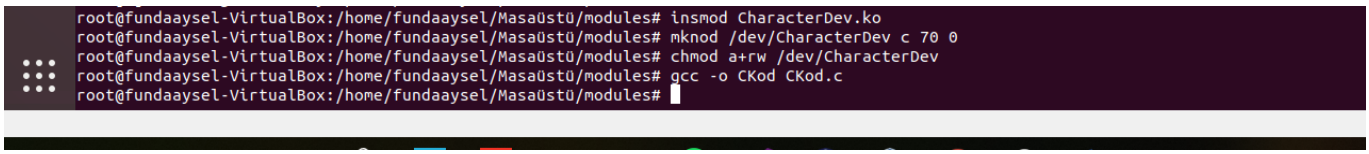
```
obj-m += CharacterDev.o

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:
	cc -o CKod CKod.c
```

2.Ekran Görüntüleri

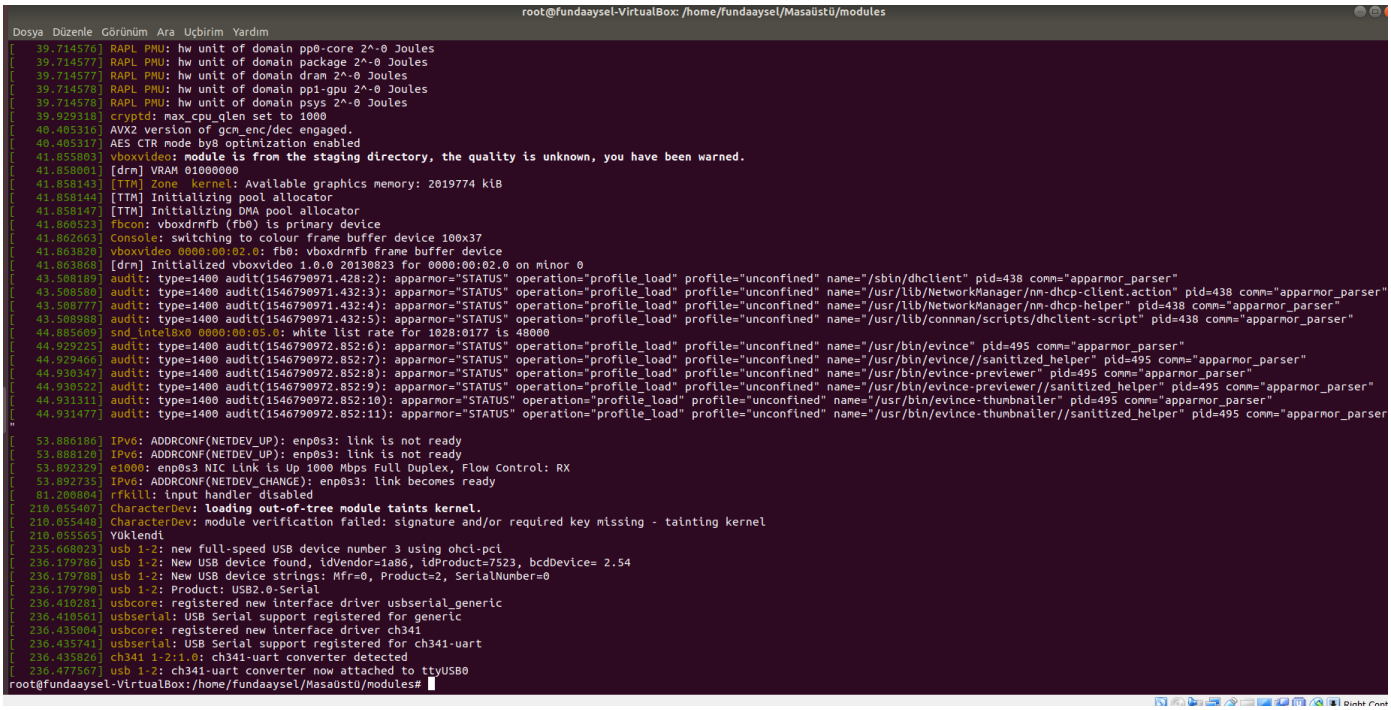
- make komutu



- insmod komutu



- dmesg



- Arduino'ya veri gönderme