

Sistem Programlama Proje Odevi

2023 - 2024 Bahar Donemi Sistem Programlama Proje Odevi

Hazirlayanlar

- Mustafa Karadeniz G211210066 2\A
- Necmi Erturk B231210501 2\B

Istenerler

- Arguman olarak verilen giris dosyasi acilir, eger yoksa varsayilan bir dosya acilir.
- Dosyada bulunan komutlara gore gerekli islemler yapilir.
- Olusan sonuc arguman olarak verilen cikis dosyasina yazilir, eger yoksa varsayilan bir dosyaya yazilir.
- Kullanicinin verdigi dosyalar ve icerikleri ne olursa olsun program cokmemelidir.
- Programda memory leak olusmamalidir.

Kullanim

- `make run` varsayilan dosyalarla calistirir
- `./bin/main.out "path_to_input" "path_to_output"`

Komutlar

yaz

- **Kullanim:** `yaz: 1 z 2 a 3 b 1 c 1 \b`
- **Islev:** Isaretcinin gosterdigi yerden itibaren verilen sayi kadar karakter yazdirir. Dosyada varolan karakterlerin ustune yazmaz. Arguman siniri yoktur.

sil

- **Kullanim:** `sil: 2 a`
- **Islev:** Isaretcinin gosterdigi yerden itibaren baslangica dogru verilen sayida karakter silene kadar ilerler. Eger yeteri sayida bulamadan basa gelirse herhangi bir islem yapmaz. Arguman olarak sadece 1 karakter alir.

sonagit

- **Kullanim:** `sonagit:`
- **Islev:** Isaretciyi dosyanin sonunu gosterecek sekilde ayarlar. Argumani yoktur

dur

- **Kullanim:** `dur:`
- **Islev:** Olusan sonuc dosyaya yazdirilir ve dosya kapatilir. Argumani yoktur.

Ornekler

- `giris.dat:`

```
yaz: 10 a 1 \b 8 k 4 r 6 u
sil: 4 a
sonagit:
yaz: 3 b 4 c 1 \n
yaz: 2 s
sonagit:
dur:
```

- `cikis.dat`

```
aaaaaa kkkkkkkrrrruuuuuubbbcccc
ss
```

- `giris.dat:`

```
yaz: 3 z 3 a 2 \b 1 c 3 a 2 \n
sil: 15 a
yaz: 1 x
sonagit:
sil: 2 \n
sonagit:
yaz: 1 F
dur:
```

- cikis.dat

```
xzzz cF
```

myio.h

Odevde istenen ozellikleri saglamak icin yazdigim stdio benzeri kutuphane.

```
#ifndef MYIO_H
#define MYIO_H

#include <stdlib.h> //malloc
#include <string.h>

# ifdef SEEK_CUR
# define SEEK_CUR 1
# endif

# ifdef SEEK_SET
# define SEEK_SET 0
# endif

# ifdef SEEK_END
# define SEEK_END 2
# endif

# ifdef EOF
# define EOF (-1)
# endif

typedef long long my_fpos;

typedef struct s_myFILE
{
    my_fpos _fpos; //file offset
    char *_buff; //buffer to store output
} myFILE;

myFILE *ft_open(); //inits an myFILE struct
void ft_close(myFILE *file); //destroys an myFILE struct

void ft_putc(myFILE *file, char c); //inserts given char into current offset (stdio writes on it instead of inserting)
void ft_puts(myFILE *file, char *str); //inserts given string into current offset (stdio writes on it instead of inserting)
void ft_truncate(myFILE *file, int len); //truncates the file to current offset (stdio truncates to given length)
int ft_getc(myFILE *file); //returns a char from current offset and increments the offset
int ft_tell(myFILE *file); //returns the current offset
int ft_seek(myFILE *file, int offset, int flag); //sets the current offset based on given flag, behaves exactly like stdio's fseek()

#endif
```