

JEGYZŐKÖNYV

Operációs rendszerek BSc

2021 tavasz féléves feladat

Készítette: Énekes Zoltán László
Neptunkód: CIJA2K

A feladat leírása: Készítsen olyan C programot, amely egy nevesített csővezetéken keresztül fogad szavakat és ezeket egy fájlba menti egy sorszámmal együtt. Illetve készítsen egy olyan kliensprogramot, ami ebbe a csőbe adatokat helyez el, ami a felhasználótól kér be.

A feladat elkészítésének lépései:

Azt a programot írom meg elsőnek, mely adatokat helyez a csőbe. Elsőnek a megfelelő könyvtárakat importálom, melyek szükségesek a program futásához.

```
#include<stdio.h>
#include<unistd.h>
#include<sys/types.h>
#include<string>
#include<sys/stat.h> //Elnevezett csőt teszi speciálissá
#include<fcntl.h> //Fájl manipulását írja le
#include<signal.h> //Jelek használata
```

Ezután írok egy függvényt, mely a jeleket kezeli. A SIGTERM fogja érzékelni a jeleket. A fájl kimeneti bufferét a fflush() üríti, majd kilép az _exit() hívással.

```
void sigkezelő(int sig) { //jelkezelő függvény
    signal(SIGTERM,SIG_IGN); //Jelek elkapása, a SIG_INT a signal függvényt segíti, a SIG_IGN a jelet ne ignorálja és kapja el
    printf("\n%d jelet kaptam \n", sig); //Kapott jelek kiírása
    fflush(stdout); //Tisztítás, konzolba íratás
    signal(SIGTERM, SIG_DFL);
    _exit(2); //kilépés
}
```

Az elnevezett cső vezeték (FIFO) író programot megírom a main függvénybe. Itt lépésről-lépésre megoldottam, a fájl megnyitását deklarálom és egy tömböt hozok létre, amire később lesz szükségem. A FIFO fájlt amit „mentes”-nek neveztem el létreálja és írhatóvá-olvasható joga lesz..

```
int main(){
    int fd; // A nyitott fájlt azonosítja
    signal(SIGINT, sigkezelő); //Függvény hívás, a SIGINT-el a futtatás előrehelyezése
    char szoveg[256]; //A tömb deklarálása 256 max hosszúságú karakterre
    mkfifo("mentes", S_IWUSR | S_IRUSR ); //Létrehozzuk a "mentes" fájlt
    A S_IWUSR| S_IRUSR pedig írható-olvashatóvá teszi a fájlt/
```

Egy végtelen ciklusba scanf hívással írhatunk bele és a szavakat eltudjuk tárolni a „mentes” fájlba. Ezek mentése és felismerése után bezárja.

```
while(1) {
    scanf("%s", szoveg); //Karakterek beolvasása
    fd=open("mentes",O_WRONLY); //fd=open megnyitja a fájlt,
    write(fd, szoveg, 12); //Fájlba írás, azonosítása és szöveg felismerése, byteok olvasásas
    close(fd); //fájl bezárása
}
}
```

Az olvasó program megírás következik. A header állományok ugyan azok mint az író programba. Szinte hasonló dolgokat lehet észre venni.

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

A jelek kiírását ismét kiíratom külön függvényben.

```
void handle_sigint(int sig) //A ctrl+c terminálása
{
    signal(SIGTERM,SIG_IGN); //Jelek elkapása
    printf("%d jelet kaptam\n", sig); //Jelek kiírása
    _exit(2);
}
```

A main függvénybe a fontosabb dolgokat deklarálom, ami szükséges a sorszámozáshoz, mert a szavakat a csővezetékéből így fogom tudni sor számozni.

```
signal(SIGINT, handle_sigint); //Függvényhívás
char s[1024]; //Tümb deklarálása, méretadás
int fd; //Fájl azonosítás
int sorszam = 1; //Listázás kezdőérték
```

Ami fontos itt az olvasáshoz, hogy a nevesített csővezetékéből tudjon is olvasni a program. A létezését próbálom feltételezni. Ha nem létezik akkor kiírja hogy nem létezik.

```
while(1) {
    fd=open("mentes",O_RDONLY); //A fájl megnyitja, de csak olvassa
    if(fd == -1) { //A fájl létezését teszteli
        printf("Fajl nem letezik.\n"); //Ha nem létezik, akkor kiírja
        return 1;
    }
```

Ha létezik a csővezeték akkor kiolvassa és egy for ciklus segítségével sor számozza, amennyi szó el van tárolva. Ezeket kiíratja és bezárja a fájlt, leáll a program.

```
        read(fd,s,sizeof(s)); //A fájl tömbjét olvassa be
        for(sorszam = 1; sorszam <= s[1024]; sorszam++) { //amennyi szó van, addig sorszámoz
            printf("%d. %s\n",sorszam, s); //A szavakat sorszámozza
        }
        return 0;
    }
    close(fd);
}
```

A teljes program:

write.c

```
#include<stdio.h>
#include<unistd.h>
#include<sys/types.h>
#include<string>
#include<sys/stat.h> //Elnevezett csöt teszi speciálissá
#include<fcntl.h> //Fájl manipulását írja le
#include<signal.h> //Jelek használata

void sigkezeló(int sig) { //jelkezelő függvény
    signal(SIGTERM,SIG_IGN); //Jelek elkapása, a SIG_INT a signal függvényt segíti, a SIG_IGN a jelet ne ignorálja és
    kapja el
    printf("\n%d jelet kaptam \n", sig); //Kapott jelek kiírása
    fflush(stdout); //Tisztítás, konzolba iratás
    signal(SIGTERM, SIG_DFL);
    _exit(2); //kilépés
}

int main(){
    int fd; // A nyitott fájlt azonosítja
    signal(SIGINT, sigkezeló); //Függvény hívás, a SIGINT-el a futtatás előrehelyezése
    char szoveg[256]; //A tömb deklarálása 256 max hosszúságú karakterre
    mkfifo("mentes", S_IWUSR | S_IRUSR ); //Létrehozzuk a "mentes" fájlt
    //A S_IWUSR| S_IRUSR pedig írható-olvashatóvá teszi a fájlt/

    while(1) {
        scanf("%s", szoveg); //Karakterek beolvasása
        fd=open("mentes",O_WRONLY); //fd=open megnyitja a fájlt,
        write(fd, szoveg, 12); //Fájlba írás, azonosítása és szöveg felismerése, byteok olvasás
        close(fd); //fájl bezárása
    }
}
```

read.c

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

void handle_sigint(int sig) //A ctrl+c terminálása
{
    signal(SIGTERM,SIG_IGN); //Jelek elkapása
    printf("%d jelet kaptam\n", sig); //Jelek kiírása
    _exit(2);
}

int main()
{
    signal(SIGINT, handle_sigint); //Függvényhívás
    char s[1024]; //Tömb deklarálása, méretadás
    int fd; //Fájl azonosítás
    int sorszam = 1; //Listázás kezdőérték
    while(1) {
        fd=open("mentes",O_RDONLY); //A fájl megnyitja, de csak olvassa
        if(fd == -1) { //A fájl létezését teszteli
            printf("Fájl nem létezik.\n"); //Ha nem létezik, akkor kiírja
            return 1;
        }
        else {
            read(fd,s,sizeof(s)); //A fájl tömbjét olvassa be
            for(sorszam = 1; sorszam <= s[1024]; sorszam++) { //amennyi szó van, addig sorszámoz
                printf("%d. %s\n",sorszam, s); //A szavakat sorszámozza
            }
            return 0;
        }
    }
    close(fd);
}
}
```

Feladatmegoldás lépései:

- 1. lépés:** Író program létrehozása, jelérzékelő függvény megírása.
- 2. lépés:** Deklarációk elvégzése a fájlhoz és szöveghez, függvényhívás.
- 3. lépés:** Csővezeték létrehozása.
- 4. lépés:** Szöveg beolvasása egy végtelen ciklusban, visszatérés.
- 5. lépés:** Olvasó program létrehozása, jelek kiírása függvénnyel, amit sikerül elkapnia
- 6. lépés:** Függvény meghívása és fontosabb dolgok deklarálása
- 7. lépés:** Nevesített csővezeték létezésének feltételezésének menete
- 8. lépés:** Ha létezik a fájl, akkor sor számozva írja ki a csővezetékből a szöveget, visszatérés

A futtatás eredménye:

enekes1@jerry:~/beadando\$./write alma masodik os alairas ^C 2 jelzest kaptam	enekes1@jerry:~/beadando\$./read 1. alma 2. masodik 3. os 4. alairas ^C 2 jelzest kaptam
--	---

```
enekes1@jerry:~/beadando$ ./read
Fajl nem letezik.
enekes1@jerry:~/beadando$
```

```
UW PICO 5
1. alma
2. masodik
3. os
4. alairas
```