Operációs rendszerek BSc

2021. 03. 24

7. gyak.

Készítette: **Munkácsi János**

Neptunkód: **X3PRVX**

1. Feladat: RR Nélkül
   - A, B, C, D Processz ebben a sorrendben
   - p_uspri = 60
   - A, B, C p_nice = 0, D p_nice = 5
   - p_cpu = 0
   - 1-től 201-ig

     p_cpu = p_cpu * KF, ahol KF értéke ½
     p_pri = P_USER + p_cpu / 4 + 2 * p_nice
     P_USER konstans, értéke 50

2. Feladat: A tanult rendszer hívásokkal (open(), read()/write(), close()) írjanak egy neptunkod_openclose.c programot amely megnyit egy fájlt – neptunkod.txt, tartalma: hallgató neve, szak, neptun kód. A program következő műveleteket végezze el:
   - olvassa be a neptunkopd.txt-t, attribútuma: O_RDWR
   - hiba ellenőrzés
   - write()
   - read()
   - lseek() – SEEK_SET-et a fájl elejére állítva

fcntl.h könyvtár:
A file-al való dolgozáshoz szükségünk lesz az fcntl.h könyvtár beágyazására. A feladata követelményét, vagyis az O_RDWR zászló használatát ez a könyvtár teszi lehetővé, ahogy az open() parancsot is.

open() és a fileDescriptor változó
Az open() parancs 2 argumentumot fogad el. Az elérési útvonalat és a fájl feldolgozásának módját. Az O_RDWR zászlóval olvasást és írást érünk el. Az open() parancs egy nemnegatív egész visszatérési értékével rendelkezik, ami -1 esetén hibát jelez.

int fileDescriptor;
fileDescriptor = open(„X3PRVX.txt”, O_RDWR);

Hibakezelés
Ha -1 értéket kapunk vissza az open()-től, akkor exit() parancsal kiiktathatjuk. Más visszatérési érték esetén haladhatunk tovább a fájl olvasásával.

```
fileDescriptor = open(„X3PRVX.txt”, O_RDWR);
if (fileDescriptor == -1)
{
        perror(„open() hiba:”);
        exit(fileDescriptor);
}
printf(„fileDescriptor visszatérési értéke: %d\n”, fileDescriptor);
```

Fájl kurzor pozícionálás lseek() metódussal
Az lseek()-hez a unistd.h könyvtárat kell beágyaznunk.

Az lseek metódus 3 argumentet fogad:
- A fileDescriptor visszatérési értéke
- Egy offset egész számot, amely megadja, hogy melyik bytre irányítjuk a kurzort.
- SEEK_SET, SEEK_CUR, SEEK_END

Visszatérési értéke az lseek()-nek egy egész szám. -1 hibát jelent, a többi visszatérési érték, pedig a kurzor helyét jelzi.

```
seekInfo = lseek(fileDescripor, 0, SEEK_SET);
if (seekInfo == -1)
{
        perror(„Pozicionálási hiba”);
        exit(seekInfo);
}
printf(„A kurzor pozíciója: %d\n”, seekInfo);
```

Fájlból olvasás read() parancsal
A read() parancs használatos fájlokból való olvasáshoz.

Az lseek()-hez hasonlóan 3 argumentumot fogad el
- Egy egész számot, ami a fájlra mutat (fileDescriptor).
- Egy buffer változót, amelybe olvassuk a fájl tartalmát.
- Egy méret változót, amivel megadhatjuk az olvasni kívánt hosszt.

Visszatérési értéke egy egész szám. -1 esetén hiba, barmely más érték esetén sikeres volt a beolvasás.

```
readInfo = read(fileDescriptor, buffer, 15);
if (readInfo == -1)
{
        perror(„Az olvasás nem volt sikeres”);
        exit(seekInfo);
}
printf(„A read() értéke: %d\n”, readInfo);
pritnf(„A beolvasott érték: %s”, buffer);
```

Fájlba írás write() paranccsal

A write() ugyan azzal a 3 argumentummal rendelkezik, mint az előbb említett read().
Visszatérési értékei is ugyan azok.

```
strcpy(buffer, „ASD123”);
bufferLength = strlen(buf);
writeInfo = write(fileDescriptor, buffer, bufferLength);

if (writeInfo == -1)
{
        perror(„Hiba írás közben”);
        exit(writeInfo);
}
printf(„A write()-al beírt byte-ok száma: %d\n”, writeInfo);
```