

JEGYZŐKÖNYV

Operációs rendszerek BSc

2022. tavasz féléves feladat

Készítette: **Szkárosi Szilárd**

Neptunkód: **DLWGQZ**

1. feladat:

IPC mechanizmus

A feladat leírása:

4.feladat: Írjon C nyelvű programot, ami létrehoz két csővezetékét (két file deszkriptor part) elforkol. A szülő elküldi a saját pidjét a gyermeknek az egyik csövön. A gyermek kiírja a képernyőre és visszaküldi egy az övét a másik csövön. Megszűnnek a processzek (a szülő megvárja a gyereket).

A feladat elkészítésének lépései:

```
// Mindkét processzhez kell egy-egy file leírót deklarálni. Ezeken keresztül fognak kommunikálni egymással
// Két elemű tömbre van szükség, mert az egyikbe írunk (1), a másikból olvasunk (0)
int pipeFileLeiroGyerek[2];
int pipeFileLeiroSzulo[2];

// Átmeneti tömb, amibe ír vagy amiből olvas az adott processz
int bufferGyerek;
int bufferSzulo;

// Egyik pipe létrehozása a szülő processzhez. Ha nem tudja létrehozni, akkor kilép.
if (pipe(pipeFileLeiroGyerek) == -1)
{
    perror("Egyik pipe hibás.");
    exit(-1);
}

// Másik pipe létrehozása a gyerek processzhez. Ha nem tudja létrehozni, akkor kilép.
if (pipe(pipeFileLeiroSzulo) == -1)
{
    perror("Másik pipe hibás.");
    exit(-1);
}

// Gyerek process ID tárolásához
// Létrehozunk egy gyerek processt, innentől kezdve lemoszódik a kód, két processz lesz.
// Az egyik a szülő, ilyenkor a fork kimenete (childPid) egyenlő a létrehozott gyerek process ID-jével.
// A másik a gyerek, ilyenkor a fork kimenete (childPid) egyenlő nullával.

pid_t childPid = fork();

if (childPid == -1)
{
    perror("Elforkol");
    exit(-1);
}
```

```
// Le kell k rni az adott process ID-j t, mert m g nem tudjuk
int aktualisProcessId = getpid();

printf("Ezt mindket process kiirja! A saját PID-em: %d\\n", aktualisProcessId);

if (childPid == 0) // Ez az  g csak a gyerek process l fut le.
{
    printf("Gyerek: gyerek process vagyok, ID-m => %d\\n", aktualisProcessId);

    // Megv rjuk, hogy a sz l  process be rjon a gyerek process file le r j ba egy int m ret  adatot (sz l  pid)
    read(pipeFileLeiroGyerek[0], &bufferGyerek, sizeof(int));

    int kapottSzuloPid = bufferGyerek;

    printf("Gyerek: szulo PID-je => %d\\n", kapottSzuloPid);

    // V laszol a sz l  process file le r j ba  s megadja a saját process ID-j t. Ez ut n r gt n tov bb megy.
    write(pipeFileLeiroSzulo[1], &aktualisProcessId, sizeof(int));

    printf("Gyerek: befejeztem\\n");

    exit(0);
}

else // Ez az  g csak a sz l  process l fut le.
{
    printf("Szulo: szulo process vagyok, ID-m => %d\\n", aktualisProcessId);

    write(pipeFileLeiroGyerek[1], &aktualisProcessId, sizeof(int));

    // Megv rjuk, hogy a gyerek process be rjon a sz l  process file le r j ba egy int m ret  adatot (gyerek pid)
    read(pipeFileLeiroSzulo[0], &bufferSzulo, sizeof(int));

    int kapottGyerekPid = bufferSzulo;

    printf("Szulo: gyerek PID-je => %d\\n", kapottGyerekPid);

    printf("Szulo: varakozok...\\n");

    // Megv rja, hogy a gyerek process befejez dj n. Ilyenkor a childPid a gyerek PID-je lesz
    wait(&kapottGyerekPid); // ugyanaz, mint wait(&childPid)

    printf("Szulo: Gyerek process befejezodott.\\n");

    exit(0);
}
```

A futtat s eredm nye:

```
szkarosil@szkarosil-VirtualBox:~$ cd OS_GYAK/OS_beadando
szkarosil@szkarosil-VirtualBox:~/OS_GYAK/OS_beadando$ gcc 4.feladat.c -o 4_feladat.out
szkarosil@szkarosil-VirtualBox:~/OS_GYAK/OS_beadando$ ./4_feladat.out
Ezt mindket process kiirja! A saját PID-em: 1823
Szulo: szulo process vagyok, ID-m => 1823
Ezt mindket process kiirja! A saját PID-em: 1824
Gyerek: gyerek process vagyok, ID-m => 1824
Gyerek: szulo PID-je => 1823
Szulo: gyerek PID-je => 1824
Szulo: varakozok...
Gyerek: befejeztem
Szulo: Gyerek process befejezodott.
szkarosil@szkarosil-VirtualBox:~/OS_GYAK/OS_beadando$
```

2. feladat:

OS algoritmusok

A feladat leírása:

35.feladat: Adott az alábbi terhelés esetén a rendszer. Határozza meg az indulás, befejezés, várakozás/átlagos várakozás és körülfordulás/átlagos körülfordulás, válasz/átlagos válaszidő és a CPU kihasználtság értékeket az FCFS ütemezési algoritmusok mellett! (cs: 0,1ms; sch: 0,1ms)

A feladat elkészítésének lépései:

FCFS Ütemezés					
	P1	P2	P3	P4	P5
Érkezés	1	4	4	7	8
CPU idő	4	11	4	7	4
Indulás	1	5	16	20	27
Befejezés	5	16	20	27	31
Várakozás	1	5	16	20	27
Körülfordulási idő	4	12	16	20	23

Körülfordulási idők átlaga	15
Válaszidők átlaga	93
CPU kihasználtság	129
Várakozási idők átlaga	13,8

FCFS Gantt-diagram:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
P1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											

érkezéstől -> befejezésig

érkezéstől -> indulásig