

JEGYZŐKÖNYV

Operációs rendszerek BSc

2022. tavasz féléves feladat

Készítette: **Tózsér Zétény**
Neptunkód: **QGNLD2**

1. IPC mechanizmus

Írjon C nyelvű programokat, ami SIGUSR1 signal érkeztekor növeli egy változó értékét. SIGUSR2 signal hatására pedig kiírja a képernyőre az aktuális értéket.

A másik program pedig: futtataskor signal-t küld az adott processznek

Az első program, 29a. A változó 0-ról indul. Az első kezelő növeli a változót, a második kiírja az aktuális értéket. A while ciklussal növeli a változó értékét és kiírja az értékét, amíg a megfelelő szignálokat megkapja.

```
#include <stdio.h>
#include <signal.h>
#include <unistd.h>

int variable = 0;
int true = 0;

void handler1() {
    variable += 1;
    printf("increased\n");
}

void handler2() {
    printf("The current value %d\n", variable);
}

int main() {
    while(true == 0){
        signal(SIGUSR1, (void (*)()) handler1);
        signal(SIGUSR2, (void (*)()) handler2);
    }

    return 0;
}
```

```
kecske@kecske-VirtualBox:~$ ./29a
The current value 0
increased
The current value 1
increased
The current value 2
increased
The current value 3
increased
The current value 4
increased
Terminated
```

A második program, 29b. A 29a pidjét manuálisan kell megadni. Amíg a while ciklus fut, addig küld egy SIGUSR1 és egy SIGUSR2 szignált, majd alszik, hogy követhetőbb legyen a 29a kimenete.

Itt a while ciklus i < 5-ig fut és a ciklusból kilépve lelövi a 29a-t, egyszerűbb tesztelés miatt.

```
#include <stdio.h>
#include <signal.h>
#include <unistd.h>

int main() {
    pid_t pid;
    int sig;
    int i = 0;

    printf("Give the pid of 29a\n");
    scanf("%d",&pid);

    //while(i == 0){
    while(i < 5){
        kill(pid, SIGUSR1);
        kill(pid, SIGUSR2);
        sleep(2);
        i++;
    }

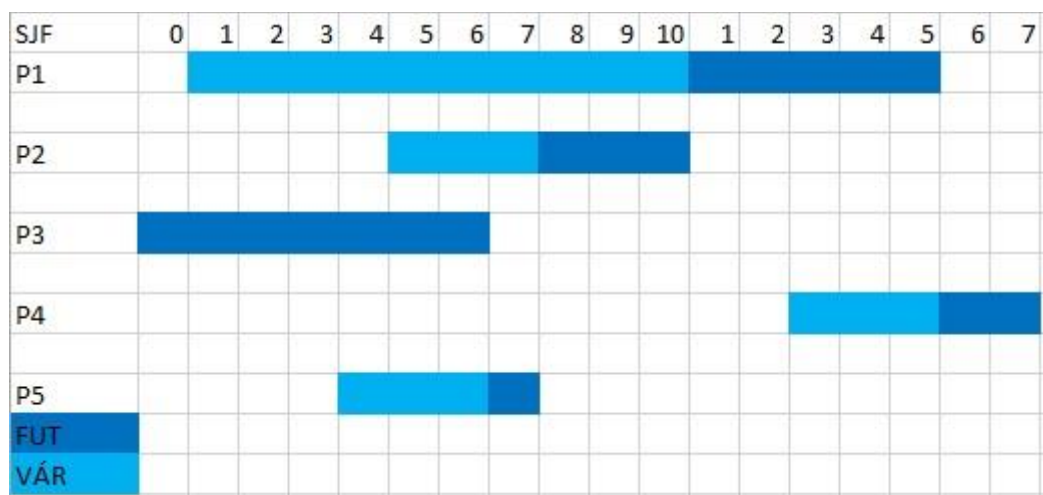
    kill(pid, SIGTERM);

    return 0;
}
```

```
kecske@kecske-VirtualBox:~$ pgrep 29a
2572
kecske@kecske-VirtualBox:~$ ./29b
Give the pid of 29a
2572
```

2. Algoritmus 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 értékeit az SJF ütemezési algoritmusok mellett! (cs: 0,1ms; sch: 0,1ms).



SJF	P1	P2	P3	P4	P5
ÉRKEZÉS	1	5	0	13	4
CPU IDŐ	5	3	7	2	1
INDULÁS	11	8	0	16	7
BEFEJEZÉS	16	11	7	18	8
VÁRAKOZÁS	10	3	0	3	3
KÖRÜLFORDULÁS	15	6	7	5	4
VÁLASZIDŐ	10	3	0	3	3

Átlagos várakozás: 3,8

Átlagos körülfordulás: 7,4

Átlagos válaszidő: 3,8

CPU kihasználtság: 97,3%