

# JEGYZŐKÖNYV

Operációs rendszerek BSc

2022. tavasz féléves feladat

**Készítette:**

Takács Bálint

P2GNFT

**Miskolc, 2022**

## A feladat leírása:

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 FCFS ütemezési algoritmusok mellett! (cs: 0,1ms; sch: 0,1ms)

## A feladat elkészítésének lépései:

Elsősorban kitöltöttem a táblázatot a futási eredményekkel:

FCFS	Érkezés	CPU idő	Indulás	Befejezés	Várakozás
P1	0	3	0	3	0
P2	2	5	3	8	1
P3	3	2	8	10	5
P4	9	5	10	15	1
P5	12	5	15	20	3

Ezután elvégeztem a feladatban szereplő számításokat a következő képpen:

$$\text{Átlagos várakozás: } \frac{\sum \text{Várakozás}}{\text{Processzek száma}}$$

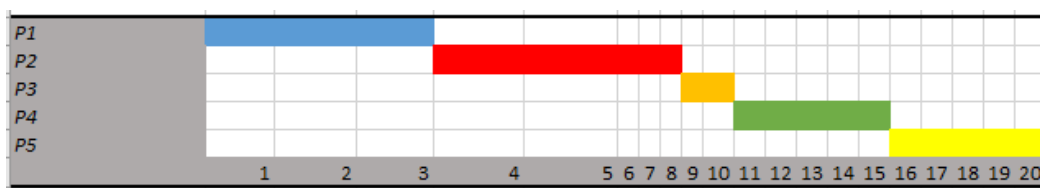
$$\text{Átlagos körülfordulás: } \frac{\sum \text{CPU idő} + \text{Várakozás}}{\text{Processzek száma}}$$

$$\text{Átlagos válaszidő: } \frac{\sum \text{Várakozás az első belépésig}}{\text{Processzek száma}}$$

$$\text{CPU kihasználtság: } \frac{\sum \text{CPU idő}}{\sum \text{CPU idő} + \text{ütemezések száma} \times \text{cs}} \times 100$$

Átlagos várakozás:	2
Átlagos körülfordulás:	6
Átlagos válaszidő:	2
CPU kihasználtság:	97,6%

Az utolsó lépésben pedig Gantt-diagrammban ábrázolom az ütemezést:



## A feladat leírása:

Írjon egy C programot, amely egy szülő processz révén készít egy gyermek processzt, a gyermekben futtasson egy másik programot az `execl()` hívással (Environemnten keresztül kapja meg, hogy mit indítson el a program), mely kiírja a PID-jét és szülője PID-jét, majd a szülő is kiírja a PID-jét és a gyermeke PID-jét.

## A feladat elkészítésének lépései:

Létrehoztam a szülő által a gyermek processzt, majd abban kiíratam a szülő és a gyermek pid-jét. Ezután létrehoztam a környezeti változót amit az `execl()`-el futtattam.

```
int parentPid = getpid();
int pid = fork();

if (pid == 0) {

    printf("Gyermek processz: \n");
    printf("A szulo pid-je: %d, A gyermek pid-je: %d\n", parentPid, getpid());
    char* prog = getenv("PROG");
    execl(prog, prog, (char*)NULL );

}
```

A következő lépésben a szülő processz által is kiíratam a gyermek és a szülő pid-jét.

```
else {

    printf("A szulo processz: \n");
    printf("A szulo pid-je: %d, A gyermek pid-je: %d\n", parentPid, pid);
    waitpid(pid, NULL, 0);

}
```