

# JEGYZŐKÖNYV

Operációs rendszerek BSc

2022. tavasz féléves feladat

**Készítette:** Sikora Dávid Ádám

**Neptunkód:** IRE699

## 1.Feladat:

Írjon C nyelvű programokat, ami  
SIGUSR1 szignál érkeztekor növeli egy változó értékét  
SIGUSR2 szignál hatására pedig kiírja a képernyőre az aktuális értéket.  
a másik program pedig: futtatáskor szignál-t küld az adott processznek

```
Activities Terminal Apr 20 8:37 PM
```

Redd15@fedora-linux--/Documents/ME/IRE699OsGyak/IRE699\_Feleves\_beadando

```
[Redd15@fedora-linux IRE699_Feleves_beadando]$ ./oi.out  
PID = 8815  
SIGUSR1 lett megadva, az értékhez hozzáadtunk egyet, a jelenlegi érték: 1  
SIGUSR1 lett megadva, az értékhez hozzáadtunk egyet, a jelenlegi érték: 2  
SIGUSR1 lett megadva, az értékhez hozzáadtunk egyet, a jelenlegi érték: 3  
SIGUSR1 lett megadva, az értékhez hozzáadtunk egyet, a jelenlegi érték: 4  
SIGUSR1 lett megadva, az értékhez hozzáadtunk egyet, a jelenlegi érték: 5  
SIGUSR1 lett megadva, az értékhez hozzáadtunk egyet, a jelenlegi érték: 6  
SIGUSR1 lett megadva, az értékhez hozzáadtunk egyet, a jelenlegi érték: 7  
SIGUSR1 lett megadva, az értékhez hozzáadtunk egyet, a jelenlegi érték: 8  
SIGUSR1 lett megadva, az értékhez hozzáadtunk egyet, a jelenlegi érték: 9  
SIGUSR2 lett megadva, az értékhez hozzáadtunk egyet, a jelenlegi érték: 10  
Kilépés!
```

Redd15@fedora-linux--/Documents/ME/IRE699OsGyak/IRE699\_Feleves\_beadando

```
[Redd15@fedora-linux IRE699_Feleves_beadando]$ ls  
IRE699_1.c IRE699_2.c Jegyzokonyv.docx oi.out oi.out Tables.xlsx  
[Redd15@fedora-linux IRE699_Feleves_beadando]$ ./oi.out  
Adja meg a másik processz PID-jét!  
PID: 8815  
Milyen signalt oajt küldeni? USR1, vagy USR2? [válaszlehetőségek: '1'/'2', más esetben a program kilép]: 1  
  
ret : OUSR1 signal kitudve a (PID: 8815) processznek!  
Milyen signalt oajt küldeni? USR1, vagy USR2? [válaszlehetőségek: '1'/'2', más esetben a program kilép]: 1  
  
ret : OUSR1 signal kitudve a (PID: 8815) processznek!  
Milyen signalt oajt küldeni? USR1, vagy USR2? [válaszlehetőségek: '1'/'2', más esetben a program kilép]: 1  
  
ret : OUSR1 signal kitudve a (PID: 8815) processznek!  
Milyen signalt oajt küldeni? USR1, vagy USR2? [válaszlehetőségek: '1'/'2', más esetben a program kilép]: 1  
  
ret : OUSR1 signal kitudve a (PID: 8815) processznek!  
Milyen signalt oajt küldeni? USR1, vagy USR2? [válaszlehetőségek: '1'/'2', más esetben a program kilép]: 1  
  
ret : OUSR1 signal kitudve a (PID: 8815) processznek!  
Milyen signalt oajt küldeni? USR1, vagy USR2? [válaszlehetőségek: '1'/'2', más esetben a program kilép]: 1  
  
ret : OUSR1 signal kitudve a (PID: 8815) processznek!  
Milyen signalt oajt küldeni? USR1, vagy USR2? [válaszlehetőségek: '1'/'2', más esetben a program kilép]: 1  
  
ret : OUSR2 signal kitudve a (PID: 8815) processznek!  
Milyen signalt oajt küldeni? USR1, vagy USR2? [válaszlehetőségek: '1'/'2', más esetben a program kilép]: 3  
  
[Redd15@fedora-linux IRE699_Feleves_beadando]$
```

Itt látható a két program futási eredménye. A jobb oldali terminálban megnyitott program a cél processz PID-jének megadása után két dolgot tud megvalósítani, a SIGUSR1 vagy SIGUSR2 szignál kiküldését.

A bal oldali terminálablakban futó program pedig ezen szignálok beérkezésére reagál. Ha USR1 szignált kap, akkor egy értéket megnövel 1-gyel, USR2 esetén pedig kiírja az adott értéket majd kilép a program.

Az előbbi program a kill rendszerhívással küldi el a választásunknak megfelelő szignált az utóbbinak.

A C programok forráskódja a mappa IRE699\_1.c illetve IRE699\_2.c fileokban láthatóak.

## 2.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)

	P1	P2	P3	P4
Érkezés	0	8	12	20
CPU idő	15	7	26	10
Indulás				
Befejezés				
Várakozás				

Ábrázolja Gantt diagram segítségével az aktív/várakozó processzek futásának menetét.

Magyarázza a kapott eredményeket!

SJF	Érkezés	CPU idő	Indulás	Befejezés	Várakozás	Körülfordulás	Válaszidő		Átlagos Várakozás:	7.25	
P1	0	15	0	15	0	15	0		Átlagos Körülfordulás:	21.75	
P2	8	7	15	22	7	14	7		Átlagos Válaszidő:	7.25	
P3	12	26	32	58	20	46	20		CPU Kihasználtság:	99%	
P4	20	10	22	32	2	12	2		Befejezési idő: 58		Végrehajtási sorrend: P1;P2;P4;P3

A fenti táblázatban láthatóak a számítások a processzek végrehajtási sorrendjéről. Az SJF algoritmus minden processz végekor a már beérkezett, megjósoltan legrövidebb processzt fogja következőként mindig futtatni.

Látható továbbá a végrehajtás sorrendje is, valamint az algoritmus jelen rendszerre vonatkoztatott hatékonysági mutatói. A processzek végrehajtásának Gantt-Diagramja az alább látható.



