

# Operációs rendszerek BSc

11. Gyak.

2022. 04. 27.

**Készítette:**

Szelényi Szabolcs Bsc

Mérnökinformatikus hallgató

TYNYS9

Miskolc, 2022

„1. Adott egy rendszer (fogalási stratégiák), melyben a következő

- Szabad területek: 30k, 35k, 15k, 25k, 75k, 45k és
- Foglalási igények: 39k, 40k, 33k, 20k, 21k állnak rendelkezésre.

A rendszerben a memória 4 kbyte-os blokkokban kerül nyilvántartásra, ennél kisebb méretű töredék igény esetén a teljes blokk lefoglalásra kerül. Határozza meg változó méretű partíció esetén a következő algoritmusok felhasználásával: first fit, next fit, best fit, worst fit a foglalási igényeknek megfelelő helyfoglalást – táblázatos formában (az ea. bemutatott mintafeladat alapján)! Hasonlítsa össze, hogy a teljes szabad memóriaterület hány százaléka vész el átlagosan az egyes algoritmusok esetén! A kapott eredményeket ábrázolja oszlop diagrammal! Magyarázza a kapott eredményeket és hogyan lehet az eredményeket javítani!

Szabad területek:	30k, 35k, 15k, 25k, 75k, 45k
Foglalási igények:	39k, 40k, 33k, 20k, 21k
	First fit, Next fit, Best fit, Worst fit

  

First fit	
<b>Memória terület - szabad terület</b>	
Foglalási igény	30      35      15      25      75      45
39	36 (75 - 39)
40	5 (45 - 40)
33	2 (35 - 33)
20	5 (25 - 20)
21	9 (30 - 31)

Next fit	
<b>Memória terület - szabad terület</b>	
Foglalási igény	30      35      15      25      75      45
39	36 (75 - 39)
40	5 (45 - 40)
33	2 (35 - 33)
20	5 (25 - 20)
21	15 (36 - 21)

Best fit	
<b>Memória terület - szabad terület</b>	
Foglalási igény	30      35      15      25      75      45
39	6 (45 - 39)
40	35 (75 - 40)
33	2 (35 - 33)
20	5 (25 - 20)
21	9 (30 - 31)

Worst fit	
<b>Memória terület - szabad terület</b>	
Foglalási igény	30      35      15      25      75      45
39	36 (75 - 39)
40	5 (45 - 40)
33	3 (36 - 33)
20	15 (35 - 20)
21	9 (30 - 31)

2. Gyakorló feladat: A feladat megoldásához először tanulmányozza Vadász Dénes: Operációs rendszer jegyzet, a témához kapcsolódó fejezetét (6.4)., azaz Írjon C nyelvű programokat, ahol

- kreál/azonosít szemafor készletet, benne N szemafor-t. A kezdő értéket 0-ra állítja – semset.c,

```
szelenyi@szelenyi-VirtualBox:~$ gcc semset.c -o semset.out
szelenyi@szelenyi-VirtualBox:~$ ./semset.out
```

- kérdezze le és írja ki a pillanatnyi szemafor értéket – semval.c

```
szelenyi@szelenyi-VirtualBox:~$ gcc semval.c -o semval.out
szelenyi@szelenyi-VirtualBox:~$ ./semval.out
Szemaforok tartalma:
0
0
0
0
0
szelenyi@szelenyi-VirtualBox:~$
```

- szüntesse meg a példacskák szemafor készletét – semkill.c

```
szelenyi@szelenyi-VirtualBox:~$ gcc semkill.c -o semkill.out
szelenyi@szelenyi-VirtualBox:~$ ./semkill.out
szelenyi@szelenyi-VirtualBox:~$
```

- sembuf.sem\_op=1 értékkel inkrementálja a szemafort – semup.c

```
szelenyi@szelenyi-VirtualBox:~$ gcc semup.c -o semup.out
szelenyi@szelenyi-VirtualBox:~$ ./semup.out
Nem sikerult szemaforokat lekerdezni
: No such file or directory
szelenyi@szelenyi-VirtualBox:~$
```

2a. Írjon egy C nyelvű programot, melyben

- egyik processz létrehozza a szemafort (egyetlen elemi szemafort; inicializálja 1-re, vagy x-re, ha még nem létezik),

```
szelenyi@szelenyi-VirtualBox:~$ gcc Gyak11_sem2a1.c -o Gyak11_sem2a1.out
szelenyi@szelenyi-VirtualBox:~$ ./Gyak11_sem2a1.out
Szam: 1
A szemafor erteke (1) : 1
```

- másik processz használja a szemafort, belépési szakasz (down), a kritikus szakaszban alszik 2-3 sec-et, m pid-et kiír, kilépési szakasz (up), ezt ismételve 2x-3x (és a hallgató egyszerre indítson el 2-3 ilyen processzt),

```
szelenyi@szelenyi-VirtualBox:~$ gcc Gyak11_sem2a2.c -o Gyak11_sem2a2.out
szelenyi@szelenyi-VirtualBox:~$ ./Gyak11_sem2a2.out
Kritikus szakasz
pid : 2525
0
kritikus szakasz vege
szelenyi@szelenyi-VirtualBox:~$
```

- harmadik processzben, ha létezik a szemafor, akkor megszünteti”

```
Kritikus szakasz vege
szelenyi@szelenyi-VirtualBox:~$ gcc Gyak11_sem2a3.c -o Gyak11_sem2a3.out
szelenyi@szelenyi-VirtualBox:~$ ./Gyak11_sem2a3.out
Torolve
szelenyi@szelenyi-VirtualBox:~$
```