

# JEGYZŐKÖNYV

Operációs rendszerek BSc

2022. tavaszi féléves feladat

Készítette: **Zavarkó Máté**

Neptunkód: **IN3BLK**

## 1. Feladat:

Adott egy számítógépes rendszer, amelyben a

- szabad memória területek: 23kB, 64kB, 10kB, 80kB, 12kB, 50kB és 40kB, melynek
- foglalási igénye: 65kB, 21kB, 48kB, 13kB, 62kB.

Határozza meg *változó méretű partíció* esetén a következő algoritmus felhasználásával:

***next fit, worst fit*** a foglalási igényeknek megfelelő helyfoglalást – táblázatos formában!

Magyarázza a kapott eredményeket és hogyan lehet az eredményeket javítani!

## Stratégiák a memóriaterületek lefoglalására:

- Első megfelelő (First fit): A rendelkezésre álló szabad területek közül a legelső elegendő méretűt foglaljuk le.
- Következő megfelelő (Next Fit): A keresést nem a tárr elejéről kezdjük, hanem az után a terület után, amit legutoljára foglaltunk.
- Legjobban megfelelő (Best fit): A legkisebbet foglaljuk le azon szabad területek közül, amelyek legalább akkorák, mint a lefoglalandó terület.
- Legrosszabban illeszkedő (Worst fit): Az elérhető legnagyobb szabad területet allokáljuk. A maradék terület még talán elegendő lesz egy újabb foglalás számára.

## Next fit

Next fit:	keresés az az után a terület után, amit legutoljára foglaltunk						
Foglalási igény	Szabad memória terület						
	23kB	64kB	10kB	80kB	12kB	50kB	40kB
65kB	23	64	10	65, 15	12	50	40
21kB	23	64	10	15	12	21, 29	40
48kB	23	48, 16	10	15	12	29	40
13kB	23	13, 3	10	15	12	29	40
62kB	-	-	-	-	-	-	-
Next fit:	[48]	[13]	3 10	[65]	15 12	[21]	29 40

## Worst fit

<b>Worst fit:</b>	az elérhető legnagyobb területet allokáljuk						
<b>Foglalási igény</b>	<b>Szabad memória terület</b>						
	<b>23kB</b>	<b>64kB</b>	<b>10kB</b>	<b>80kB</b>	<b>12kB</b>	<b>50kB</b>	<b>40kB</b>
<b>65kB</b>	23	64	10	65, 15	12	50	40
<b>21kB</b>	23	21, 43	10	15	12	50	40
<b>48kB</b>	23	43	10	15	12	48, 2	40
<b>13kB</b>	23	13, 30	10	15	12	2	40
<b>62kB</b>	-	-	-	-	-	-	-
<b>Worst fit:</b> [21] [13] 30 10 [65] 15 12 [48] 2 40							

Sem a *next fit*, sem a *worst fit* nem tudja allokálni a memóriát.

## Módszerek az eredmények javítására:

- **Szemétgyűjtés (garbage collection) alkalmazása**, vagyis a memória allokáció futás idejű átrendezése nem kielégíthető igények esetén. Hátránya, hogy erőforrás igényes.
- **Lapszervezés használata**. Hátránya, hogy kell hozzá MMU (Memory Management Unit) támogatás.

## 2. Feladat:

Adott az alábbi terhelés esetén a rendszer. Határozza meg az *indulás*, *befejezés*, *várakozás/átlagos várakozás* és *körülfordulás/átlagos körülfordulás*, *válasz/átlagos válasz*idő és a *CPU kihasználtság* értékeket az **RR: 6ms** ütemezési algoritmus mellett!

(cs: 0,1ms; sch0,1ms)

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

Magyarázza a kapott eredményt!

**CPU kihasználtsága:** A hasznos munkával töltött idő aránya az összes időhöz képest.

$$\frac{\sum t_{CPU,hasznos\ munka}}{\sum t_{CPU}} \cdot 100 [\%]$$

**Várakozási idő:** Az összes idő, amit a feladat várakozással töltött.

$$t_{körülfordulás} = t_{CPU,végrehajtás} + t_{várakozás} [s]$$

**Körülfordulási idő:** Egy feladatra vonatkozóan a rendszerbe helyezéstől a teljesítésig eltelt idő.

$$t_{körülfordulás} = t_{CPU,végrehajtás} + t_{várakozás} [s]$$

**Válaszidő:** A feladat megkezdésétől az első kimenetek produkálásáig eltelt idő.

## Round-Robin (6ms)

RR: 6ms	P1	P2	P3	P4	P5
Érkezés	1	4	4	7	8
CPU idő	4	11	4	7	4
Indulás	1	5	11	15	21
Befejezés	5	30	15	31	25
Várakozás	0	15	7	17	13

RR: 6ms	
CPU kihasználtsága	7 context switch + 2 ütemezés: $31 / 31,9 = 97,17\%$
Körülfordulási idők átlaga	$4 + 26 + 11 + 24 + 17 = 82 / 5 = 16,4 \text{ ms}$
Várakozási idők átlaga	$0 + 15 + 7 + 17 + 13 = 52 / 5 = 10,4 \text{ ms}$
Válaszidők átlaga	$1 + 5 + 11 + 15 + 21 / 5 = 10,6 \text{ ms}$

