

# Operációs rendszerek

Készítette: Novák László Lajos

Neptun: T8F0F0

„1. Adott egy rendszerbe az alábbi erőforrások: R (R1: 10; R2: 5; R3: 7)

A rendszerbe 5 processz van: P0, P1, P2, P3, P4

Kérdés: Kielégíthető-e P4 (3,3,0) ill. P0 (0,2,0) kérése úgy, hogy biztonságos legyen, holtpontmentesség szempontjából a rendszer - a következő *kiinduló állapot* alapján.

Igazolja a processzek végrehajtásának sorrendjét – számolással.”

Az összes osztály -erőforrások száma: (10, 5, 7)							
Kiinduló állapot							
1. lépés					2. lépés		
	MAX. IGÉNY				FOGLAL		
	R1	R2	R3		R1	R2	R3
P0	7	5	3		0	1	0
P1	3	2	2		2	0	0
P2	9	0	2		3	0	2
P3	2	2	2		2	1	1
P4	4	3	3		0	0	2

Bankár algoritmus segítségével megállapítottam, hogy a P4 (3, 3, 0) kérés nem elégíthető ki, úgy, hogy biztonságos legyen. P0 (0, 2, 0) magában kielégíthető lenne, és ha a 6. lépésben jönne be a P4(3, 3, 0) kérés, akkor az is kielégíthető lenne.

A számolások a **T8F0F0\_OS\_4gyak\_1feladat.xlsx** fájlban találhatóak.

2. Adott egy rendszer (foglalási stratégiák), melyben a következő

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

Határozza meg *változó partíció esetén* a következő algoritmusok felhasználásával: first fit, next fit, best fit, worst fit a foglalási igényeknek megfelelő helyfoglalást!

A **first fit**, **next fit**, **best fit** és **worst fit** foglalási igényeknek megfelelő helyfoglalást **T8f0f0\_OS\_4gyak\_2feladat.xlsx** fájlban határoztam meg.



		= Adott lépésben használt																				
SC		Laphivatkozások																				
Memóriakeret		7	6	5	4	6	7	3	2	6	7	6	5	1	2	5	6	7	6	5	2	
1. lap		7	7	7	4	4	4	3	3	3	7	7	7	1	1	1	6	6	6	6	6	
SC bit																			1	1	1	
2. lap			6	6	6	6	6	6	6	6	6	6	6	6	6	5	5	5	5	5	5	
SC bit						1	1	1		1	1	1	1	1					1	1		
3. lap				5	5	5	7	7	2	2	2	2	5	5	2	2	2	7	7	7	2	
SC bit																						
Laphibák		1	1	1	1	0	1	1	1	0	1	0	1	1	1	1	1	1	0	0	1	Össz: 15

  

		Laphivatkozások																				
LRU		Laphivatkozások																				
Memóriakeret		7	6	5	4	6	7	3	2	6	7	6	5	1	2	5	6	7	6	5	2	
1. lap		7	7	7	7	7	7	7	7	7	7	7	7	7	7	5	5	5	5	5	5	
SC bit							1	1	1	1	1	1	1	1					1	1		
2. lap			6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	
SC bit						1	1	1	1	1	1	1	1	1			1	1	1	1	1	
3. lap				5	5	5	5	3	3	3	3	3	5	5	2	2	2	2	2	2	2	
SC bit																					1	
4. lap					4	4	4	4	2	2	2	2	2	1	1	1	1	7	7	7	7	
SC bit																						
Laphibák		1	1	1	1	0	0	1	1	0	0	0	1	1	1	1	0	1	0	0	0	Össz: 11

Mivel nagyobb memóriakerettel több lapot tud megjegyezni a rendszer, így kevesebb lapbeolvasásra van szükség.