

Operációs rendszerek BSc

4. konzultáció gyakorlat

2021.04.30

Készítette: **Hudák Dániel**

Mérnökinformatikus MSc

Neptunkód: **MWB0JW**

Operációs rendszerek – 4. konzultáció - gyakorlat

Erőforrások kezelése (Bankár algoritmus),

Memóriamenedzselés (foglалási stratégiák, lapcsere algoritmusok)

Tölts fel az aktuális mappába: **Neptunkod_0430**

- Jegyzőkönyv neve: *neptunkod_gyak4.pdf*
- Forrás fájlok feltöltése

Határidő: 2021. 05. 09.

Irodalom

Tanulmányozza a Vadász Dénes: Operációs rendszerek, 2006. ME, jegyzet, ill. Vincze Dávid: Operációs rendszerek diasort - az adott témához kapcsolódó fejezeteket.

Benyó Balázs, Fék Márk, Kiss István, Kóczy Annamária, Kondorosi Károly, Mészáros Tamás, Román Gyula, Szeberényi Imre, és Sziray, József: Operációs rendszerek mérnöki megközelítésben, Panem Kiadó, 2000, jegyzet/diák.

Szintén tanulmányozza az előadáson kivetített URL linkhez tartozó irodalmat, majd oldja meg a feladatot.

Feladatok

„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

A tanult bankár algoritmust használva igazoltam a processek sorrendjét, majd megállapítottam, hogy nem teljesíthető P4 kérése, holtpontmentesség szempontjából a rendszer nem lesz biztonságos. A feladat megoldásának menete az MWBOJW_OS_4gy1.xlsx fájlban található.

Teljesíthető-e? a P0 (0,2,0) ill. P4 (3,3,0)				Kérés teljesítése (0,2,0); (3,3,0)						
3a. Lépés				Foglal: P0 (0,1,0) + Kérés(0,2,0) = P0 (0,3,0)						
				Foglal: P4 (0,0,2) + Kérés(3,3,0) = P0 (3,3,2)						
MAX IGÉNY				FOGLAL			Készlet	IGÉNY		
	R1	R2	R3	R1	R2	R3	0,-2,2	R1	R2	R3
P0	7	5	3	0	3	0		7	2	3
P1	3	2	2	2	0	0		1	2	2
P2	9	0	2	3	0	2		6	0	0
P3	2	2	2	2	1	1		0	1	1
P4	4	3	3	3	3	2		1	0	1
				10	7	5				

Teljesíthető-e? a P0 (0,2,0) ill. P4 (3,3,0)				Kérés teljesítése (0,2,0); (3,3,0)						
3a. Lépés				Foglal: P0 (0,1,0) + Kérés(0,2,0) = P0 (0,3,0)						
MAX IGÉNY				FOGLAL			Készlet	IGÉNY		
	R1	R2	R3	R1	R2	R3	3,1,2	R1	R2	R3
P0	7	5	3	0	3	0		7	2	3
P1	3	2	2	2	0	0		1	2	2
P2	9	0	2	3	0	2		6	0	0
P3	2	2	2	2	1	1		0	1	1
P4	4	3	3	0	0	2		4	3	1
				7	4	5				

Teljesíthető-e? a P4 (3,3,0)				Kérés teljesítése (0,2,0); (3,3,0)						
3a. Lépés				Foglal: P4 (0,0,2) + Kérés(3,3,0) = P0 (3,3,2)						
MAX IGÉNY				FOGLAL			Készlet	IGÉNY		
	R1	R2	R3	R1	R2	R3	0,0,2	R1	R2	R3
P0	7	5	3	0	1	0		7	4	3
P1	3	2	2	2	0	0		1	2	2
P2	9	0	2	3	0	2		6	0	0
P3	2	2	2	2	1	1		0	1	1
P4	4	3	3	3	3	2		1	0	1
				10	5	5				

4. Lépés										
MAX IGÉNY				FOGLAL			Készlet	IGÉNY		
	R1	R2	R3	R1	R2	R3	3,1,2	R1	R2	R3
P0	7	5	3	0	3	0		7	2	3
P1	3	2	2	2	0	0		1	2	2
P2	9	0	2	3	0	2		6	0	0
P3	2	2	2	2	1	1		0	1	1
P4	4	3	3	0	0	2		4	3	1

4a. Lépés				P1 képes lefutni						
MAX IGÉNY				FOGLAL			Készlet	IGÉNY		
	R1	R2	R3	R1	R2	R3	5,2,3	R1	R2	R3
P0	7	5	3	0	3	0		7	2	3
P1	3	2	2	2	0	0		1	2	2
P2	9	0	2	3	0	2		6	0	0
P3	2	2	2	0	0	0				
P4	4	3	3	0	0	2		4	3	1

5. Lépés				P3 igénye kielégíthető, majd lefut (5,1,2)						
MAX IGÉNY				FOGLAL			Készlet	IGÉNY		
	R1	R2	R3	R1	R2	R3	7,2,3	R1	R2	R3
P0	7	5	3	0	3	0		7	2	3
P1	3	2	2	0	0	0				
P2	9	0	2	3	0	2		6	0	0
P3	2	2	2	0	0	0				
P4	4	3	3	0	0	2		4	3	1

6. Lépés				P2 igénye kielégíthető, majd lefut (7,2,3)						
MAX IGÉNY				FOGLAL			Készlet	IGÉNY		
	R1	R2	R3	R1	R2	R3	10,2,5	R1	R2	R3
P0	7	5	3	0	3	0		7	2	3
P1	3	2	2	0	0	0				
P2	9	0	2	0	0	0				
P3	2	2	2	0	0	0				
P4	4	3	3	0	0	2		4	3	1

7. Lépés				P0 igénye kielégíthető, majd lefut (10,2,5)						
MAX IGÉNY				FOGLAL			Készlet	IGÉNY		
	R1	R2	R3	R1	R2	R3	10,5,5	R1	R2	R3
P0	7	5	3	0	0	0				
P1	3	2	2	0	0	0				
P2	9	0	2	0	0	0				
P3	2	2	2	0	0	0				
P4	4	3	3	0	0	2		4	3	1

7a. Lépés				P4 igénye kielégíthető, majd lefut (10,5,5)						
MAX IGÉNY				FOGLAL			Készlet	IGÉNY		
	R1	R2	R3	R1	R2	R3	10,5,7	R1	R2	R3
P0	7	5	3	0	0	0				
P1	3	2	2	0	0	0				
P2	9	0	2	0	0	0				
P3	2	2	2	0	0	0				
P4	4	3	3	0	0	0				

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!

Változó partíciók esetén, a tanult foglalási stratégiák felhasználásával az MWBOJW_OS_4gy2.xlsx táblázatban látható módon az igényeknek megfelelő allokációkat.

First Fit	Legelső elegendő méretű szabad terület.					
Foglalási igény	Memória területek - szabad területek					
	30	35	15	25	75	45
39	30	35	15	25	75, 36	45
40	30	35	15	25	75, 36	45, 5
33	30	35, 2	15	25	75, 36	45, 5
20	30, 10	35, 2	15	25	75, 36	45, 5
21	30, 10	35, 2	15	25, 4	75, 36	45, 5

Best Fit	A legkisebbet foglaljuk le.					
Foglalási igény	Memória területek - szabad területek					
	30	35	15	25	75	45
39	30	35	15	25	75	45, 6
40	30	35	15	25	75, 35	45, 6
33	30	35, 2	15	25	75, 35	45, 6
20	30	35, 2	15	25, 5	75, 35	45, 6
21	30, 9	35, 2	15	25, 5	75, 35	45, 6

Next Fit	Megjegyzi, hol találta az előzőt és innen indul keresni.					
Foglalási igény	Memória területek - szabad területek					
	30	35	15	25	75	45
39	30	35	15	25	75, 36	45
40	30	35	15	25	75, 36	45, 5
33	30	35, 2	15	25	75, 36	45, 5
20	30	35, 2	15	25, 5	75, 36	45, 5
21	30	35, 2	15	25, 5	75, 36, 15	45, 5

Worst Fit	A legnagyobb elérhető szabad területet foglaljuk le.					
Foglalási igény	Memória területek - szabad területek					
	30	35	15	25	75	45
39	30	35	15	25	75, 36	45
40	30	35	15	25	75, 36	45, 5
33	30	35	15	25	75, 36, 3	45, 5
20	30	35, 15	15	25	75, 36, 3	45, 5
21	30, 9	35, 15	15	25	75, 36, 3	45, 5

3. Adott egy igény szerinti lapozást használó rendszerben a következő laphivatkozás és 4 fizikai memóriakeret a processzek számára.

Laphivatkozások sorrendje: 7 6 5 4 6 7 3 2 6 7 6 5 1 2 5 6 7 6 5 2

Memóriakeret (igényelt lapok): 3 és 4 memóriakeret.

Mennyi laphiba keletkezik (mindkét memóriakeret esetén külön-külön) az alábbi algoritmusok esetén: FIFO, LRU és SC? Hasonlítsa össze és magyarázza az eredményeket.

FIFO	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			3	3	3	3		5	5	5		5	7		7	
2. lap		6	6	6			6	2	2	2		2	1	1		1	1		5	
3. lap			5	5			5	5	6	6		6	6	2		2	2		2	
4. lap				4			4	4	4	7		7	7	7		6	6		6	
FIFO	7	6	5	4	3	2	6	7	5	1	2	6	7	5						

Laphibák száma: 4+10

FIFO	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	2	2	2		5	5	5		6	6		6	2
2. lap		6	6	6		7	7	7	6	6		6	1	1		1	7		7	7
3. lap			5	5		5	3	3	3	7		7	7	2		2	2		5	5
FIFO	7	6	5	4	7	3	2	6	7	5	1	2	6	7	5	2				

Laphibák száma: 3+13

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	2			2				
2. lap		6	6	6			6	6				6	6	6			6				
3. lap			5	5			3	3				5	5	5			5				
4. lap				4			4	2				2	1	1			7				
Laphibák száma: 4+6																					

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	4		4	3	3	3	7		7	1	1		6	6			6	
2. lap			6	6	6		6	6	2	2	2		5	5	5		5	5			5	
3. lap				5	5		7	7	7	6	6		6	6	2		2	7			2	
Laphibák száma: 3+12																						

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		7,1	7,1	7,1	7,1			7,0	7,0	7,0	7,0	3,1	3,1	3,1		3,0	3,0	3,0	3,0	5,1	5,1	5,1		5,1	5,0	5,0	5,0	5,0	5,0	7,1	7,1	7,1	7,1	
2. lap			6,1	6,1	6,1			6,1	6,0	6,0	6,0	6,0	2,1	2,1	2,1		2,1	2,0	2,0	2,0	2,0	1,1	1,1		1,1	1,1	1,0	1,0	1,0	1,0	1,0	5,1	5,1	5,1
3. lap				5,1	5,1			5,1	5,1	5,0	5,0	5,0	5,0	6,1	6,1		6,1	6,1	6,0	6,0	6,0	6,0	2,1		2,1	2,1	2,1	2,0	2,0	2,0	2,0	2,0	2,1	2,1
4. lap					4,1			4,1	4,1	4,1	4,0	4,0	4,0	4,0	7,1		7,1	7,1	7,1	7,0	7,0	7,0	7,0		6,1	6,1	6,1	6,1	6,0	6,0	6,1	6,1	6,1	6,1
FIFO		7	6	5	4			7	6	5	4	3	2	6	7		3	2	6	7	5	1	2		6	5	1	2	6	7	6	5	2	
Laphibák száma: 4+10																																		

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		7,1	7,1	7,1	7,0	#	4,1	4,1	4,1	4,0	4,0	4,0	3,1	3,1	#	3,0	3,0	3,0	7,1	7,1	7,0	7,0	#	5,1	5,1	5,1	5,0	5,0	5,0	6,1	6,1	6,1	6,0	6,0	6,0	2,	
2. lap			6,1	6,1	6,1	#	#	6,0	6,1	6,1	6,1	6,0	6,0	6,0	2,1	#	2,1	2,0	2,0	2,0	2,1	2,1	2,0	#	2,0	1,1	1,1	1,1	1,0	1,0	1,0	7,1	7,1	7,1	7,0	7,0	7,
3. lap				5,1	5,1	#	#	5,0	5,0	7,1	7,1	7,1	7,0	7,0	7,0	#	6,1	6,1	6,0	6,0	6,1	6,1	6,1	#	6,0	6,0	2,1	2,1	2,1	2,0	2,0	2,0	5,1	5,1	5,1	5,0	5,
FIFO		7	6	5	7	6	5	4	6	7	4	6	7	3	2	6	3	2	6	7	6	7	2	6	5	1	2	5	1	2	6	7	5	6	7	5	2
Laphibák száma: 3+13																																					

A laphibák száma a memóriakeret függvényében:

		Laphibák	
Memóriakeret		4	3
FIFO		14	16
LRU		12	13
SC		14	16

Mivel kevesebb memóriakeret esetén, kevesebb a megjegyzett laphivatkozások száma, ezért több lapcsere következik be. Ez magyarázza, hogy ilyen esetben több a laphiba.