

Bankár algoritmus

Excel táblázat megtalálható a Github repo-ban
KLYSO1_gyak10.xlsx néven

1-2 lépés: kitöltjük a táblázatokat:

1.lépés	MAX.IGÉNY		
	R1	R2	R3
P0	7	5	3
P1	3	2	2
P2	9	0	2
P3	2	2	2
P4	4	3	3

2.lépés	FOGLAL		
	R1	R2	R3
P0	0	1	0
P1	2	0	0
P2	3	0	2
P3	2	1	1
P4	0	0	2

3.lépés: Kiszámítjuk az igényeket, amit a (MAX.IGÉNY-FOGLAL)-ból kapunk:

P0: $(7-0=7 ; 5-1=4 ; 3-0=3)$

P1: $(3-2=1 ; 2-0=2 ; 2-0=2)$

P2: $(9-3=6 ; 0-0=0 ; 2-2=0)$

P3: $(2-2=0 ; 2-1=1 ; 2-1=1)$

P4: $(4-0=4 ; 3-0=3 ; 3-2=1)$

3.lépés	IGÉNY = MAX.IGÉNY-FOGLAL		
	R1	R2	R3
P0	7	4	3
P1	1	2	2
P2	6	0	0
P3	0	1	1
P4	4	3	1

4.lépés: Kiszámoljuk a szabad erőforrások számát

Összeadjuk az erőforrásokat:

R1 ($0+2+3+2+0=7$)

R2 ($1+0+0+1+0=2$)

R3 ($0+0+2+1+2 = 5$)

FOGLAL		
R1	R2	R3
0	1	0
2	0	0
3	0	2
2	1	1
0	0	2
7	2	5

Kivonjuk a foglalt erőforrásokat az összes erőforrásból:

R1 erőforrások száma: $10-7 = 3$
R2 erőforrások száma: $5-2 = 3$
R3 erőforrások száma: $7-5 = 2$

4.lépés	
Szabad erőforrások száma:	
KÉSZLET:	(3,3,2)

5.lépés: Megnézzük melyik processz igényét tudjuk kielégíteni a szabad erőforrásunkkal.

P1	1	2	2
----	---	---	---

Utána ezt elindítjuk, és ha lefutott akkor a **FOGLAL** táblázat **P1** sorában lévő értékeket hozzáadjuk a szabad erőforrások számához.

Szabad erőforrások száma = $3+2=5$; $3+0=3$; $2+0=2$

5.lépés		Aktuális készlet	Igény	Felszabadult készlet
P1 kielégíthető:		(3,3,2)	(1,2,2)	(5,3,2)

Ezután az újonnan bővült szabad erőforrásokkal keressük a következő futtatható programot, és visszatérünk az 5. lépésre.

További futtatások:

5.lépés		Aktuális készlet	Igény	Felszabadult készlet
P1 kielégíthető:		(3,3,2)	(1,2,2)	(5,3,2)
P3 kielégíthető:		(5,3,2)	(0,1,1)	(7,4,3)
P0 kielégíthető:		(7,4,3)	(7,4,3)	(7,5,3)
P2 kielégíthető:		(7,5,3)	(6,0,0)	(10,5,5)
P4 kielégíthető:		(10,5,5)	(4,3,1)	(10,5,7)

Végül a biztonságos futáshoz szükséges végrehajtási sorrend:

Végrehajtási sorrend:
P1-P3-P0-P2-P4
Biztonságos rendszer

Kérdés: Kielégíthető-e P4 (3,3,0):

1-2.lépés:

P4(3,3,0)													
1.lépés					2.lépés								
				MAX.IGÉNY							FOGLAL		
				R1	R2	R3							
P0				7	5	3	P0				0	1	0
P1				3	2	2	P1				2	0	0
P2				9	0	2	P2				3	0	2
P3				2	2	2	P3				2	1	1
P4				4	3	3	P4				0+3=3	0+3=3	2

3.lépés:

3.lépés			
	IGÉNY = MAX.IGÉNY-FOGLAL		
	R1	R2	R3
P0	7	4	3
P1	1	2	2
P2	6	0	0
P3	0	1	1
P4	1	0	1

4.lépés:

2.lépés				
	FOGLAL			
	R1	R2	R3	
P0	0	1	0	R1 erőforrások száma: $10-10 = 0$
P1	2	0	0	
P2	3	0	2	R2 erőforrások száma: $5-5 = 0$
P3	2	1	1	
P4	$0+3=3$	$0+3=3$	2	R3 erőforrások száma: $7-5 = 2$
	10	5	5	

4.lépés				
Szabad erőforrások száma:				
KÉSZLET:	(0,0,2)			
Nem biztonságos a rendszer, mert a készletből egy processz sem tud lefutni.				

Nem biztonságos így a rendszer!

Kérdés: Kielégíthető-e P0 (0,2,0)

1-2.lépés:

P0(0,2,0)							
1.lépés				2.lépés			
	MAX.IGÉNY				FOGLAL		
	R1	R2	R3		R1	R2	R3
P0	7	5	3	P0	0	1+2=3	0
P1	3	2	2	P1	2	0	0
P2	9	0	2	P2	3	0	2
P3	2	2	2	P3	2	1	1
P4	4	3	3	P4	0	0	2

3.lépés:

3.lépés			
	IGÉNY = MAX.IGÉNY-FOGLAL		
	R1	R2	R3
P0	7	2	3
P1	1	2	2
P2	6	0	0
P3	0	1	1
P4	4	3	1

4.lépés:

2.lépés					
	FOGLAL				
	R1	R2	R3		
P0	0	1+2=3	0	R1 erőforrások száma: 10-7 = 3	
P1	2	0	0		
P2	3	0	2	R2 erőforrások száma: 5-4 = 1	
P3	2	1	1		
P4	0	0	2	R3 erőforrások száma: 7-5 = 2	
	7	4	5		

4.lépés	
Szabad erőforrások száma:	
KÉSZLET:	(3,1,2)

5.lépés: Processz futtatások:

5.lépés		Aktuális készlet	Igény	Felszabadult készlet
P3 kielégíthető		3,1,2	0,1,1	5,2,3
P1 kielégíthető		5,2,3	1,2,2	7,2,3
P0 kielégíthető		7,2,3	7,2,3	7,5,3
P2 kielégíthető		7,5,3	6,0,0	10,5,5
P4 kielégíthető		10,5,5	4,3,1	10,5,7

Végrehajtási sorrend:

Végrehajtási sorrend:
P3-P1-P0-P2-P4
Biztonságos rendszer