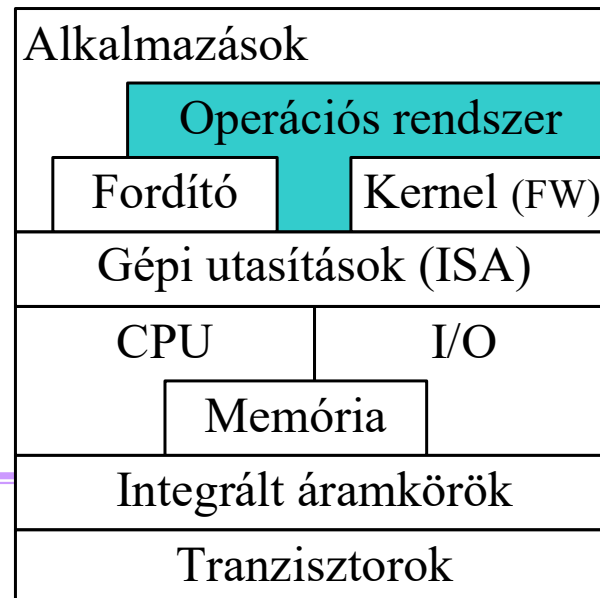


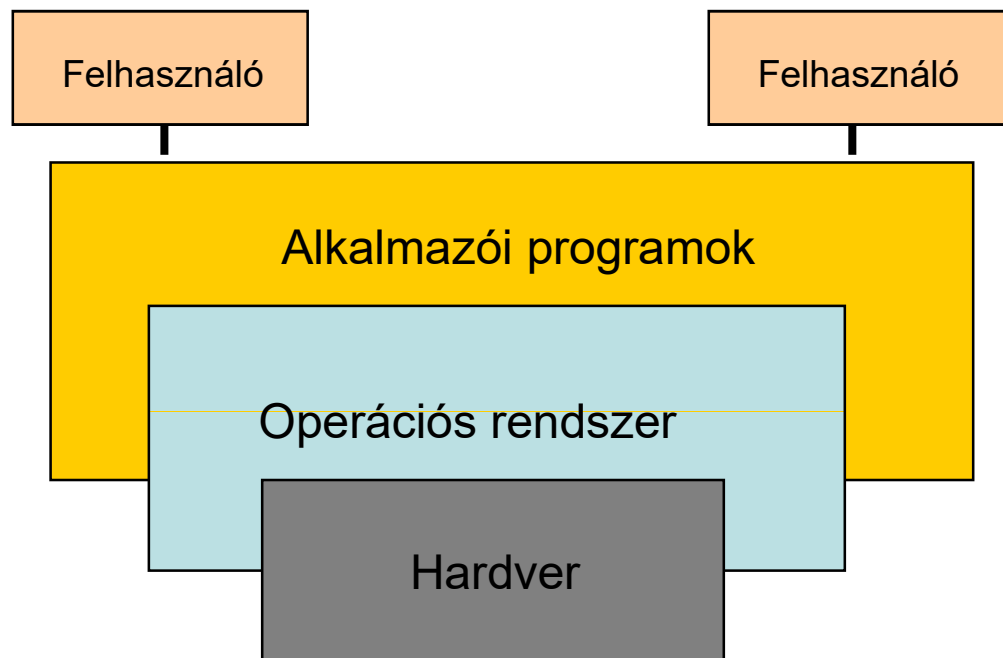
INFORMATIKA I.

BMEVIIIAB04

Operációs rendszerek



Számítógép rendszer



Operációs rendszer
kapcsolat a hardver és a
felhasználó között

Cél
Hatékony hardver kihasználás
A felhasználó kényelme



Operációs rendszerek

Rendszer

Erőforrások és az értük versengő folyamatok

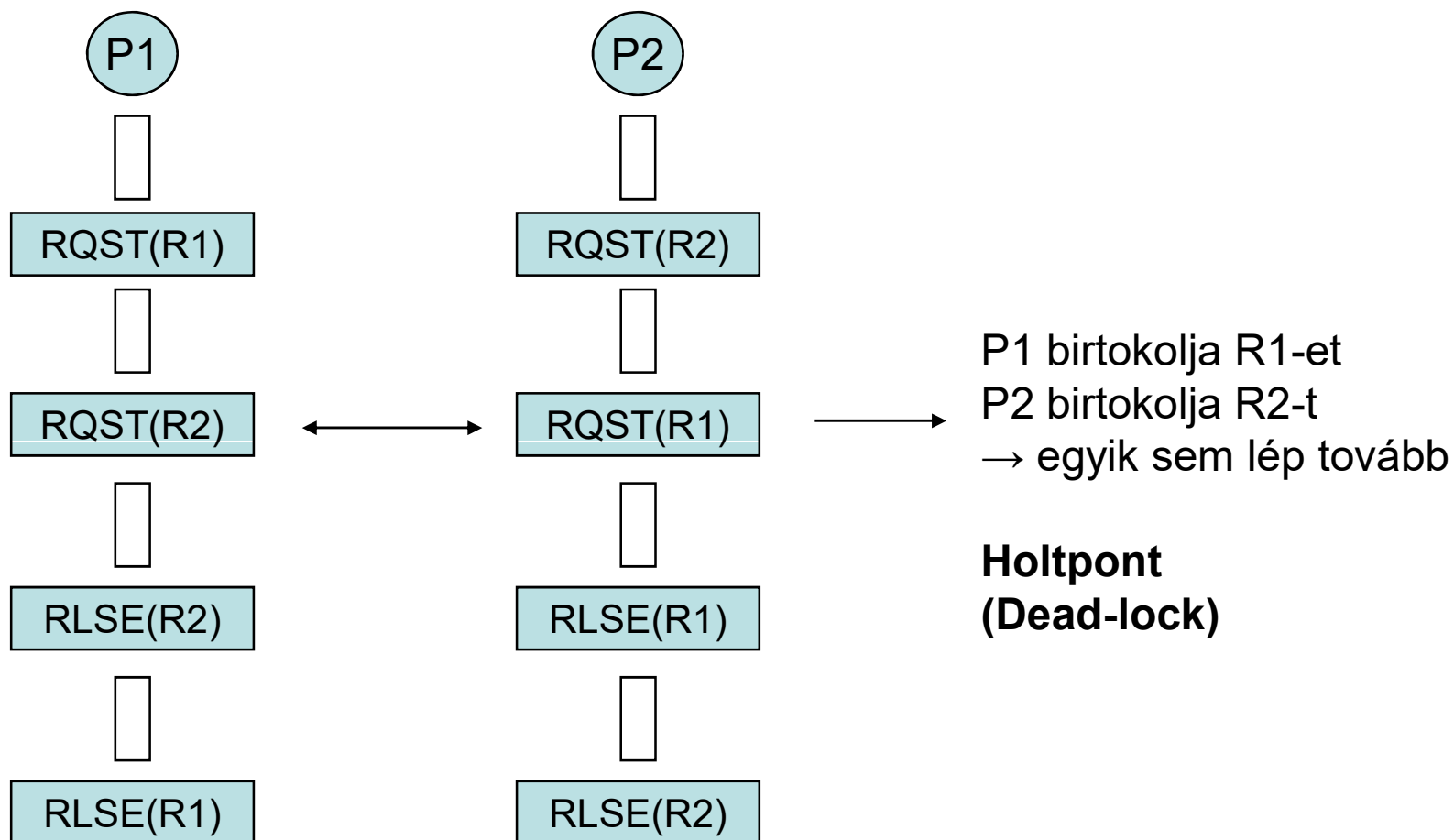
Erőforrás

Eszköz amelyet egy időben korlátozott számú folyamat használhat
(CPU, memória, periféria, fájl)

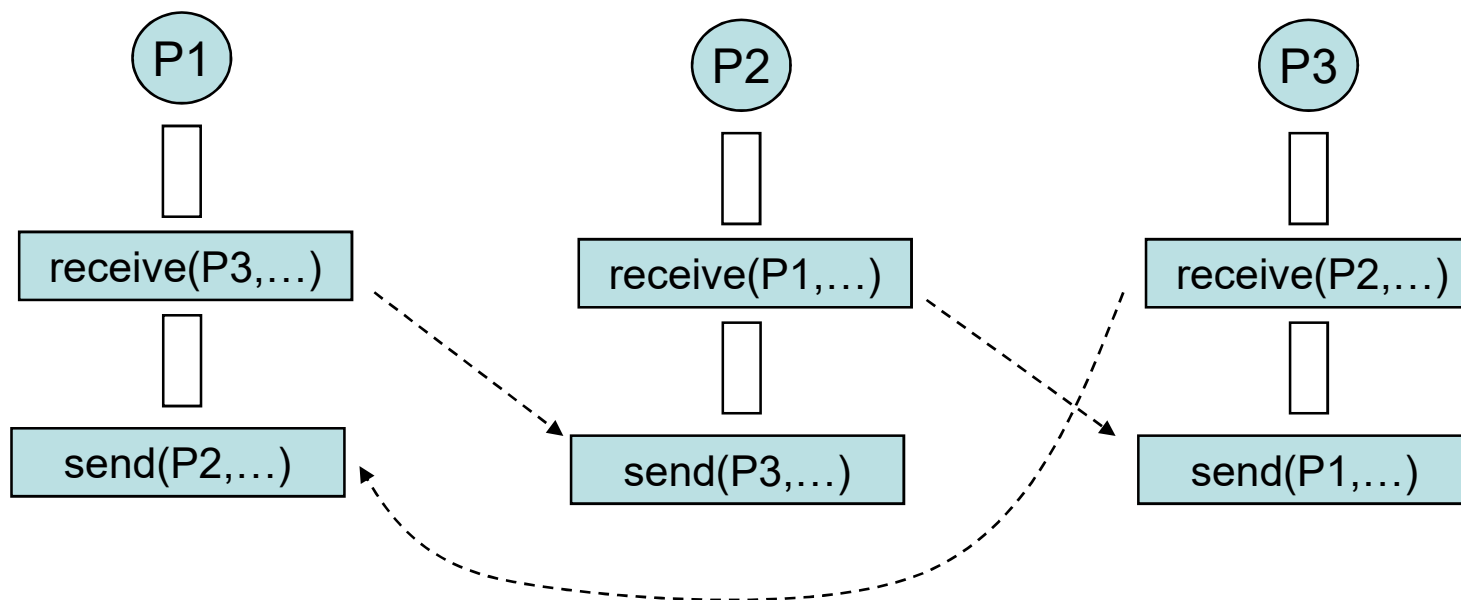
Műveletek

Kérés (REQUEST)

Felszabadítás (RELEASE)



Kommunikációs holtpont





Operációs rendszerek

Holtpont

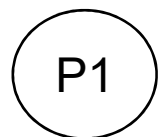
A folyamatok egy halmaza holtponton van, ha minden halmazbeli folyamat olyan eseményre várakozik, amit csak egy halmazbeli folyamat tud előidézni

Kialakulásának feltételei

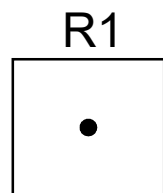
- Kölcsönös kizárás (mutual exclusion)
- Foglalva várakozás (Hold and wait)
- Nincs erőszakos elvétel (No preemption)
- Körkörös várakozás (circular wait)

Pillanatnyi állapot modellezése

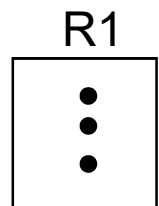
Erőforrás foglalási gráf



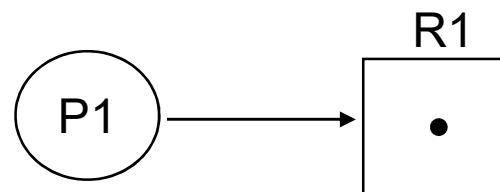
Folyamat



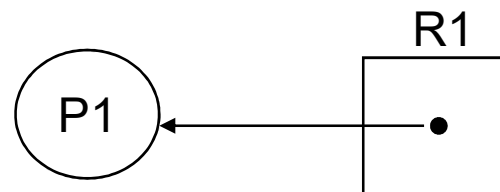
Erőforrás
(egy példány)



Erőforrás
(három példány)



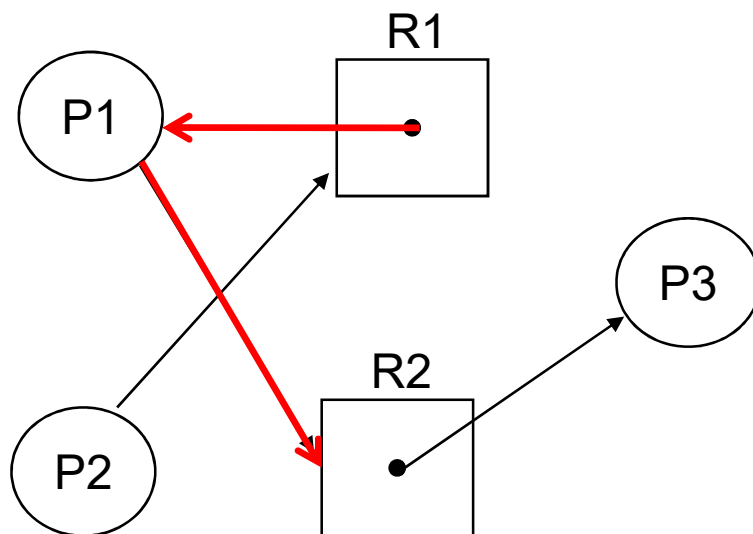
P1 folyamat vár R1 erőforrásra



P1 folyamat birtokolja R1 erőforrást

Pillanatnyi állapot modellezése

Erőforrás foglalási gráf



P1 foglalva vár

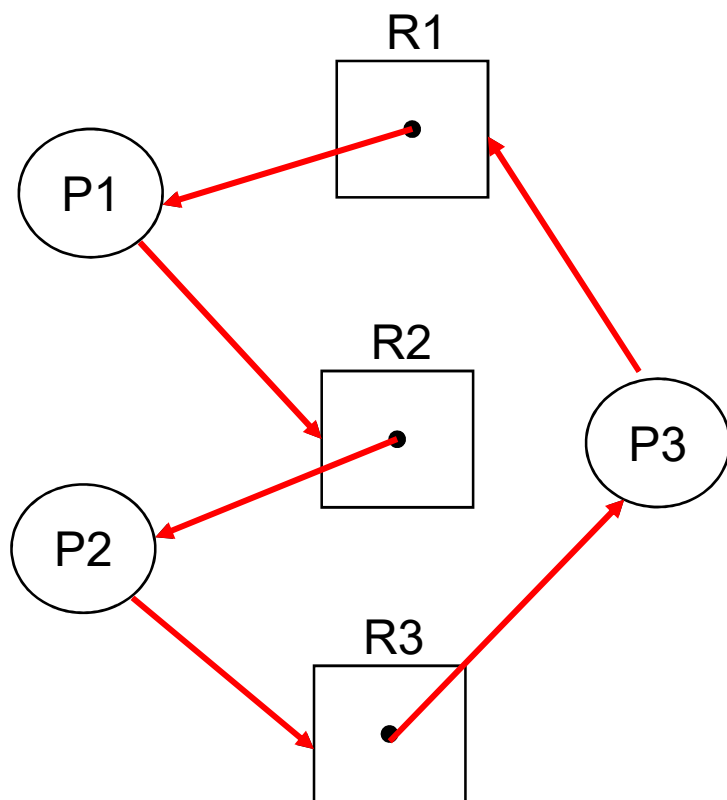
Ha P3 felszabadítja R2-t, P1 továbblép

Ha P1 felszabadítja R1, R2-t P2 továbblép

Nincs holtpont

Pillanatnyi állapot modellezése

Erőforrás foglalási gráf



P1, P2, P3 foglalva vár

Hurok van a gráfban

P1 – P2-re

P2 – P3-ra

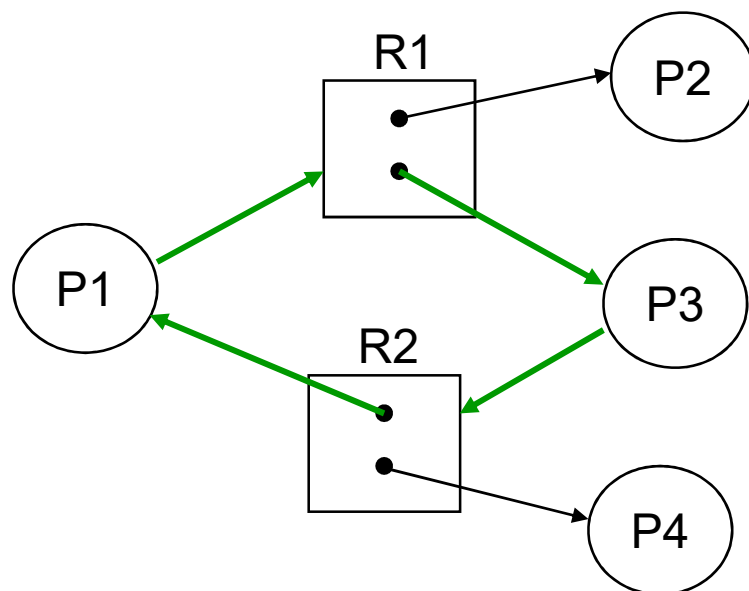
P3 – P1-re

vár

Holtpont

Pillanatnyi állapot modellezése

Erőforrás foglalási gráf



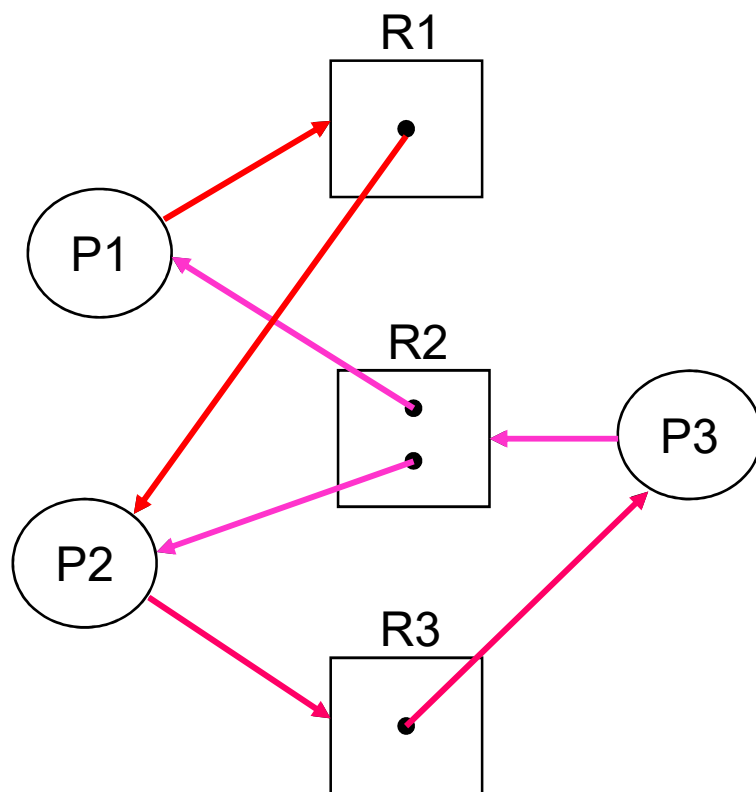
P1, P3 foglalta vár

Ha P4 felszabadítja R2-t, P3 továbblép
Ha P2 felszabadítja R1-t, P1 továbblép

Nincs holtpon

Pillanatnyi állapot modellezése

Erőforrás foglalási gráf



P1, P2, P3 foglalva vár

P1 – P2-re vár

P2 – P3-ra vár

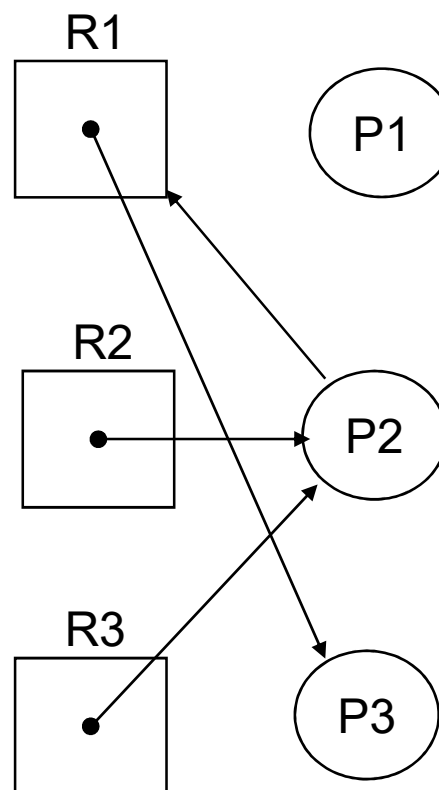
P3 – P1, P2-re vár

Holtpont

Holtpont analízis

	P1	P2	P3
A			
B	RQ (R1)	RQ (R3)	RQ (R3)
C			
D	RL (R1)	RQ (R2)	RQ (R1)
E			
F	RQ (R3)	RQ (R1)	RL (R3)
G			
H	RQ (R2)	RL (R1)	RQ (R2)
I			...
J	RL (R3)	RL (R3)	RL (R1)
K	RL (R2)	RL (R2)	...
L	...	...	RL (R2)

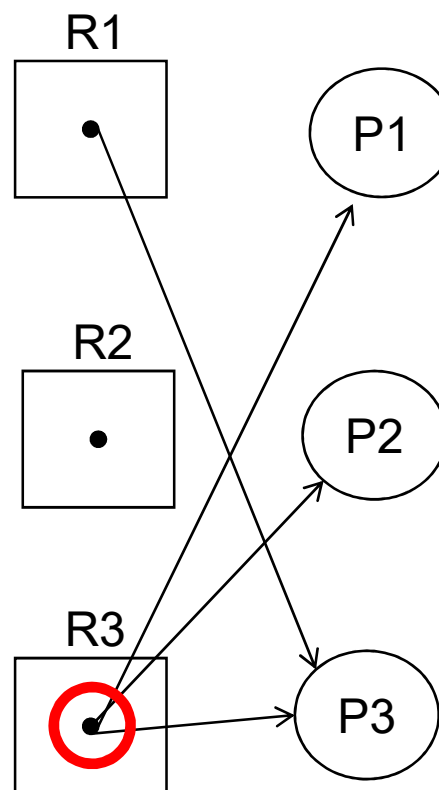
Rajzoljuk fel az erőforrás foglalási gráfot, ha a rendszer { P1:E, P2:F; P3:G} állapotban van



Holtpont analízis

	P1	P2	P3
A			
B	RQ (R1)	RQ (R3)	RQ (R3)
C			
D	RL (R1)	RQ (R2)	RQ (R1)
E			
F	RQ (R3)	RQ (R1)	RL (R3)
G			
H	RQ (R2)	RL (R1)	RQ (R2)
I			...
J	RL (R3)	RL (R3)	RL (R1)
K	RL (R2)	RL (R2)	...
L	...	...	RL (R2)

Eljuthat-e a rendszer { P1:**G**, P2:**C**; P3:**E**} állapotba?





Holtpont kezelése

1. Nem foglalkozunk a holtponttal

Strucc algoritmus

2. Védekezünk a kialakulása ellen

Megelőzés

Elkerülés

3. Észrevesszük és megszüntetjük

Feloldás

Holtpont kezelése - Megelőzés

Az egyik szükséges feltételt kiiktatjuk

(1) Kölcsönös kizárás

→ Nem használunk kölcsönös kizárást igénylő erőforrásokat ☹

(2) Foglalva várakozás

→ Egyszerre foglalunk le minden szükséges erőforrást

→ Mielőtt foglalunk, mindent elengedünk

(3) Nincs erőszakos elvétel

→ Van – de kitől veszünk el?

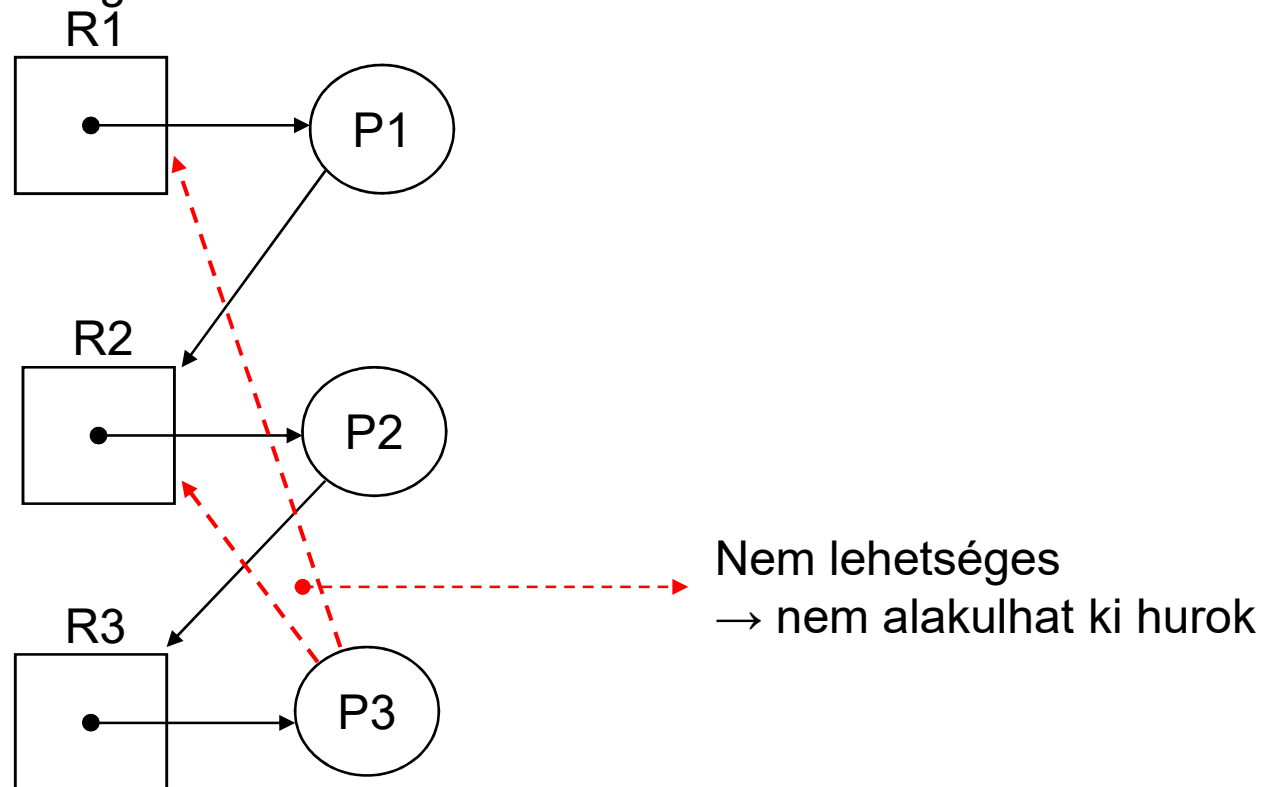
- mindent attól aki kér

- másoktól (mindent megkap aki kér)

Holtpont kezelése - Megelőzés

(4) Körkörös várakozás

→ sorszámozzuk az erőforrásokat, lefoglalás csak a sorszámok növekvő sorrendjében lehetséges





Operációs rendszerek

Holtpont kezelése - Elkerülés

Csak olyan igényt elégítünk ki, ami nem vezet holtpontra

Biztonságos állapot:

létezik olyan lefutási sorrendje a folyamatoknak, ahol minden folyamat lefut

Apriori ismeretekre van szükség

→ tudnunk kell, hogy egy folyamat a futása során mely erőforrásból mennyit fog használni

A kiértékelés időt vesz igénybe

Holtpont kezelése - Elkerülés

R: 10 példányos erőforrás
P2, P3 kér 1-1 példányt
Mit tegyünk?

	Maximális igény	Aktuálisan birtokolt	Kérés
P1	8	4	
P2	3	2	1
P3	9	2	1
Szabad		2	

Adjuk oda az erőforrásokat

	Maximális igény	Birtokol	Még kérhet
P1	8	4	4
P2	3	3	0
P3	9	3	6
Szabad		0	

P2 futhat
Ha lefut, felszabadul: 3

	Maximális igény	Birtokol	Még kérhet
P1	8	4	4
P2	3	0	0
P3	9	3	6
Szabad		3	

Nem elegendő hogy
P1 vagy P3 lefusson

Nem biztonságos állapotba kerülünk

Holtpont kezelése - Elkerülés

R: 10 példányos erőforrás
Mit tegyünk?

Csak P2-nek adjuk oda az erőforrást

	Maximális igény	Aktuálisan birtokolt	Kérés
P1	8	4	
P2	3	2	1
P3	9	2	1
Szabad		2	

	Maximális igény	Aktuálisan birtokolt	Kérés	Birtokol	Még kérhet	Futhat
P1	8	4		4	4	
P2	3	2	1	3	0	x
P3	9	2	1	2	7	
Szabad		2		1		

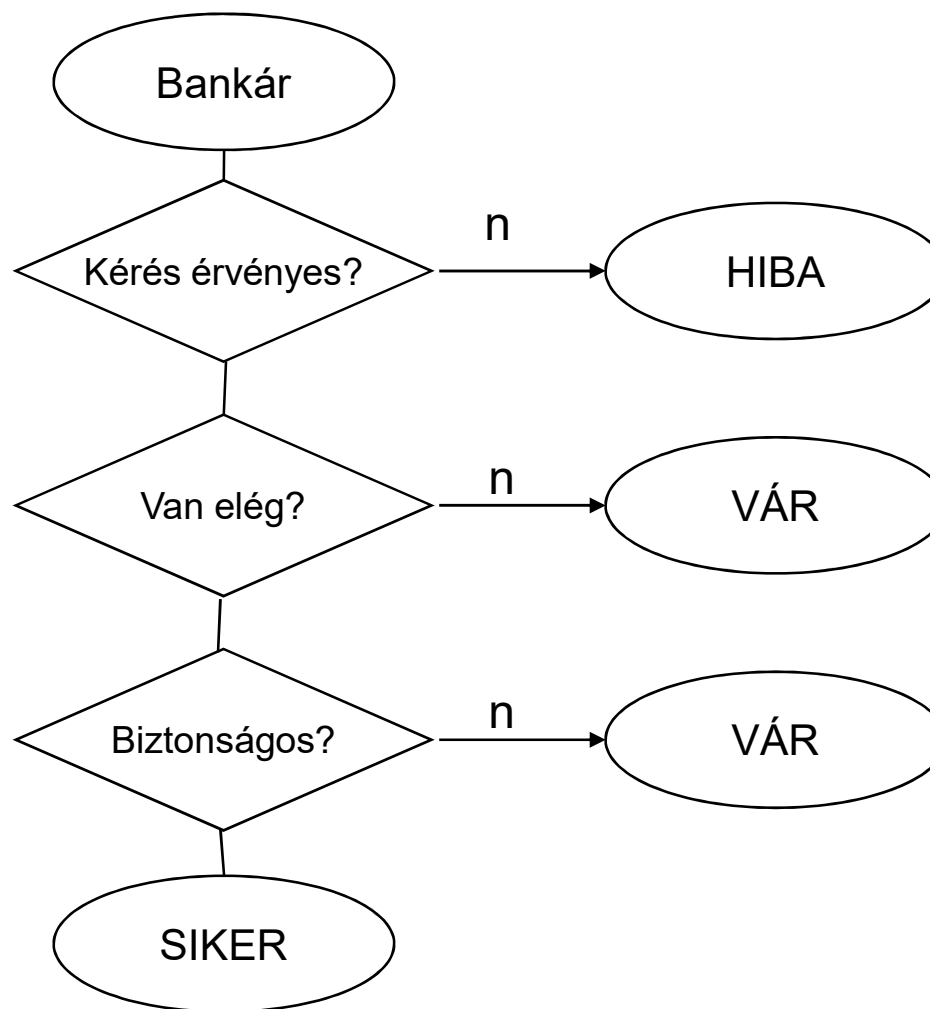
P2 futhat
Ha lefut, felszabadul: 3

	Maximális igény	Aktuálisan birtokolt	Kérés	Birtokol	Még kérhet	Futhat
P1	8	4		4	4	x
P2	3			0	0	
P3	9	2	1	2	7	
Szabad		2		4		

Rendelkezésre áll :4
Elegendő hogy P1 lefusson

P2,P1,P3 sorrendben lefuthatnak
Biztonságos

Holtpont kezelése - Bankár algoritmus



Holtpont kezelése - Bankár algoritmus

N – a folyamatok száma

`max[1..N]` – egész // apriori maximális igények

`bir[1..N]` – egész // aktuálisan birtokolt erőforrások

`szabad` – egész // a még szabad példányok száma

`Bankár( kér: egész, PID: folyamat azonosító)`

```
if kér > (max[PID] - bir[PID]) then return(HIBA) end_if
```

```
if kér > szabad then return( wait(PID) ) end_if
```

```
bir[PID] = szabad + kér
```

```
szabad = szabad - kér
```

```
if nem Biztonságos() then
```

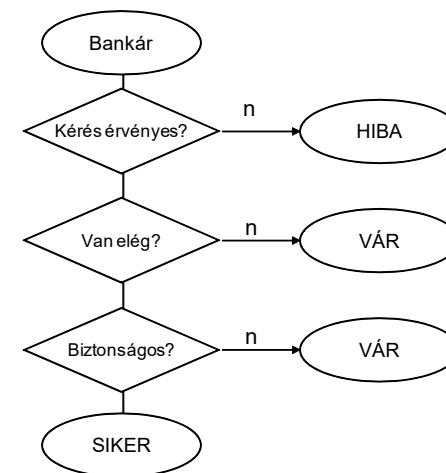
```
    bir[PID] = bir[PID] - kér
```

```
    szabad = szabad + kér
```

```
    return( wait(PID) )
```

```
end_if
```

```
return( SIKER )
```





Operációs rendszerek

Holtpont kezelése - Bankár algoritmus

Biztonságos()

Keresünk olyan folyamatot amelyiket

- még nem vizsgáltunk
- a rendelkezésre álló készlettel tud futni
 - Ha találtunk: adjuk vissza az általa foglalt erőforrásokat (mintha lefutott volna) és jegyezzük meg hogy futhat és kezdjük előlről a keresést
 - Ha nem találunk: vége

Ha mindenki futhat: az állapot biztonságos



Operációs rendszerek

Holtpont kezelése - Bankár algoritmus

Az algoritmus kiterjesztése több erőforrás típusra

max: $n \times m$ mátrix (n : folyamatok, m : erőforrások)

bir: $n \times m$ mátrix

szabad: m elemű vektor

$<$, $>$, $=$ operátorok: a vektorok elemenkénti komparálása

Bonyolult, nagy overhead

Holtpont kezelése - Bankár algoritmus

Egy rendszer az alábbi erőforrásokkal gazdálkodik:

R1: **10** példány, R2: **9** példány, R3: **12** példány

A rendszerben jelenlévő folyamatok: P1, P2, P3, P4.

Az alábbi helyzetben **P1** folyamat R1,R2,R3 **(1,2,2)** példányt kér.

Az erőforrás-kiosztó Bankár-algoritmust használ a holtpont elkerülésére.

Kiszolgálja-e a kérést, vagy várakoztatja?

	Maximális igény			Aktuális foglалás		
	R1	R2	R3	R1	R2	R3
P1	4	4	5	2	2	3
P2	1	4	3	1	2	2
P3	6	7	7	0	1	3
P4	3	7	10	2	1	2

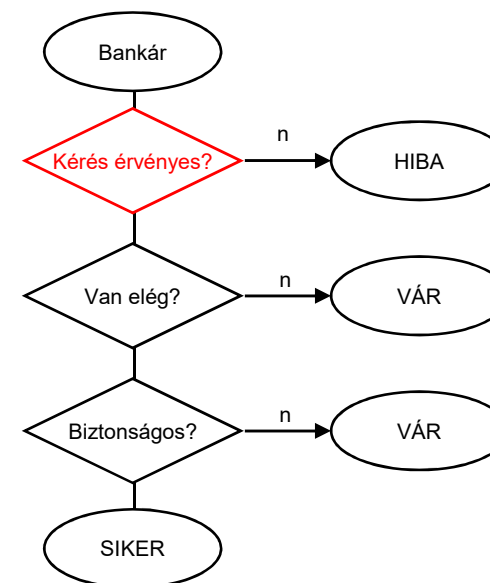
Holtpont kezelése - Bankár algoritmus

R1: **10** példány, R2: **9** példány, R3: **12** példány

Kérés: **P1** folyamat R1,R2,R3 (**1,2,2**)

Kérés érvényes? - Igen

	Maximális igény			Aktuális foglалás			Még kérhet		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	4	4	5	2	2	3	2	2	2
P2	1	4	3	1	2	2	0	2	1
P3	6	7	7	0	1	3	6	6	4
P4	3	7	10	2	1	2	1	6	8
szabad				5	3	2			



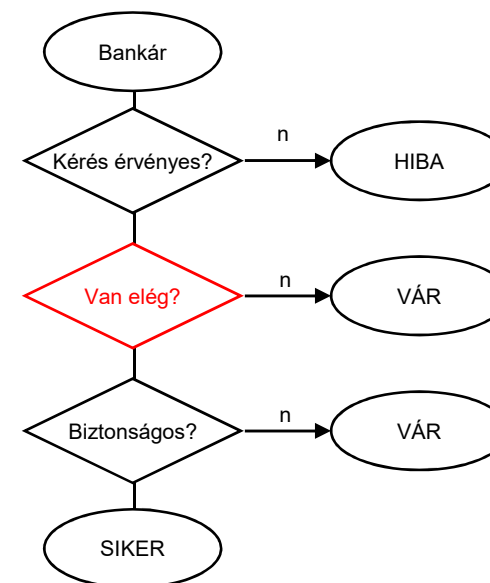
Holtpont kezelése - Bankár algoritmus

R1: **10** példány, R2: **9** példány, R3: **12** példány

Kérés: P1 folyamat R1,R2,R3 (**1,2,2**)

Van elég erőforrás? - Van

	Maximális igény			Aktuális foglalás			Még kérhet		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	4	4	5	2	2	3	2	2	2
P2	1	4	3	1	2	2	0	2	1
P3	6	7	7	0	1	3	6	6	4
P4	3	7	10	2	1	2	1	6	8
szabad				5	3	2			



Holtpont kezelése - Bankár algoritmus

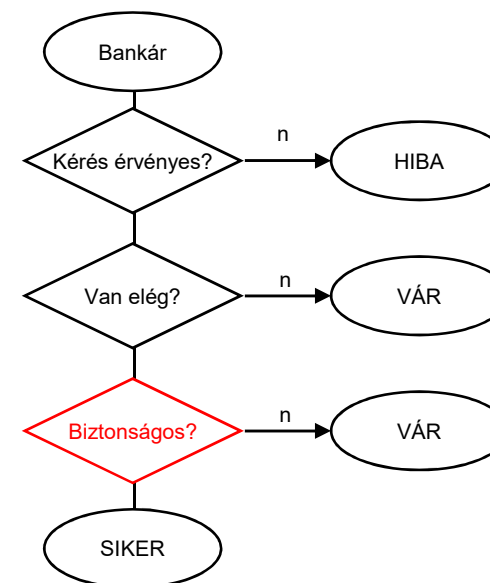
R1: **10** példány, R2: **9** példány, R3: **12** példány

Kérés: **P1** folyamat R1,R2,R3 (**1,2,2**)

Biztonságos? - adjuk oda az erőforrásokat

Futhat

	Maximális igény			Aktuális foglалás			Még kérhet		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	4	4	5	3	4	5	1	0	0
P2	1	4	3	1	2	2	0	2	1
P3	6	7	7	0	1	3	6	6	4
P4	3	7	10	2	1	2	1	6	8
szabad				4	1	0			



Holtpont kezelése - Bankár algoritmus

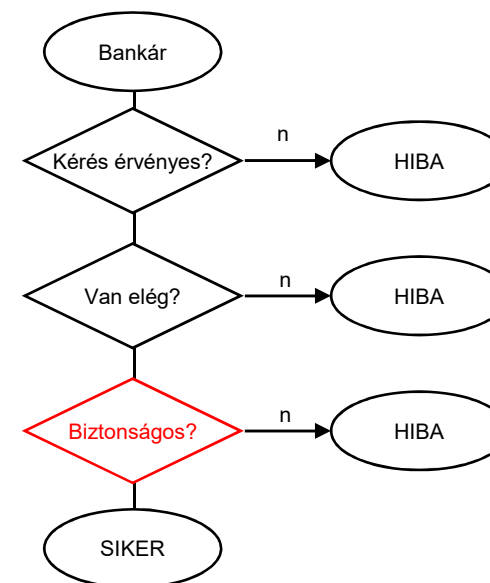
R1: **10** példány, R2: **9** példány, R3: **12** példány

Kérés: **P1** folyamat R1,R2,R3 (**1,2,2**)

Futtassuk le P1-et

Futhat

	Maximális igény			Aktuális foglалás			Még kérhet		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	4	4	5	0	0	0	0	0	0
P2	1	4	3	1	2	2	0	2	1
P3	6	7	7	0	1	3	6	6	4
P4	3	7	10	2	1	2	1	6	8
szabad				7	5	5			



Holtpont kezelése - Bankár algoritmus

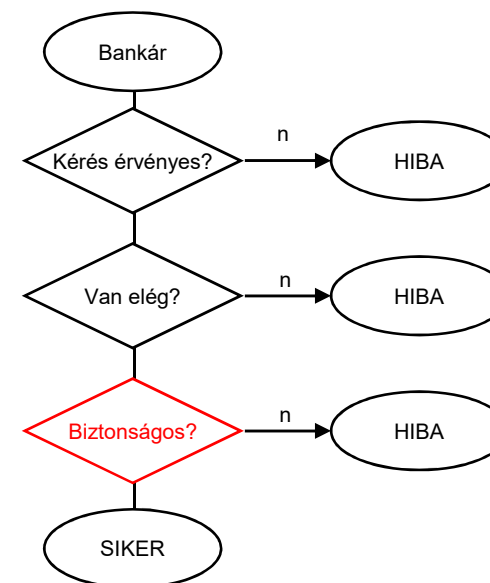
R1: **10** példány, R2: **9** példány, R3: **12** példány

Kérés: **P1** folyamat R1,R2,R3 (**1,2,2**)

Futtassuk le P2-t

Futhat

	Maximális igény			Aktuális foglалás			Még kérhet		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	4	4	5	0	0	0	0	0	0
P2	1	4	3	0	0	0	0	0	0
P3	6	7	7	0	1	3	6	6	4
P4	3	7	10	2	1	2	1	6	8
szabad				8	7	7			



Holtpont kezelése - Bankár algoritmus

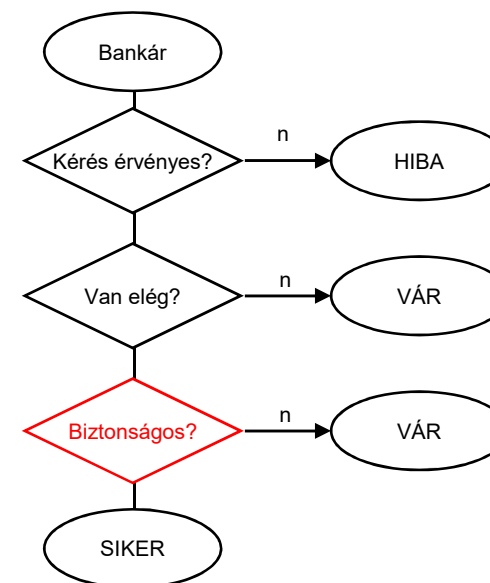
R1: **10** példány, R2: **9** példány, R3: **12** példány

Kérés: **P1** folyamat R1,R2,R3 (**1,2,2**)

Futtassuk le P3-t

	Maximális igény			Aktuális foglalás			Még kérhet		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	4	4	5	0	0	0	0	0	0
P2	1	4	3	0	0	0	0	0	0
P3	6	7	7	0	0	0	0	0	0
P4	3	7	10	2	1	2	1	6	8
szabad				8	8	10			

P4 le tud futni : **a** kérés kiszolgálható



Holtpont kezelése - Bankár algoritmus

Egy rendszer az alábbi erőforrásokkal gazdálkodik:

R1: **8** példány, R2: **9** példány, R3: **13** példány

A rendszerben jelenlévő folyamatok: P1, P2, P3, P4.

Az alábbi helyzetben **P2** folyamat R1,R2,R3 **(1,2,2)** példányt kér.

Az erőforrás-kiosztó Bankár-algoritmust használ a holtpont elkerülésére.

Kiszolgálja-e a kérést, vagy várakoztatja?

	Maximális igény			Aktuális foglалás		
	R1	R2	R3	R1	R2	R3
P1	4	2	5	2	2	3
P2	7	7	7	0	1	1
P3	1	4	3	1	2	2
P4	3	7	4	2	1	2

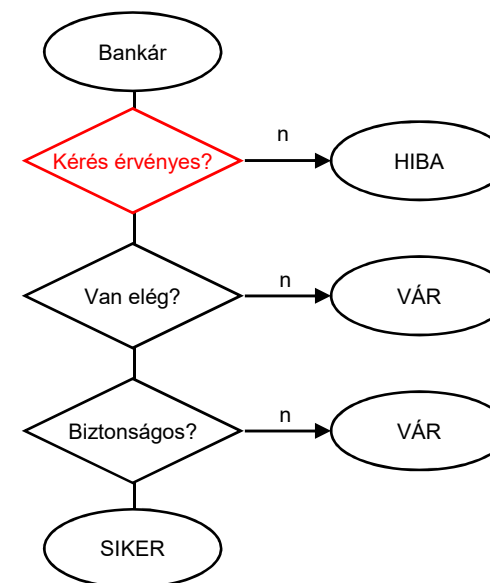
Holtpont kezelése - Bankár algoritmus

R1: 8 példány, R2: 9 példány, R3: 13 példány

Kérés: P2 folyamat R1,R2,R3 (1,2,2)

Kérés érvényes? - Igen

	Maximális igény			Aktuális foglалás			Még kérhet		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	4	2	5	2	2	3	2	0	2
P2	7	7	7	0	1	1	7	6	6
P3	1	4	3	1	2	2	0	2	1
P4	3	7	4	2	1	2	1	6	2
szabad				3	3	5			



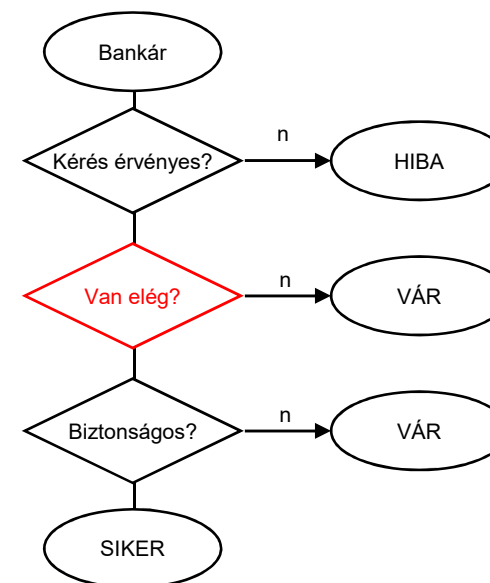
Holtpont kezelése - Bankár algoritmus

R1: 8 példány, R2: 9 példány, R3: 13 példány

Kérés: P2 folyamat R1,R2,R3 (1,2,2)

Van elég erőforrás? - Van

	Maximális igény			Aktuális foglalás			Még kérhet		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	4	2	5	2	2	3	2	0	2
P2	7	7	7	0	1	1	7	6	6
P3	1	4	3	1	2	2	0	2	1
P4	3	7	4	2	1	2	1	6	2
szabad				3	3	5			



Holtpont kezelése - Bankár algoritmus

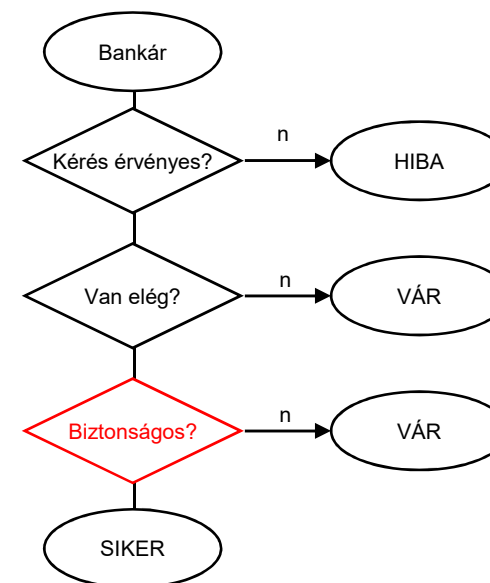
R1: 8 példány, R2: 9 példány, R3: 13 példány

Kérés: P2 folyamat R1,R2,R3 (1,2,2)

Biztonságos? - adjuk oda az erőforrásokat

Futhat

	Maximális igény			Aktuális foglалás			Még kérhet		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	4	2	5	2	2	3	2	0	2
P2	7	7	7	1	3	3	6	4	4
P3	1	4	3	1	2	2	0	2	1
P4	3	7	4	2	1	2	1	6	2
szabad				2	1	3			



Holtpont kezelése - Bankár algoritmus

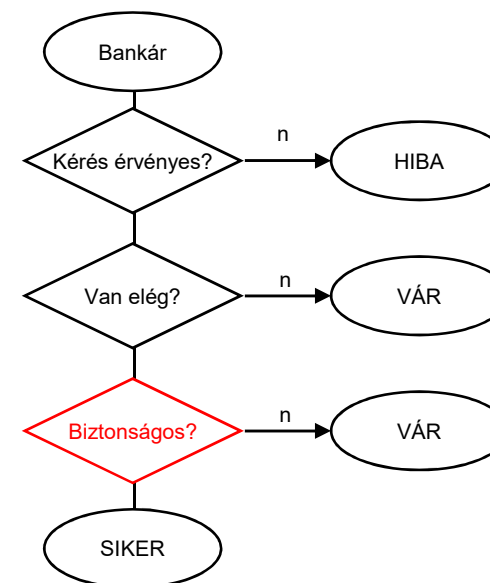
R1: 8 példány, R2: 9 példány, R3: 13 példány

Kérés: P2 folyamat R1,R2,R3 (1,2,2)

Futtassuk le P1-et

Futhat

	Maximális igény			Aktuális foglalás			Még kérhet		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	4	2	5	0	0	0	0	0	0
P2	7	7	7	1	3	3	6	4	4
P3	1	4	3	1	2	2	0	2	1
P4	3	7	4	2	1	2	1	6	2
szabad				4	3	6			



Holtpont kezelése - Bankár algoritmus

R1: 8 példány, R2: 9 példány, R3: 13 példány

Kérés: P2 folyamat R1,R2,R3 (1,2,2)

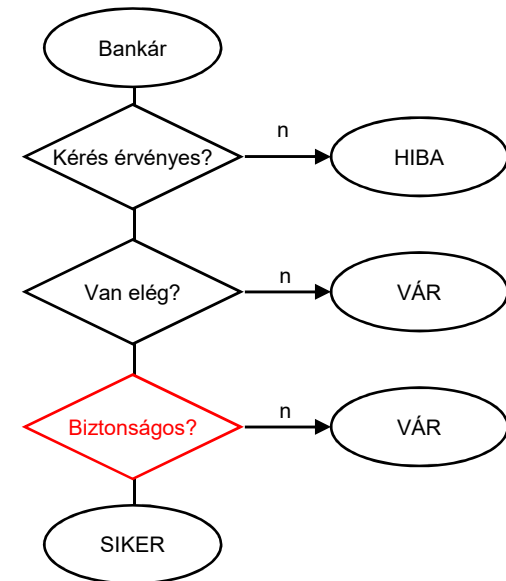
Futtassuk le P3-t

	Maximális igény			Aktuális foglалás			Még kérhet		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	4	2	5	0	0	0	0	0	0
P2	7	7	7	1	3	3	6	4	4
P3	1	4	3	0	0	0	0	0	0
P4	3	7	4	2	1	2	1	6	2
szabad				5	5	8			

A szabad mennyiség sem P2-nek (R1 erőforrás),
sem P4-nek (R2 erőforrás) nem elegendő

Az állapot nem biztonságos

A kérést várakoztatni kell

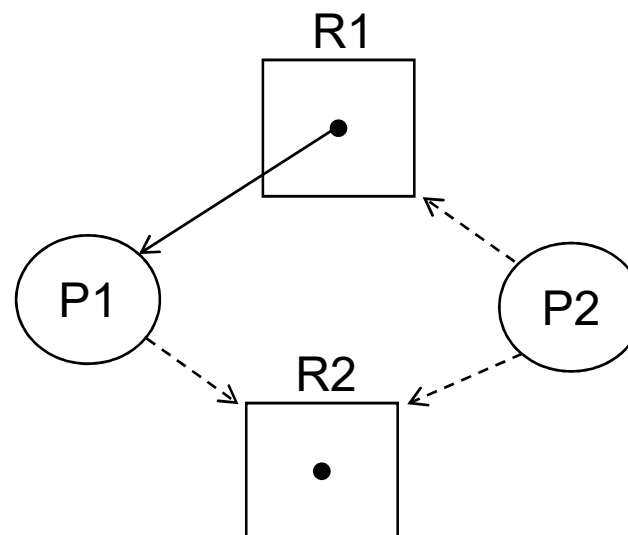
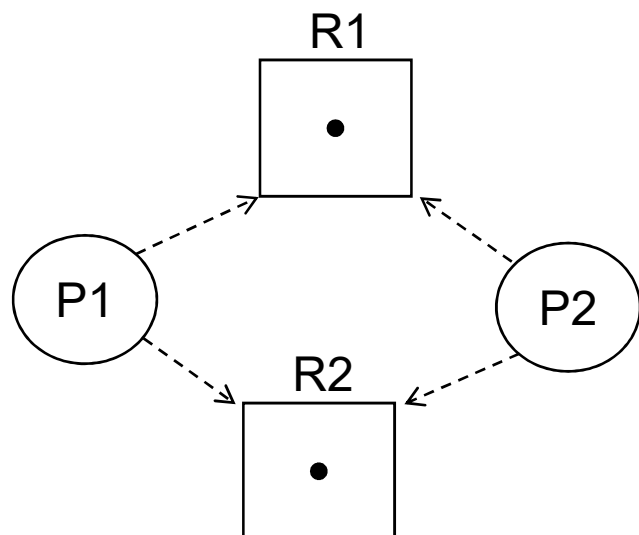


Holtpont kezelése - Egy példányos erőforrások

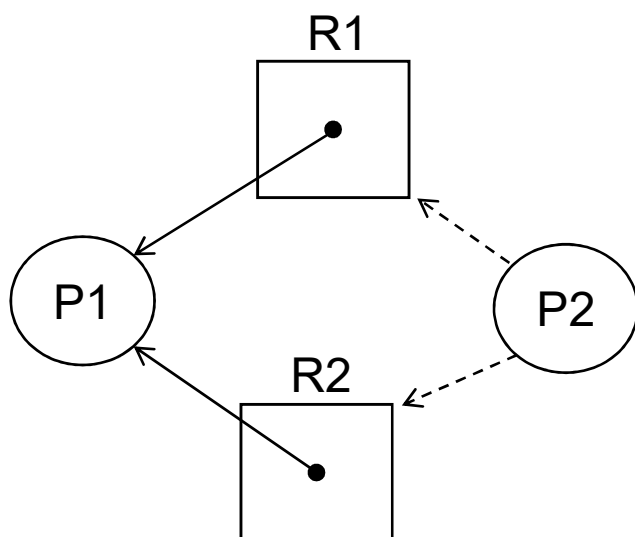
Erőforrás foglalási gráf: új él bevezetése az apriori ismeretek alapján

—————> birtokol
 - - - - -> kérhet

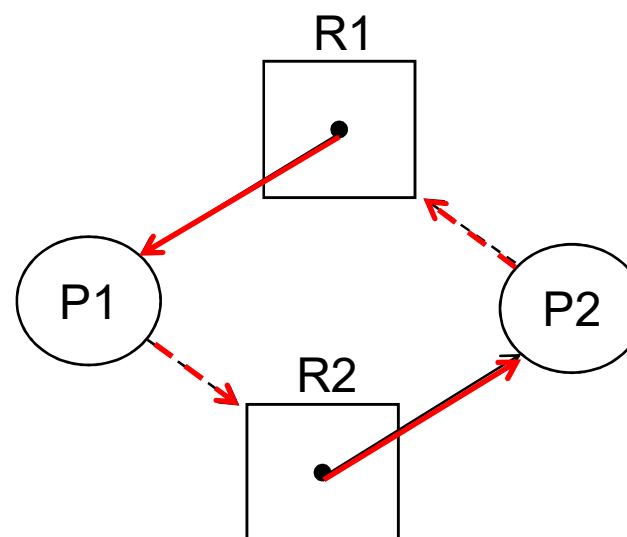
Biztonságos, ha nincs hurok



Holtpont kezelése - Egy példányos erőforrások



Biztonságos
(nincs hurok)



Nem biztonságos
(nem biztos, hogy
gond lesz, de lehet)

Holtpont kezelése - Észlelés és feloldás

Nincsenek apriori ismereteink

Hasonló a biztonságosság vizsgálatához

Ellenőrizzük, hogy a kérések teljesíthetők-e

Ha valamelyik igen, a folyamat lefut és felszabadít

Újra ellenőrizzük, hogy van-e olyan aki futhat

Ha valamelyik igen, a folyamat lefut és felszabadít

Ha mindenki le tudott futni: **nincs holtpont**

Akik nem: **holtponton vannak**

N=10

	Birtokol	Kér
P1	4	4
P2	1	0
P3	3	4
P4	1	2

szabad 1

P2 fut
lefutott

	Birtokol	Kér
P1	4	4
P2	0	0
P3	3	4
P4	1	2

szabad 2

P4 futhat
lefutott

	Birtokol	Kér
P1	4	4
P2	0	0
P3	3	4
P4	0	0

szabad 3

P1, P3 holtponton van

Holtpont kezelése - Észlelés (1)

Egy rendszer az alábbi erőforrásokkal gazdálkodik:

R1: **12** példány, R2: **10** példány, R3: **15**példány

A rendszerben jelenlévő folyamatok: P1, P2, P3, P4.

Vannak-e holtponton lévő folyamatok az alábbi helyzetben?

	Aktuális foglалás			Várakozó kérések		
	R1	R2	R3	R1	R2	R3
P1	2	2	3	0	3	1
P2	1	3	4	9	4	6
P3	3	2	4	0	0	0
P4	3	1	3	3	6	2

Holtpont kezelése - Észlelés (1)

Egy rendszer az alábbi erőforrásokkal gazdálkodik:

R1: **12** példány, R2: **10** példány, R3: **15**példány

A rendszerben jelenlévő folyamatok: P1, P2, P3, P4.

Vannak-e holtponton lévő folyamatok az alábbi helyzetben?

	Aktuális foglалás			Várakozó kérések		
	R1	R2	R3	R1	R2	R3
P1	2	2	3	0	3	1
P2	1	3	4	9	4	6
P3	3	2	4	0	0	0
P4	3	1	3	3	6	2

P3 fut

Felszabadíthat 3, 2, 4 példányt

Holtpont kezelése - Észlelés (1)

Egy rendszer az alábbi erőforrásokkal gazdálkodik:

R1: **12** példány, R2: **10** példány, R3: **15**példány

A rendszerben jelenlévő folyamatok: P1, P2, P3, P4.

Vannak-e holtponton lévő folyamatok az alábbi helyzetben?

P3 lefutott

	Aktuális foglалás			Várakozó kérések		
	R1	R2	R3	R1	R2	R3
P1	2	2	3	0	3	1
P2	1	3	4	9	4	6
P3	0	0	0	0	0	0
P4	3	1	3	3	6	2
szabad	6	4	5			

P1 futhat

Felszabadíthat 2, 2, 3 példányt

Holtpont kezelése - Észlelés (1)

Egy rendszer az alábbi erőforrásokkal gazdálkodik:

R1: **12** példány, R2: **10** példány, R3: **15**példány

A rendszerben jelenlévő folyamatok: P1, P2, P3, P4.

Vannak-e holtponton lévő folyamatok az alábbi helyzetben?

P1 lefutott

	Aktuális foglалás			Várakozó kérések		
	R1	R2	R3	R1	R2	R3
P1	0	0	0	0	0	0
P2	1	3	4	9	4	6
P3	0	0	0	0	0	0
P4	3	1	3	3	6	2
szabad	8	6	8			

P4 futhat

Felszabadíthat 3, 1, 3 példányt

Holtpont kezelése - Észlelés (1)

Egy rendszer az alábbi erőforrásokkal gazdálkodik:

R1: **12** példány, R2: **10** példány, R3: **15**példány

A rendszerben jelenlévő folyamatok: P1, P2, P3, P4.

Vannak-e holtponton lévő folyamatok az alábbi helyzetben?

P4 lefutott

	Aktuális foglалás			Várakozó kérések		
	R1	R2	R3	R1	R2	R3
P1	0	0	0	0	0	0
P2	1	3	4	9	4	6
P3	0	0	0	0	0	0
P4	0	0	0	0	0	0
szabad	11	7	11			

P2 kérése is teljesíthető

NINCS HOLTPONT

Holtpont kezelése - Észlelés (2)

Egy rendszer az alábbi erőforrásokkal gazdálkodik:

R1: **12** példány, R2: **10** példány, R3: **15**példány

A rendszerben jelenlévő folyamatok: P1, P2, P3, P4.

Vannak-e holtponton lévő folyamatok az alábbi helyzetben?

	Aktuális foglалás			Várakozó kérések		
	R1	R2	R3	R1	R2	R3
P1	1	3	4	9	4	6
P2	3	2	4	0	0	0
P3	3	1	3	3	7	2
P4	2	2	3	0	3	1

Holtpont kezelése - Észlelés (2)

Egy rendszer az alábbi erőforrásokkal gazdálkodik:

R1: **12** példány, R2: **10** példány, R3: **15**példány

A rendszerben jelenlévő folyamatok: P1, P2, P3, P4.

Vannak-e holtponton lévő folyamatok az alábbi helyzetben?

	Aktuális foglалás			Várakozó kérések		
	R1	R2	R3	R1	R2	R3
P1	1	3	4	9	4	6
P2	3	2	4	0	0	0
P3	3	1	3	3	7	2
P4	2	2	3	0	3	1
szabad	3	2	1			

P2 fut

Felszabadíthat 3, 2, 4 példányt

Holtpont kezelése - Észlelés (2)

Egy rendszer az alábbi erőforrásokkal gazdálkodik:

R1: **12** példány, R2: **10** példány, R3: **15**példány

A rendszerben jelenlévő folyamatok: P1, P2, P3, P4.

Vannak-e holtponton lévő folyamatok az alábbi helyzetben?

P2 lefutott

	Aktuális foglалás			Várakozó kérések		
	R1	R2	R3	R1	R2	R3
P1	1	3	4	9	4	6
P2	0	0	0	0	0	0
P3	3	1	3	3	7	2
P4	2	2	3	0	3	1
szabad	6	4	5			

P4 futhat

Felszabadíthat 2, 2, 3 példányt

Holtpont kezelése - Észlelés (2)

Egy rendszer az alábbi erőforrásokkal gazdálkodik:

R1: **12** példány, R2: **10** példány, R3: **15**példány

A rendszerben jelenlévő folyamatok: P1, P2, P3, P4.

Vannak-e holtponton lévő folyamatok az alábbi helyzetben?

P4 lefutott

	Aktuális foglалás			Várakozó kérések		
	R1	R2	R3	R1	R2	R3
P1	1	3	4	9	4	6
P2	0	0	0	0	0	0
P3	3	1	3	3	7	2
P4	0	0	0	0	0	0
szabad	8	6	8			

Nem elég P1-nek: R1

Nem elég P3-nak: R2

P1,P3 holtponton van



Operációs rendszerek

Holtpont kezelése - Észlelés és feloldás

Észrevettük – mit tegyünk

Radikális megoldás

Minden holtponton lévő folyamatot kilövünk

Gyors, de nagy veszteség lehet

Kíméletes megoldás

Döntünk, hogy kit lőjük ki (idő!)

Mi alapján döntsünk?

A lehető legkevesebb folyamatot lőjük ki

A prioritás befolyásolható

Minél kevesebb munka vesszen kárba

A folyamatokat fel kell készíteni a visszaállíthatóságra



Operációs rendszerek

Holtpont kezelése - Észlelés és feloldás

Észrevettük – mit tegyünk

Erőforrás elvétel

Elveszünk erőforrást és átadjuk egy másik folyamatnak

Kitől vegyünk el?

Akitől elvesszük, honnan folytassa a működést?

Mikor futtassuk az észlelést

Szemponatok:

Milyen gyakran számítunk rá?

Hány folyamat ragadhat be?

Elvileg minden kéreşkor futtatható, de ez hatalmas overhead