

Operációs rendszerek BSc

10. Gyak.

2022. 04. 12.

Készítette:

Tihor Fruzsina Bsc
Mérnökinformatikus
THDWDR

Miskolc, 2022

1.feladat: Az előadáson bemutatott mintaprogram alapján készítse el a következő feladatot.

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 P1 (1,0,2), P4 (3,3,0) ill. P0 (0,2,0) kérése úgy, hogy biztonságos legyen, holtpontmentesség szempontjából a rendszer - a következő *kiinduló állapot* alapján.

Külön-külön táblázatba oldja meg a feladatot!

- Határozza meg a processzek által igényelt erőforrások mátrixát?
- Határozza meg pillanatnyilag szabad erőforrások számát?
- Igazolja, magyarázza az egyes processzek végrehajtásának lehetséges 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

	Max.igény				Foglalás			Ideiglenes igények(max.igény-foglalás)		
	R1	R2	R3		R1	R2	R3	R1	R2	R3
p0	7	5	3		0	1	0	7	4	3
p1	3	2	2	Runnable	2	0	0	1	2	2
p2	9	0	2		3	0	2	6	0	0
p3	2	2	2	Runnable	2	1	1	0	1	1
p4	4	3	3		0	0	2	4	3	1

eroforras készlet	foglalatok összesen szabad ef.sza	szabad ef.szam-kiel.igény			készlet igény				
		R1	R2	R3	R1	R2	R3		
		7	2	5					
		10	5	7					
		3	3	2	-4	-1	-1	-1	
					2	1	0	0	Runnable
					-3	3	2	2	
					3	2	1	1	Runnable
					-1	0	1	1	

2.feladat: Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy csővezetékét, a gyerek processz beleír egy szöveget a csővezetékbe (A kiírt szöveg: XY neptunkod), a szülő processz ezt kiolvassa, és kiírja a standard kimenetre.

Mentés: neptunkod_unnamed.c

```
tihor@jerry:~/Documents/Gyakorlat/10/2$ ./THDWDR_unnamed
Gyermek processz, írja be a szöveget:
THDWDR
Szülő processz, a kapott string: THDWDR
tihor@jerry:~/Documents/Gyakorlat/10/2$
```

3.feladat: Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy nevesített csővezetékét (neve: neptunkod), a gyerek processz beleír egy szöveget a csővezetékbe (A hallgató neve: pl.: Keserű Ottó), a szülő processz ezt kiolvassa, és kiírja a standard kimenetre.

Mentés: neptunkod_named.c

```
tihor@jerry:~/Documents/Gyakorlat/10/3$ ./THDWDR_named
Gyermek processz, írja be a szöveget:
TihorFruzsina
Szülő processz, a kapott string: TihorFruzsina
tihor@jerry:~/Documents/Gyakorlat/10/3$
```

4.Gyakorló feladat: Írjon három C nyelvű programot, ahol készít egy *üzenetsort* és ebbe két *üzenetet* tesz bele – **msgcreate.c**, majd olvassa ki az üzenetet - **msgrcv.c**, majd szüntesse meg az *üzenetsort* (takarít) - **msgctl.c**.

A futtatás eredményét is tartalmazza a jegyzőkönyv. Mentés:

msgcreate.c; msgrcv.c; msgctl.c.

4a.: Írjon egy C nyelvű programot, melyben

- az egyik processz létrehozza az *üzenetsort*, és szövegeket küld bele, **exit** üzenetre kilép,
- másik processzben lehet választani a feladatok közül: üzenetek darabszámának lekérdezése, 1 üzenet kiolvasása, összes üzenet kiolvasása, *üzenetsor* megszüntetése, kilépés.

Mentés: **gyak10_4.c**

```
tihor@jerry:~/Documents/Gyakorlat/10/4$ ./msgcreate
Az msgid létrejött: 196608, 30000
Az 1. msgsnd() visszaadott: 0
A kiküldött üzenet: Egyik üzenet
A 2. msgsnd visszaadott: 0
A kiküldött üzenet: Masik üzenet
```

```
tihor@jerry:~/Documents/Gyakorlat/10/4$ gcc -o msgrcv msgrcv.c
tihor@jerry:~/Documents/Gyakorlat/10/4$ ./msgrcv
Az msgid: 196608
Az üzenetek száma az üzenetsorban: 2
msgrcv() visszaadta: 13, a kapott üzenet: Egyik üzenet
msgrcv() visszaadta: 13, a kapott üzenet: Masik üzenet
```

```

tihor@jerry:~/Documents/Gyakorlat/10/4$ gcc -o msgctl msgctl.c
tihor@jerry:~/Documents/Gyakorlat/10/4$ ./msgctl

Visszatért: 0

```

5. Gyakorló feladat: Először tanulmányozzák Vadász Dénes: Operációs rendszer jegyzet - a témához kapcsolódó fejezetét (5.3.2), azaz

Írjon három C nyelvű programot, ahol

- készít egy osztott memóriát, melyben választott kulccsal kreál/azonosít osztott memória szegmenst - **shmcreate.c**.
- az **shmcreate.c** készített osztott memória szegmens *státusának lekérdezése* – **shmctl.c**
- opcionális: **shmop.c** shmid-del azonosít osztott memória szegmenst. Ezután a segm nevű pointervál-tozót használva a processz virtuális címtartományába kapcsolja (attach) a szegmest (shmat() rendszerhívás). Olvassa, írja ezt a címtartományt, végül lekapcsolja (detach) a shmdt() rendszerhívással).

5a. Írjon egy C nyelvű programot, melyben

- egyik processz létrehozza az *osztott memóriát*,
- másik processz rácsatlakozik az osztott memóriára, ha van benne valamilyen szöveg, akkor kiolvassa, majd beleír új üzenetet,
- harmadik processznél lehet választani a feladatok közül: státus lekérése (szegmens mérete, utolsó shmop-os proc. pid-je), osztott memória megszüntetése, kilépés (2. és 3. proc. lehet egyben is)”

A futtatás eredményét is tartalmazza a jegyzőkönyv.

Mentés: gyak10_5.c

```

tihor@jerry:~/Documents/Gyakorlat/10/5$ ./shmcreate
Nincs meg szegmens! Készítsuk el!

Az shm szegmens azonosítója 360449:

```

```

tihor@jerry:~/Documents/Gyakorlat/10/5$ ./shmctl

Add meg a parancs számát:
0. IPC_STAT (status)
1. IPC_RMID (torles)
> 0
Szegmens merete: 512
Utolso shmop()-os processz pid-je: 0

```

```
tihor@jerry:~/Documents/Gyakorlat/10/5$ ./shmop
```

```
Uj szoveget kerek!
```

```
valami
```

```
Az uj szoveg: valami
```