**Operációs rendszerek BSc**

10. Gyak.

2022. 04. 20.

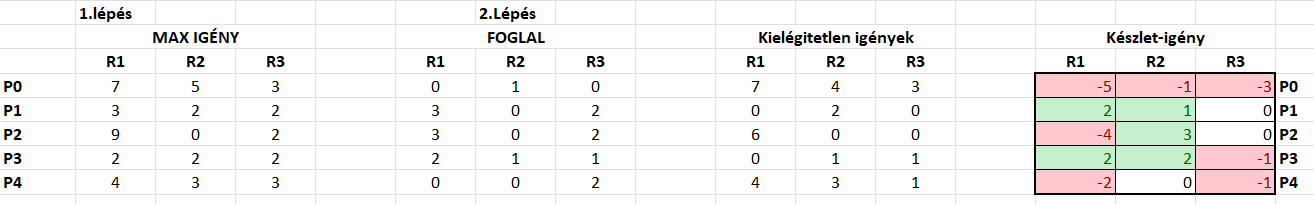
**Készítette:**Sikora Dávid Ádám Bsc  
Mérnökinformatika  
IRE699

Miskolc, 2022

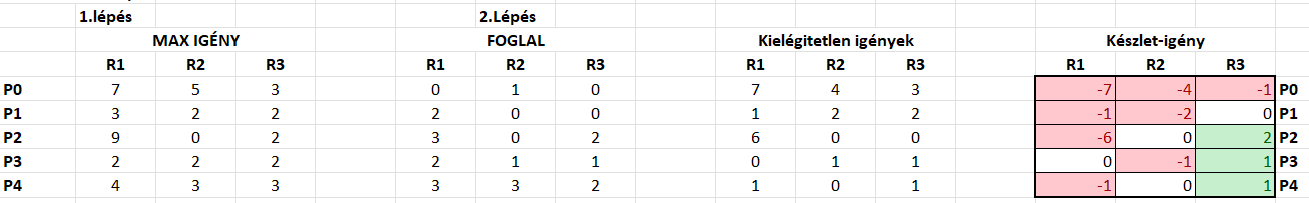
**1.Feladat** Az előadáson bemutatott mintaprogram alapján készítse el a következő feladatot. Adott egy rendszerbe az alábbi erőforrások: R (R1: 10; R2: 5; R3: 7)

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!   
a) Határozza meg a processzek által igényelt erőforrások mátrixát?   
b) Határozza meg pillanatnyilag szabad erőforrások számát?   
c) Igazolja, magyarázza az egyes processzek végrehajtásának lehetséges sorrendjét - számolással?

P1:  


P4:



P0:

A képen asztal látható

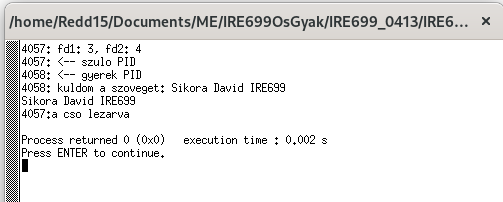
Automatikusan generált leírás

A rendszer a processzek igényeit ki tudja elégíteni, kivéve a P4 igényét. Az alábbi táblázatban látható egy-egy lehetséges végrehajtási sorrend melyben biztonságos marad a rendszer.

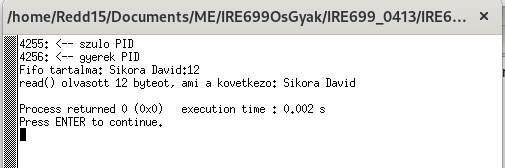
A képen asztal látható

Automatikusan generált leírás

**2.Feladat** Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy csővezetéket, 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

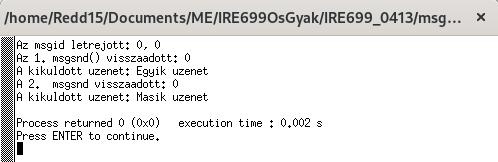
****

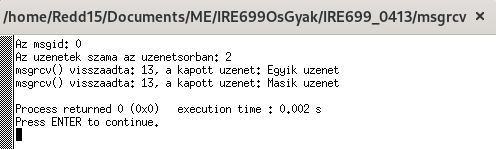
**3.Feladat** Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy nevesített csővezetéket (neve: neptunkod), a gyerek processz beleír egy szöveget a csővezetékbe (A hallgató neve: pl.: Keserű Ottó), a szülő processz ezt kiolvassa, és kiírja a standard kimenetre. Mentés: neptunkod\_named.c

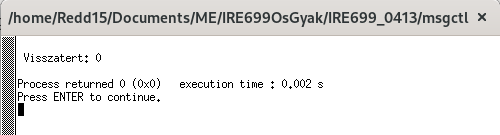
****

**4.Feladat** Először tanulmányozzák Vadász Dénes: Operációs rendszer jegyzet, a témához kapcsolódó fejezetét (5.3)., azaz

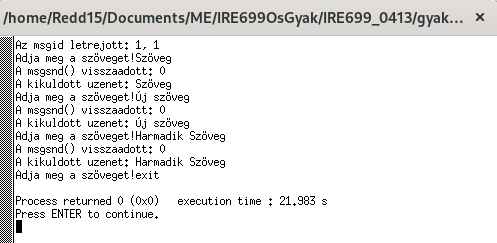
Írjon három C nyelvű programot, ahol készít egy üzenetsort és ebbe két üzenetet tesz bele – msgcreate.c, majd olvassa ki az üzenetet - msgrcv.c, majd szüntesse meg az üzenetsort (takarít) - msgctl.c.   
A futtatás eredményét is tartalmazza a jegyzőkönyv.   
Mentés: msgcreate.c; msgrcv.c; msgctl.c.





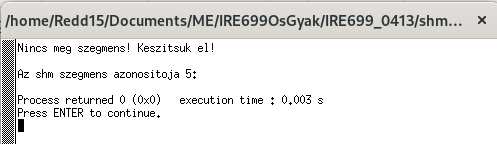


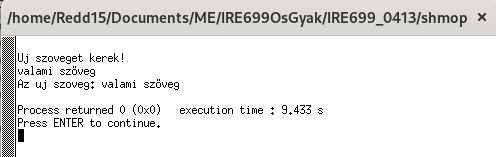
**4a**. Írjon egy C nyelvű programot, melyben  az egyik processz létrehozza az üzenetsort, és szövegeket küld bele, exit üzenetre kilép,   
 másik processzben lehet választani a feladatok közül: üzenetek darabszámának lekérdezése, 1 üzenet kiolvasása, összes üzenet kiolvasása, üzenetsor megszüntetése, kilépés.   
Mentés: gyak10\_4.c   
A futtatás eredményét is tartalmazza a jegyzőkönyv.

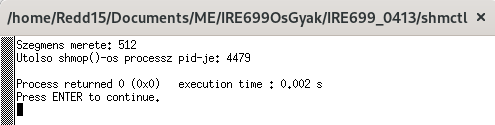




**5.Feladat** Először tanulmányozzák Vadász Dénes: Operációs rendszer jegyzetet - a témához kapcsolódó fejezetét (5.3.2), azaz   
Írjon három C nyelvű programot, ahol   
 készít egy osztott memóriát, melyben választott kulccsal kreál/azonosít osztott memória szegmenst - shmcreate.c.   
 az shmcreate.c 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 címtartomanyába kapcsolja (attach) a szegmest (shmat() rendszerhívás). Olvassa, irja 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

