

Problémák csövek használatakor

A cső állapotának lekérdezése nem garantálja az adatérvényességet.

Például, ha két folyamat közel egyidőben lekérdezi, hogy van-e adat a csőben, amiben egyetlen adat van, akkor mindkettő azt az információt kapja, hogy van kiolvasható adat. Ha ezután mindkettő megpróbál olvasni a csőből, az egyik kiolvassa az adatot, a másik pedig nem talál adatot benne.



Problémák osztott memória használata során



- **Olvasáskor:** Érvényes-e az adat, amit a folyamat felhasználna az osztott memóriaterületről,
 - vagy még nem került oda érvényes adat,
 - vagy már megkezdte a feldolgozását egy másik folyamat?
- **Íráskor:** Szabad a memóriaterület, ahová írna a folyamat,
 - vagy még feldolgozatlan (érvényes) adatot fogunk felülírni,
 - vagy mialatt olvassa ki az adatot egy másik folyamat, módosítjuk azt?



Megoldandó problémák



Kritikus szakasz: a folyamat azon szakasza, amikor egy közösen használt erőforráson manipulál. A folyamatok szinkronizálásához meg kell oldani, hogy a folyamat ne lépjen be a kritikus szakaszba, ha

- nem érvényes az információ, amit a kritikus szakaszban olvasni kíván,
- érvényes még feldolgozásra váró információ van azon a helyen, ahová írni szeretne a kritikus szakaszban,
- másik folyamat manipulál a kritikus szakaszban használandó erőforráson.



IPC szemafor parancssori műveletek:

- **ipcs**: kilistázza az IPC-ket.
- **ipcrm** -s {szemaforkészlet azonosító}: eltávolítja a szemaforkészletet (pl. beragadt).
- **ipcmk** -S {elemszám}: új szemaforkészletet hoz létre.

IPC parancssori műveletek:

- **ipcrm** -a: eltávolítja az összes üzenetsort, osztott memóriát és szemaforkészletet.



SystemV szemaforkészlet létrehozása



<sys/sem.h>

- **semget**({kulcs}, {szemaforok száma}, {flagek})
Létrehoz egy {szemaforok száma} darab szemaforból álló szemaforkészletet.

{kulcs}: key_t típusú (ftok-kal generált vagy IPC_PRIVATE)

{flagek}:

- fájlhozzáférési engedélyek (S_IRUSR, S_IWUSR, ...),
- IPC_CREAT,
- IPC_EXCL.

értéke: nem negatív szemaforkészlet azonosító (semid), vagy -1 ha hiba.



SystemV szemaforkészlet inicializálása és törlése



- **semctl**({semid}, {szemafor}, {parancs}) vagy
- **semctl**({semid}, {szemafor}, {parancs}, {érték})
Elvégzi a {semid} szemaforkészlet {szemafor} azonosítójú szemaforján a {parancs} által meghatározott műveletet.

értéke: {parancs} függő, vagy -1 ha hiba.

{parancs}:

- IPC_RMID: Törli a szemaforcsaládot, {szemafor} értéke lényegtelen (0), {érték} paraméter nem kell.
- SETVAL: Beállítja a szemafor értékét (semval) az {érték} paraméterben megadott értékre
- ...



SystemV szemaforműveletek végrehajtása



- **semop**({semid}, &{műveletek tömbje}, {műveletek száma})

Sorban végrehajtja a {műveletek tömbjé}-ben meghatározott {műveletek száma} darab műveletet a {semid} szemaforkészlet tagjain.

{műveletek tömbje}: egy sembuf típusú struktúrák alkotta tömb

értéke: 0 ha sikeres, vagy -1 ha hiba.



SystemV szemaforműveletek megadása



- **sembuf**: struktúra, adattagjai:
 - sem_num (unsigned short): a szemafor száma, melyen a művelet elvégzendő,
 - sem_flag (short):
 - IPC_NOWAIT: nem vár a művelet sikeres végrehajtására,
 - SEM_UNDO: a folyamat terminálásakor visszavonásra kerül,
 - 0: vár és termináláskor nem kerül visszavonásra.
 - sem_op (short): a végrehajtandó művelet.



SystemV szemaforműveletek megadása



sem_op:

- pozitív: növeli sem_op értékével a szemafor értékét (semval),
- 0: nullára várakozás művelet. Felfüggeszti a folyamat végrehajtását, míg:
 - a szemafor értéke (semval) 0 nem lesz. Ha eleve 0, rögtön folytatja.
 - a szemaforkészletet törlik,
 - szignál érkezik.
- negatív:
 - ha sem_op abszolútértéke kisebb vagy egyenlő mint a szemafor értéke (semval): levonja sem_op értékét a szemafor értékéből és a végrehajtást rögtön folytatja,
 - ha sem_op abszolútértéke nagyobb mint a szemafor értéke (negatív lenne a kivonás eredménye) és nincs IPC_NOWAIT beállítva, akkor vár, míg:
 - a szemafor értéke nagyobb lesz, mint sem_op abszolútértéke, majd levonja azt a szemafor értékéből és folytatja a folyamat végrehajtását,
 - a szemaforkészletet törlik,
 - szignál érkezik,
 - ha sem_op abszolútértéke nagyobb mint a szemafor értéke és be van állítva IPC_NOWAIT flag, akkor hibát generál.



Feladat: osztott memóriaterület kezelésének vezérlése

Az osztott memória egy területén ugyanazon típusú információt adnak át egymásnak a folyamatok. Több olyan folyamat van, melyek létrehozzák az információt, majd beírják a tárterületre (író folyamatok), és több olyan folyamat is van, melyek kiolvassák az információt, majd feldolgozzák azt (olvasó folyamatok).

Az azonos típusú folyamatok egyenrangúak. Tehát, ha egyszerre több író folyamat készül el a beírandó információval, addig kell várniuk, míg felszabadul a közösen használt memóriaterület, majd az egyik beírja az adatot, és ezután nekilát a következő információ létrehozásának, a többi pedig vár a sorára.

Az olvasó folyamatok ugyanígy viselkednek, várnak a sorukra, míg lesz számukra kiolvasható adat. Egy adatot csak egy olvasó folyamat olvashat ki!



Egy szemafor: Kritikus szakasz

Szemafor:

- szemafor[0] – A tárterületet manipuláló kritikus szakaszba beléphet-e a folyamat?
 - 1: beléphet,
 - 0: nem léphet be, mert már egy másik benn van.

Kezdőérték: 1 (mert kezdetben nincs benne folyamat)

Író folyamat:

- semnum: 0; semop: -1; semflag: 0 (beléphet a kritikus szakaszba?)
- írás
- semnum: 0; semop: +1; semflag: 0 (kilépett a kritikus szakaszból)

Olvasó folyamat:

- semnum: 0; semop: -1; semflag: 0 (beléphet a kritikus szakaszba?)
- olvasás
- semnum: 0; semop: +1; semflag: 0 (kilépett a kritikus szakaszból)

Nem jó! Könnyen belátható, hogy pl. egymás után két író folyamat beléphet, felülírva az előzőt anélkül, hogy kiolvasnák!



Egy szemafor - Adatérvényesség



Szemafor:

- szemafor[0] – A tárterület, érvényes adatot tartalmaz?
 - 1: érvényes,
 - 0: érvénytelen.

Kezdőérték: 0 (mert kezdetben nincs benne érvényes adat)

Író folyamat:

- semnum: 0; semop: 0; semflag: 0 (írhat a közös területre?)
- írás
- semnum: 0; semop: +1; semflag: 0 (érvényes olvasható adat van a közös területen)

Olvasó folyamat:

- semnum: 0; semop: -1; semflag: 0 (olvashat a közös területről?)
- (nem tartalmaz a terület olvasható adatot más számára)
- olvasás



Egy szemafor: Adatérvényesség

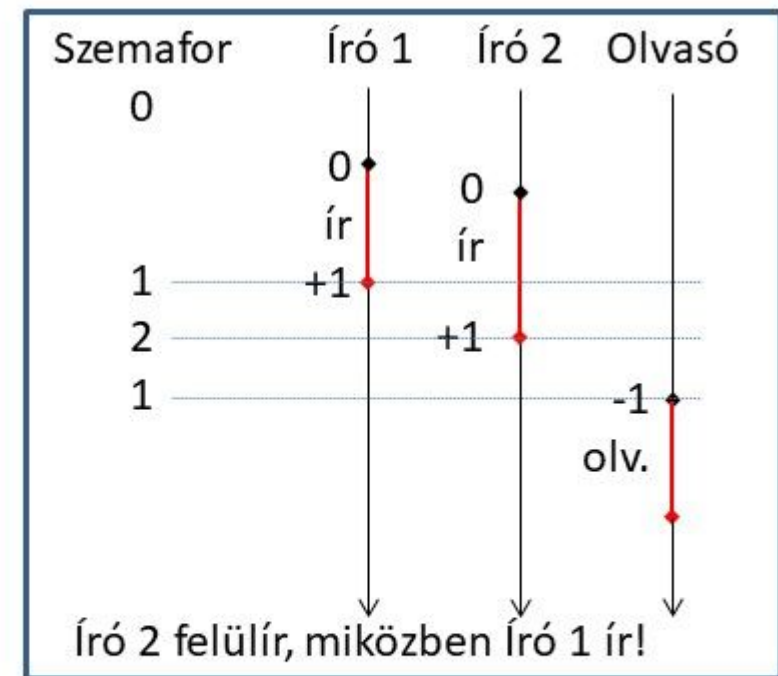
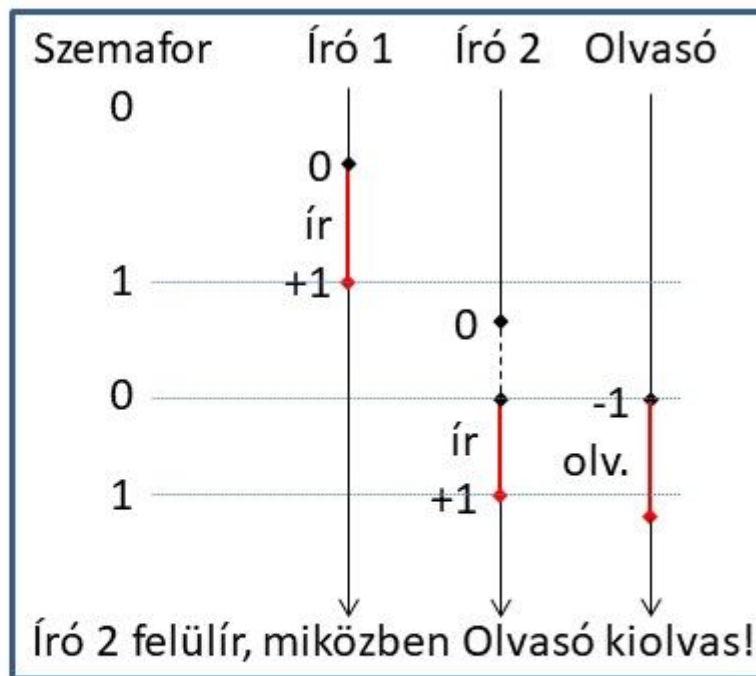
Szemafor: 1 = érvényes, 0 = érvénytelen, kezdőérték: 0

Író folyamat:

- semop: 0
- írás
- semop: +1

Olvasó folyamat:

- semop: -1
- olvasás



Két szemafor: Adatérvényesség

Szemaforok:

- szemafor[0] – Írható a tárterület, nem tartalmaz érvényes adatot!
 - 0: Nem írható. Érvényes adatot tartalmaz, vagy épp írják vagy olvassák.
 - 1: Írható. Nem tartalmaz érvényes adatot, és nem manipulálnak rajta.Kezdőérték: 1 (mert nincs kezdetben adat beírva)
- szemafor[1] – Olvasható a tárterület, érvényes adatot tartalmaz!
 - 0: Nem olvasható. Nincs benne érvényes adat, vagy épp manipulálják.
 - 1: Olvasható. Érvényes adat van benne, és nem manipulálnak rajta.Kezdőérték: 0 (mert kezdetben nincs benne érvényes adat)

Író folyamat:

- semnum: 0; semop: -1; semflag: 0 (írhat a közös területre?)
- (lefoglalta, nem szabad a terület írásra)
- írás
- semnum: 1; semop: +1; semflag: 0 (érvényes olvasható adat van a közös területen)

Olvásó folyamat:

- semnum: 1; semop: -1; semflag: 0 (olvashat a közös területről?)
- (lefoglalta, nem tartalmaz a terület olvasható adatot más számára)
- olvasás
- semnum: 0; semop: +1; semflag: 0 (írható a közös terület)



Két szemafor: Adatérvényesség

Szemafor[0]: 1 = írható, 0 = nem írható, kezdőérték: 1

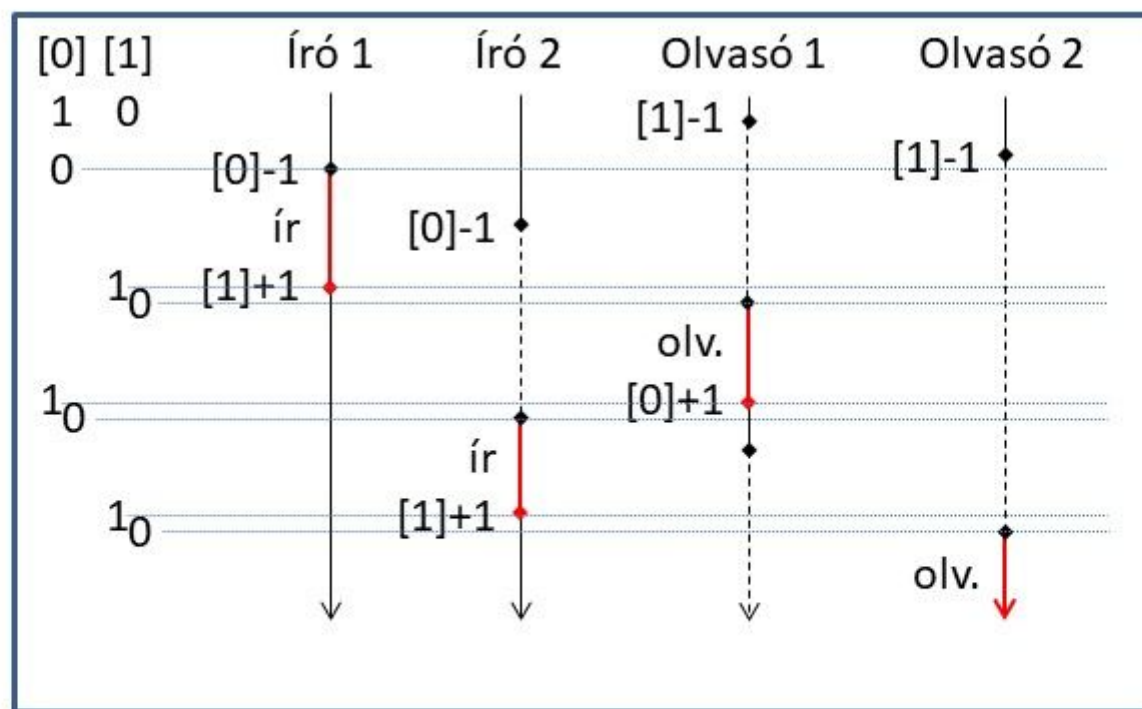
Szemafor[1]: 1 = olvasható, 0 = nem olvasható, kezdőérték: 0

Író folyamat:

- semnum: 0; semop: -1
- írás
- semnum: 1; semop: +1

Olvásó folyamat:

- semnum: 1; semop: -1
- olvasás
- semnum: 0; semop: +1





POSIX szemaforok

<semaphore.h>

- **sem_init**: név nélküli szemafort hoz létre.
- **sem_open**({nev}, {mód}) vagy
- **sem_open**({nev}, {mód}, {hozzáférési engedélyek}, {kezdőérték})

Létrehoz egy nevesített szemaforot {kezdőérték}
indulóértékkel, vagy kapcsolódik egy már létezőhöz.

{nev}: char* típusú (pl. "/temp")

{mód}:

- O_CREAT,
- O_EXCL,
- ...

{hozzáférési engedélyek}: fájlhozzáférési engedélyek
(S_IRUSR, S_IWUSR, ...)

értéke: (sem_t) típusú, a szemafor címe (semaddr), vagy SEM_FAILED ha hiba.

POSIX szemaforok



- **sem_post({semaddr})**
1-gyel növeli a szemafor értékét (felnyitja, unlock).
- **sem_wait({semaddr})**: 1-gyel csökkenti a szemafor értékét (lezárja, lock). Ha a szemafor értéke 0, akkor vár, míg >0 lesz, és csak akkor dekrementálja.
- **sem_trywait({semaddr})**: Működése megegyezik sem_wait működésével, de ha 0 a szemafor értéke, nem várakozik, hanem -1 értékkel tér vissza.



POSIX szemaforok



- **sem_timedwait**({semaddr}, {timespec})
Működése megegyezik sem_wait működésével, de ha 0 a szemafor értéke, akkor legfeljebb {timespec}-ben megadott időpontig vár.
- **sem_unlink**({név})
Törli a {név} azonosítójú szemafort.



Étterem beléptetés probléma^{ik}

- Étterem kapacitása adott.
- Új társaság csak akkor léphet be az étterembe, amikor van megfelelő mennyiségű üres hely.



Termelő-raktár-fogyasztó probléma

- **Raktár** kapacitása adott.
- **Termelő** raktárra termel. Álljon le a termeléssel, amíg a raktár tele van.
- **Fogyasztó** raktárból vételez. Várjon, ha a raktár üres.

