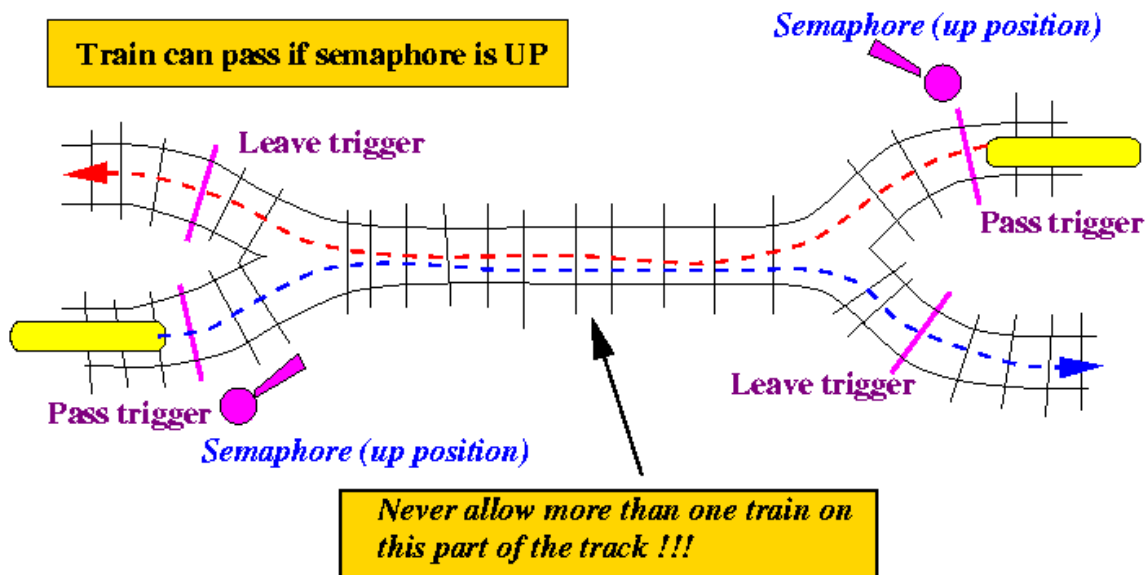


9. Gyakorlat

Processzek közötti kommunikáció: semaforok



Létrehozott semaforok listája: `ipcs -s`

Megadott azonosítójú semafor törlése: `ipcrm -s semid`

Szemafor létrehozása:

Erre a célra a `semget` rendszerhívás használható, ami valójában egy megadott darabszámú semafor készletet hoz létre.

A függvény paraméterei: egy egyedi kulcs, hány elemű semafor készletet kell létrehozni, illetve a hozzáférési jogosultságok.

A visszatérési értéke normál esetben a létrehozott semafor készlet azonosítója, hiba esetén -1.

Pl.: `int semid = semget(19950914, 1, 00600 | IPC_CREAT);`

Részletekért, lásd: **man semget**

Szemafor kezelése:

A semaforok kezelésére a `semctl` rendszerhívás használható, melynek segítségével információkat kérdezhetünk le a semaforokról, megszüntethetjük azokat, stb.

A függvény paraméterei: a semafor készlet azonosítója, a semafor készlet hányadik elemére vonatkozik a művelet (nullás indexelés), a végrehajtandó művelet, és egy további érték (lehet pl.: `struct semid_ds *`, vagy `int` típusú), melynek felhasználása a megadott művelettől függ.

A megadható műveletek a következők:

- `IPC_STAT`: az üzenetsor adatait a megadott struktúrába másolja,
- `IPC_SET`: a megadott struktúrán keresztül módosíthatók az üzenetsor adatai,
- `IPC_RMID`: üzenetsor megszüntetése,
- `SETVAL`: a megadott indexen lévő semafor értékének beállítása (mindegyikre: `SETALL`),

A visszatérési érték hiba esetén -1, egyébként 0.

Pl.: `int ret = semctl(semid, 0, SETVAL, 1);`

Részletekért, lásd: **man semctl**

Szemafor értékének változtatása:

A szemaforok értékének megváltoztatása a **semop** segítségével oldható meg.

A függvény paraméterei: a szemafor készlet azonosítója, az elvégzendő műveleteket reprezentáló *struct sembuf* típusú struktúrákat tartalmazó tömb, illetve a tömb mérete.

A struktúra a következőképpen néz ki:

```
struct sembuf {
    unsigned short sem_num; // hányadik szemaforra vonatkozik a művelet
    short sem_op; // a végrehajtandó művelet: -1 = DOWN, 1 = UP
    short sem_flg; // flag, értéke: 0, IPC_NOWAIT vagy SEM_UNDO (vagy ezek kombinációja)
};
```

A visszatérési értéke hiba esetén -1, egyébként 0.

Pl.: `int ret = semop(semid, up, 1);`

Részletekért, lásd: **man semop**

Holtpont:**Feladat 1:**

Eddigi ismereteinket felhasználva írjunk programot, amelyben szemaforok segítségével holtpont helyzetet alakítunk ki a szülő és a gyerek processz között.

Feladat 2:

Eddigi ismereteinket felhasználva írjunk programot, amelyben 5 gyerek processz párhuzamosan futva megpróbálja tízszer egymás után 1-10 a számokat pipe-on keresztül elküldeni a szülőnek, ami ezeket kiolvassa és kiírja a képernyőre. Szemaforok segítségével oldjuk meg, hogy az eredményben a kiírásakor ne keveredjenek a számok (pl.: 1, 2, 3, 1, 4, 2, 5, 6, 3, 1, ...), hanem folyamatosan 1-10-ig ismétlődjenek, keveredés nélkül.