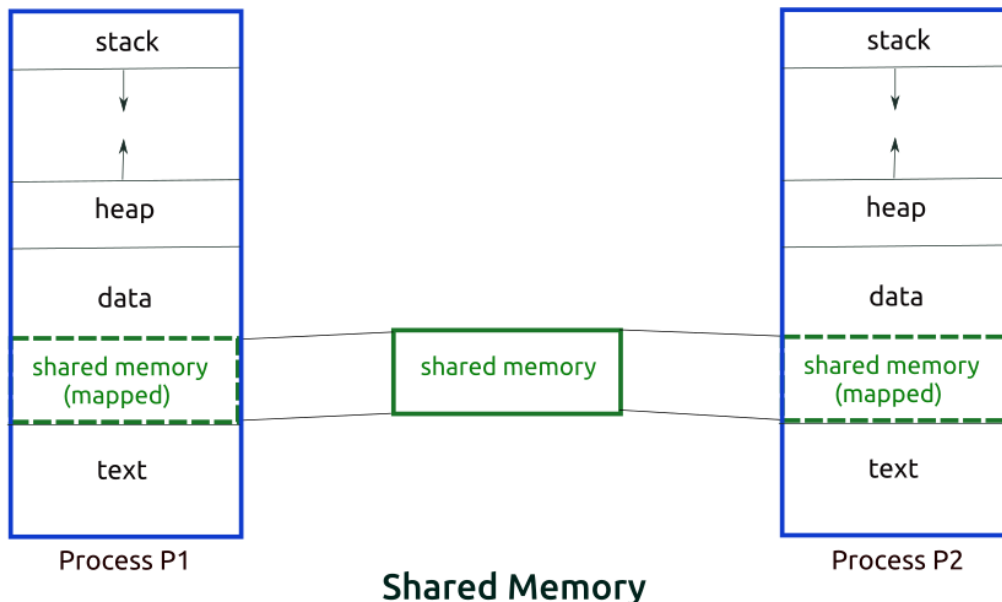


## 8. Gyakorlat

### Processzek közötti kommunikáció: osztott memória szegmensek



Létrehozott osztott memória szegmensek listája: `ipcs -m`

Megadott azonosítójú osztott memória szegmens törlése: `ipcrm -m shmid`

#### Osztott memória szegmens létrehozása:

Osztott memória szegmenseket létrehozásához az **shmget** rendszerhívás áll rendelkezésünkre.

A függvény paraméterei: egy egyedi kulcs, a létrehozandó osztott memória szegmens mérete, a hozzáférési jogosultságok.

A visszatérési értéke hiba esetén -1, egyébként a létrehozott osztott memória szegmens azonosítója, amellyel a létrehozott szegmensre hivatkozhatunk.

Pl.: `int shmid = shmget(19950914, 4096, 00600 | IPC_CREAT);`

Részletekért, lásd: **man shmget**

#### Osztott memória szegmens hozzátcsolása a processz címtartományához:

Ahhoz, hogy egy létrehozott osztott memória szegmessel dolgozni is tudjunk, hozzá kell csatolnunk a processzünk címtartományához. Ezt az **shmat** segítségével tehetjük meg.

A függvény paraméterei: az osztott memória szegmens azonosítója, milyen címre kell csatolni a szegmenst (ha NULL, akkor a kernel dönti el), és egy flag.

A flag a következőket veheti fel értékül:

- `0`: normál működés, a szegmens írható és olvasható is,
- `SHM_RDONLY`: a szegmens csak olvasható (csak írható nincs)
- `SHM_RND`: ha a megadott cím nem NULL, akkor a címet egy SHMLBA (a lapméret többszöröse) érték legközelebbi többszörösére "kerékíti".

A visszatérési értéke hiba esetén (`void *`)-1, egyébként a csatolt osztott memória szegmens címe.

Pl.: `struct MySHM *myShmSegment = (struct MySHM *) shmat(shmid, NULL, 0);`

Részletekért, lásd: **man shmat**

### Osztott memória szegmens leválasztása a processz címtartományáról:

Miután befejeztük a munkát az osztott memória szegmensünkkel, le kell, hogy válasszuk azt a processzünk címtartományáról. Ezt az **shmdt** segítségével tehetjük meg.

A függvény paraméterként a leválasztandó osztott memória szegmens címét várja.

A visszatérési értéke hiba esetén -1, egyébként 0.

Pl.: `int ret = shmdt(myShmSegment);`

Részletekért, lásd: **man shmdt**

### Osztott memória szegmens kezelése:

Az osztott memória szegmensek kezelésére az **shmctl** rendszerhívás használható, melynek segítségével például információkat kérdezhetünk le a szegmensünkről, illetve megszüntethetjük azt. A függvény paraméterei: az osztott memória szegmens azonosítója, a végrehajtandó művelet, és egy `struct shmids` típusú struktúrára mutató pointer, melynek felhasználása a megadott művelettől függ.

A megadható műveletek a következők:

- **IPC\_STAT**: az osztott memória szegmens adatait a megadott struktúrába másolja,
- **IPC\_SET**: a megadott struktúrán keresztül módosíthatók az osztott memória szegmens adatai,
- **IPC\_RMID**: osztott memória szegmens megszüntetése.

A visszatérési értéke hiba esetén -1, egyébként 0.

Pl.: `int ret = shmctl(shmid, IPC_STAT, &buf);`

Részletekért, lásd: **man shmctl**

### Feladat 1:

A már jól ismert tippelgetős programunkat írjuk át olyan módon, hogy a gyerek a tippjeit osztott memórián keresztül juttassa el a szülőhöz.

### Feladat 2:

Eddigi ismereteinket felhasználva írjunk chat programot, amelyben a szülő processzt üzenetküldésre, a gyereket pedig üzenet fogadásra használjuk. A programmal csoportos beszélgetést valósítunk meg, amikor elindul a program, az osztott memória szegmensen keresztül jelzi a többinek, hogy becsatlakozott a beszélgetésbe. Új üzenetet csak akkor küldhetünk, ha az előzőt már mindenki látta, ha hamarabb próbáljuk küldeni, várakoztatni kell, addig még végre nem tudja hajtani. A program leállásakor jelezni kell a többi felé, hogy kilépett a beszélgetésből.

### Feladat 3:

Eddigi ismereteinket felhasználva írjunk programot, amelyben a szülő processz beolvassa egy fájl tartalmát, majd osztott memórián keresztül elküldi azt a gyerekeknek. A gyerek processz kiolvassa a szöveget az osztott memória szegmensből, megszámolja hány darab normál karaktert tartalmaz (nem speciális karakter, nem írásjel, nem whitespace, stb.), majd az eredményt elküldi a szülőnek, ami pedig kiírja azt a képernyőre.