

# Deljena memorija

- Segment memorije koji se mapira i deli između više procesa
- Najbrži oblik IPC, nema posrednika
- Memorijski segment se mapira direktno u adresni prostor procesa
- Kada jedan proces kreira segment, više procesa može da čita iz njega (za razliku od redova za poruke) i piše u njega
- `#include <sys/shm.h>`

# Kreiranje segmenta deljene memorije

- `int shmget(key_t key, size_t size, int shmflg);`
- `key` – identifikator segmenta deljene memorije
- `size` - veličina segmenta u bajtovima
- `shmflg`– šta raditi sa segmentom (kreiranje, permisije)
- Primer:
  - `shmid = shmget(kljuc, 1024, 0644 | IPC_CREAT);`
- Povratna vrednost – id segmenta ili `-1` ako segment nije uspešno alociran

# shmat()

- `void *shmat(int shmid, const void *shmaddr, int shmflg);`
- Kači memorijski segment u adresni prostor procesa
- `shmid` – id segmenta
- `shmaddr` – stavljamo `NULL`, koristi se veoma retko
- `shmflg` – fleg, uglavnom ćemo koristiti `0`
- Primer:
  - `pointer = shmat(id_segmenta, NULL, 0);`
- Povratna vrednost – adresa početka segmenta ili `-1` ako je došlo do greške

# shmdt()

- `int shmdt(const void *shmaddr);`
- Odvaja segment deljenje memorije lociran na adresi definisanoj sa `shmaddr` od adresnog prostora pozivajućeg procesa.
- Parametar `shmaddr` je povratna vrednost sistemskog poziva `shmat()`.
- Primer:
  - `shmdt(pointer);`
- Povratna vrednost – 0 ako je uspešno, -1 ako je došlo do greške

# Uklanjanje segmenta deljene memorije

- `int shmctl(int shmid, int cmd, struct shmid_ds *buf);`
- `shmid` – id segmenta
- `cmd` - `IPC_RMID` - Obeležava segment za brisanje. Stvarno brisanje se vrši kada se poslednji proces koji je prikačen na segment otkači.
- `buf` - Ne treba nam za brisanje, `NULL`
- Primer
  - `shmctl(id_segmenta, IPC_RMID, NULL);`
- Ako je brisanje uspešno, vraća 0, a ako je došlo do greške pri uklanjanju, vraća -1.