

Semafori - implementacija

Kreiranje semafora

- `#include <sys/sem.h>`
- `int semget(key_t key, int nsems, int semflg);`
- Ova funkcija kreira **skup** semafora
- `key` – identifikator skupa semafora
- `nsems` – broj semafora u skupu
- `semflg` – fleg (kreiranje, permisije)
- Povratna vrednost – id skupa semafora ili `-1` ako je došlo do greške
- Primer:
 - `semid = semget(key, 1, 0644 | IPC_CREAT);`

Inicijalizacija i brisanje semafora

- `int semctl(int semid, int semnum, int cmd, ... /* arg */);`
- `semid` – id skupa semafora
- `semnum` – redni broj semafora u skupu (broji se od 0)
- `cmd` – naredba koju je potrebno izvršiti nad semaforom
- `arg` – ovaj parametar se nekad koristi, a nekad ne, zavisno od toga koja se naredba zadaje sa `cmd`
 - Arg mora biti tipa unije `semun`. Ova unija se mora definisati u kodu na sledeći način:
 - `union semun {`
 - `int val; /* used for SETVAL only */`
 - `struct semid_ds *buf; /* used for IPC_STAT and IPC_SET */`
 - `ushort *array; /* used for GETALL and SETALL */`
 - `};`
- Povratna vrednost: `-1` ako je došlo do greške

Inicijalizacija i brisanje semafora

- cmd naredbe:
 - **SETVAL** - Postavlja vrednost promenljive zadatog semafora na vrednost val člana unije semun.
 - **GETVAL** - Vraća vrednost promenljive zadatog semafora kao povratnu vrednost funkcije semctl().
 - **SETALL** - Postavlja vrednosti svih semafora u skupu na vrednosti iz niza n array unije semun. Parametar semnum funkcije semctl() se u ovom slučaju ignoriše.
 - **GETALL** - Vraća vrednost svih semafora u skupu i skladišti ih u niz array unije semun. Parametar semnum se ignoriše.
 - **IPC_RMID** - Uklanja skup semafora.

Primeri

- Definisanje unije i deklarisanje promenljive arg:
 - ```
union semun {
 int val; /* used for SETVAL only */
 struct semid_ds *buf; /* used for IPC_STAT and IPC_SET */
 ushort *array; /* used for GETALL and SETALL */
};
```
  - `union semun arg;`
- Inicijalizacija prvog semafora u skupu:
  - `arg.val = 1;`
  - `semctl(semid, 0, SETVAL, arg);`
- Brisanje prvog semafora u skupu:
  - `semctl(semid, 0, IPC_RMID);`

# Operacije nad semaforima

- `int semop(int semid, struct sembuf *sops, size_t nsops);`
- `semid` – id skupa semafora
- `sops` - pokazivač na niz operacija. Svaka operacija je definisana jednom strukturom `sembuf`
- `nsops` – broj operacija, broj elemenata niza `sops`
  - Najčešće ćemo definisati samo jednu operaciju u jednom pozivu funkcije `semop`

# Operacije nad semaforima

- Definicija strukture sembuf:

- ```
struct sembuf
{
    ushort sem_num; /* semaphore index in array */
    short sem_op; /* semaphore operation */
    short sem_flg; /* operation flags */
};
```

- sem_num – indeks semafora u skupu

- sem_op – Od vrednosti ove promenljive zavisi koja operacija će biti izvršena nad semaforom:

- Negativna vrednost – Zauzima se $\text{abs}(\text{sem_op})$ resursa semafora, vrednost semafora se smanjuje za $\text{abs}(\text{sem_op})$, ili se proces blokira ako nema dovoljno resursa na raspolaganju
- Pozitivna vrednost - Oslobađa se sem_op resursa semafora, vrednost semafora se povećava za sem_op
- 0 – proces se blokira dok vrednost semafora ne dostigne 0.

- sem_flg - Služi za flegove operacija:

- IPC_NOWAIT – Kada dođe do situacije da proces treba da se blokira na semaforu, on se ne blokira, već semop vraća -1.
- SEM_UNDO - Označava da nakon kraja programa treba poništiti efekat operacije. Korisno kada postoji mogućnost da se program neočekivano završi (npr. SIGKILL signalom), a potrebno je osloboditi resurse

Primeri

- Struktura sembuf ne mora da se definiše, samo se deklariše promenljiva
- Izvođenje P operacije nad prvim semaforom u skupu:
 - struct sembuf P;
 - P.sem_num = 0;
P.sem_op = -1;
P.sem_flg = 0;
 - semop(semid, &P, 1);
- Izvođenje V operacije nad prvim semaforom u skupu:
 - struct sembuf V;
 - V.sem_num = 0;
V.sem_op = 1;
V.sem_flg = 0;
 - semop(semid, &V, 1);