

Međuprocesna komunikacija

- IPC - Interprocess communication
- Mehanizmi operativnog sistema koji omogućavaju procesima da komuniciraju i dele podatke
 - Redovi za poruke (message queues)
 - Deljena memorija
 - Semafori
 - Signali
 - ...

Redovi za poruke

- Povezana lista u adresnom prostoru operativnog sistema
- Procesi mogu da razmenjuju informacije preko zajedničkog reda za poruke
- Naredbe u terminalu:
 - `ipcs -q` : prikaz svih redova za poruke
 - `ipcrm <id_reda>`: uklanjanje reda za poruku

Kreiranje reda

- `int msgget(key_t key, int msgflg);`
- `key` – identifikator reda za poruke, celi broj (long)
- `msgflg` – šta raditi sa redom (kreiranje, permisije)
- Primer:
 - `id_reda = msgget(kljuc, IPC_CREAT|0666);`
- `IPC_CREAT` – proces kreira red za poruke
- Bez `IPC_CREAT` (`msgflg=0`) - proces se povezuje na postojeći red za poruke
- `0666` - permisije
- Povratna vrednost – id reda ili `-1` ako red nije uspešno kreiran

Permisije

- Read (r) – pravo čitanja
- Write (w) – pravo upisa
- Execute (x) – pravo izvršavanja
- Oktalni režim

Pravo :	r	w	x
EkspONENT	2^2	2^1	2^0
Vrednost pozicije :	4	2	1

Oktalna vrednost	Suma prava po binarnoj vrednosti	Odgovarajuća prava	Definicaj
7	$4 + 2 + 1$	r w x	čitanje, izmena i izvršavanje
6	$4 + 2 + 0$	r w -	čitanje i izmena
5	$4 + 0 + 1$	r - x	čitanje i izvršavanje
4	$4 + 0 + 0$	r - -	čitanje
3	$0 + 2 + 1$	- w x	izmena i izvršavanje
2	$0 + 2 + 0$	- w -	izmena
1	$0 + 0 + 1$	- - x	izvršavanje
0	$0 + 0 + 0$	- - -	bez prava pristupa

- 660 – korisnik i grupa imaju pravo čitanja i pisanja, ostali nemaju nikakvo pravo

Ključ

- Da se ne bi slučajno neka druga grupa procesa povezala na isti red za poruke, ključ mora da bude jedinstveni identifikator
- `key_t ftok(const char *path, int id);`
- Ključ se pravi kombinacijom putanje do fajla i celog broja
- Primer:
 - `kljuc = ftok("./pr1-writer.c", 1);`
- Povratna vrednost – generisan ključ ili `-1` ako red nije uspešno kreiran

Uklanjanje reda

- Red za poruke ostaje u sistemu i nakon kraja procesa ako nije eksplicitno uklonjen
- `int msgctl(int msqid, int cmd, struct msqid_ds *buf);`
- `cmd` - `IPC_RMID` - Briše odgovarajući red za poruke
- `buf` - Ne treba nam za brisanje, `NULL`
- Primer
 - `msgctl(id_reda, IPC_RMID, NULL);`

Struktura poruke

- Šablon

```
struct msgbuf
{
    long mtype; // tip poruke
    char mtext[1]; // tekst poruke
};
```

- Mora imati celobrojno polje za tip poruke. Ostala polja predstavljaju samu poruku.
- Tip - Mogu se slati poruke različitih tipova, primalac naznačava koji tip poruke želi da primi
- Primer:

```
struct imi_poruka
{
    long tip_poruke;
    char tekst_poruke[100];
};
```

```
struct imi_poruka moja_poruka;
moja_poruka.tip_poruke = 1;
fgets(moja_poruka.tekst_poruke, sizeof(moja_poruka.tekst_poruke), stdin);
```

Slanje poruke

- `int msgsnd(int msqid, struct msgbuf *msgp, int msgsz, int msgflg);`
- Primer:
 - `msgsnd(id_reda, &moja_poruka, sizeof(moja_poruka), 0);`
- Povratna vrednost `-1`: poruka nije poslata

Primanje poruke

- `int msgrcv(int msqid, struct msgbuf *msgp, int msgsz, long mtype, int msgflg);`
- Primer
 - `msgrcv(id_reda, &moja_poruka, sizeof(moja_poruka), 1, 0);`
- Ovakav poziv funkcije blokira proces dok ne stigne poruka očekivanog tipa
- `mtype=0` - prima se poruka bilo kog tipa
- `msgflg=IPC_NOWAIT` - poziv `msgrcv` funkcije ne blokira program. Ako nema poruke u redu, onda `msgrcv` odmah završava i vraća -1