

HEADER

```
unistd.h (za fork), stdio.h, stdlib.h, sys/types.h (ako vam treba time_t ili sl),  
sys/ipc.h, sys/sem.h ..., signal.h
```

SIGNAL

```
signal(SIGINT, sighandler);  
void sighandler(int sig) { printf("Not this time!\n");  
/* eventualno bez: exit(0); */
```

SEMAFORI

Zahtev za objektom tipa semafor

```
semget(ftok("sem.c", 'S'), 1, 0666);
```

Inicijalizacija

```
ushort initniz[1];  
initniz[0]=1;  
if ((semctl(sid, 0, SETALL, initniz)) == -1) {  
    perror("Greska u semctl");  
    exit(1);  
};
```

Operacije nad semaforima

Definisanje operacije

```
struct sembuf P_op[1];  
P_op[0].sem_num=0;  
P_op[0].sem_op=-1;  
P_op[0].sem_flg=SEM_UNDO;
```

Izvršavanje operacije

```
#define P semop(sid, P_op, 1)
```

Uklanjanje objekta

```
semctl(sid, 0, IPC_RMID, 0);
```

REDOVI ZA PORUKE

```
struct zahtev_t {  
    long mtype;  
    ...  
} zahtev;
```

Zahtev za objektom

```
servermsqid=msgget(ftok("el_del_s.c", 's'), 0660)
```

Slanje poruke

```
vel_zahteva=sizeof(zahtev)-sizeof(long);  
if(msgsnd(servermsqid, &zahtev, vel_zahteva, 0) == -1) {  
    perror("Klijent: greska u slanju poruke(msgsnd)");  
    exit(MAINERR);  
}
```

Preuzimanje najstarije poruke tipa 1

```
msgrcv(mojmsqid, &odgovor, vel_odgovora, 1, 0)
```

Oslobadjanje objekta

```
msgctl(mojmsqid, IPC_RMID, 0);
```

DELJENA MEMORIJA

```
char *shmptr;  
struct delmem_t {  
    int rbroj;  
    char tekst[30];  
};
```

Zahtev za objektom

```
mojid=shmget(IPC_PRIVATE, sizeof(struct delmem_t), 0666);
```

Attach

```
if ((int)(shmptr=(char *)shmat(mojid, 0, 0)) == -1) {  
    perror("Greska u shmget");  
    exit(MAINERR);  
};
```

Detach

```
(shmdt(shmptr) == -1)
```

Oslobadjanje objekta

```
shmctl(mojid, IPC_RMID, 0)
```