

# Упражнение 8

## IPC обект

*ipc* := { msg | shm | sem } - опашка | памет | семафор

## Създаване (отваряне) на IPC обект

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/ipc.h>
int ipcget(key_t key, int ipcflg) - създава (отваря) IPC обект
```

IPC\_PRIVATE - ключ за опашка между родствени процеси  
*ipcflg*:  
 IPC\_CREAT - създава IPC обект  
 IPC\_EXCL - при създаване, грешка EEXIST, ако съществува

```
int msgget(key_t key, int msgflg)
int shmget(key_t key, size_t size, int shmflg)
int semget(key_t key, int nsems, int semflg)
```

## Структура за права и собственост

```
#include <sys/ipc.h>
struct ipc_perm {
    uid_t uid; - собственик
    gid_t gid; - група на собственика
    uid_t cuid; - създател
    gid_t cgid; - група на създателя
    mode_t mode; - права на достъп
    ...
};
```

## Структура за IPC обект

```
#include <sys/ipc.h>
struct ipcid_ds {
    struct ipc_perm ipc_perm;
    ...
};
```

## Управление на IPC обект

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/ipc.h>
int ipcctl(int ipcid, int cmd, struct ipcid_ds *buf) - управлява IPC обект

cmd:
IPC_STAT - получава информация за IPC обект
```

IPC\_SET - променя собственика и правата на IPC обект

IPC\_RMID - унищожаване на IPC обект

```
int msgctl(int msqid, int cmd, struct msqid_ds *buf)
int shmctl(int shmid, int cmd, struct shmid_ds *buf)
int semctl(int semid, int semnum, int cmd, union semun arg)
```