

# Упражнение 10

## Алокиране (създаване) на сегмент обща памет

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
int shmget(key_t key, size_t size, int shmflg) - алокира сегмент обща памет
```

IPC\_PRIVATE - ключ за обща памет между родствени процеси  
shmflg:  
IPC\_CREAT - създава сегмент обща памет  
IPC\_EXCL - при създаване, грешка EEXIST, ако съществува

## 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 обща памет

```
#include <sys/shm.h>
struct shmid_ds {
    struct ipc_perm shm_perm;
    size_t shm_segsz; - размер на сегмента (байта)
    time_t shm_atime; - време на последно присъединяване
    time_t shm_dtime; - време на последно отделяне
    time_t shm_ctime; - време на последна промяна
    pid_t shm_cpid; - процеса създал общата памет
    pid_t shm_lpid; - последния процес, изпълнил shmat()/shmdt()
    msgqnum_t shm_nattch; - брой на присъединяванията
};
```

## Присъединяване на сегмент обща памет

```
#include <sys/types.h>
#include <sys/shm.h>
void *shmat(int shmid, const void *shmaddr, int shmflg) - присъединява сегмент към адресното пространство

shmflg:
SHM_RND - подравнява адреса надолу
SHM_RDONLY - присъединява сегмента само за четене
```

## Отделяне на сегмент обща памет

```
#include <sys/types.h>
#include <sys/shm.h>
int shmctl(const void *shmaddr) - отделя сегмент обща памет
```

## Управление на общата памет

```
#include <sys/ipc.h>
#include <sys/shm.h>
int shmctl(int shmid, int cmd, struct shmid_ds *buf) - управлява общата памет
```

cmd:

IPC\_STAT - получава информация за общата памет

IPC\_SET - променя собственика и правата на общата памет

IPC\_RMID - маркира общата памет за унищожаване