

Упражнение 9

Създаване (отваряне) на опашка за съобщения

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
int msgget(key_t key, int msgflg) - създава (отваря) опашка за съобщения
```

IPC_PRIVATE - ключ за опашка между родствени процеси
msgflg:
IPC_CREAT - създава опашка
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 опашка

```
#include <sys/msg.h>
struct msqid_ds {
    struct ipc_perm msg_perm;
    time_t msg_stime; - време на последно изпращане
    time_t msg_rtime; - време на последно получаване
    time_t msg_ctime; - време на последна промяна
    msgqnum_t msg_qnum; - брой на съобщенията в опашката
    msglen_t msg_qbytes; - максимален брой байтове в опашката
    pid_t msg_lspid; - последния процес, изпратил съобщение
    pid_t msg_lrpid; - последния процес, получил съобщение
};
```

Изпращане на съобщение

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
int msgsnd(int msqid, const void *msgp, size_t msgsz, int msgflg) -
изпраща съобщение

msgflg:
IPC_NOWAIT - ако няма място, завършва с грешка EAGAIN
```

Получаване на съобщение

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
ssize_t msgrcv(int msqid, void *msgp, size_t msgsz, long msgtyp, int
msgflg) - получава съобщение
```

msgflg:

IPC_NOWAIT - ако няма съобщение, завършва с грешка ENOMSG

MSG_NOERROR - отрязва дългите съобщения, ако не - завършва с грешка E2BIG

Структура на съобщение

```
struct msgbuf {
    long mtype; - тип (>0)
    char mtext[]; - съобщение
};
```

Управление на опашка

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
int msgctl(int msqid, int cmd, struct msqid_ds *buf) - управлява опашката
```

cmd:

IPC_STAT - получава информация за опашката

IPC_SET - променя собственика и правата на опашката

IPC_RMID - унищожава опашката и събужда блокираните процеси с грешка EIDRM