

msgcreate.c (~/Asztal)

Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó

msgcreate.c msgrcv.c msgctl.c

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <string.h>
#include <stdlib.h>
#define MSGKEY 654321L

struct msgbuf1 {
    long mtype;
    char mtext[512];
} sndbuf, *msgp; // Üzenet buffer és pointere

int main()
{
    int msgid; // Üzenetsor azonosító
    key_t key; // Kulcs az üzenetsorhoz
    int msgflg; // flag a kreációhoz, üzenetküldéshez
    int rtn, msgs; // return es méret

    key = MSGKEY;
    msgflg = 00666 | IPC_CREAT;
    msgid = msgget( key, msgflg);
    if ( msgid == -1) {
        perror("\n Az msgget rendszerhívás sikertelen!");
        exit(-1);
    }
    printf("\n Az msgid %d, %x : ", msgid, msgid);

    msgp = &sndbuf;
    msgp->mtype = 1; // Text típus
    strcpy(msgp->mtext,"Az egyik üzenet");
    msgs = strlen(msgp->mtext) + 1; // Az üzenet hossza

    // Majd elküldöm,
    rtn = msgsnd(msgid,(struct msgbuf *) msgp, msgs, msgflg);
    printf("\n Az 1. msgsnd visszaadott %d-t", rtn);
    printf("\n A kiküldött üzenet: %s", msgp->mtext);

    strcpy(msgp->mtext,"Másik üzenet");
    msgs = strlen(msgp->mtext) + 1;
    rtn = msgsnd(msgid,(struct msgbuf *) msgp, msgs, msgflg);
    printf("\n A 2. msgsnd visszaadott %d-t", rtn);
    printf("\n A kiküldött üzenet: %s \n", msgp->mtext);

    exit (0);
}
```

C ▾ Szóközök: 4 ▾ 1. sor, 1. oszlop BESZ

marty@MartyPC: ~/Asztal

Fájl Szerkesztés Nézet Keresés Terminál Súgó

marty@MartyPC:~/Asztal\$ cc msgcreate.c -o main.out

marty@MartyPC:~/Asztal\$./main.out

Az msgid 0, 0 :
Az 1. msgsnd visszaadott 0-t
A kiküldött üzenet: Az egyik üzenet
A 2. msgsnd visszaadott 0-t
A kiküldött üzenet: Másik üzenet

Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó

msgcreate.c x msgrcv.c x msgctl.c x

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <stdlib.h>
#define MSGKEY 654321L

struct msgbuf1 {
    long tipus;
    char mtext[512];
} rdbuf, *msgp; // Message buffer és pointere

struct msqid_ds ds, *buf; // Üzenetsorhoz asszociált struktura és pointere

int main()
{
    int msgid; // Üzenetsor azonosító ID
    key_t kulcs; // Kulcs az üzenetsorhoz
    int tipus, msgflg; // Típus, flag
    int rtn, meret; // return es méret

    kulcs = MSGKEY; // Beállítom a kulcsot
    msgflg = 00666 | IPC_CREAT | MSG_NOERROR;

    msgid = msgget(kulcs, msgflg);
    if (msgid == -1) {
        perror("\n Az msgget rendszerhívás sikertelen!");
        exit(-1);
    }
    printf("\n Az üzenet ID: %d",msgid);

    msgp = &rdbuf; // Üzenetfogadó buffer
    buf = &ds; // Üzenetsor jellemző adataihoz
    meret = 20; // Max hossz
    tipus = 0; // minden típust várok
    rtn = msgctl(msgid,IPC_STAT,buf); // Üzenetsor adatokat lekerdezem
    printf("\n Az üzenetek szama: %ld",buf->msg_qnum);

    while (buf->msg_qnum) { // Van-e üzenet?

        // Veszem a következő üzenetet:
        rtn = msgrcv(msgid,(struct msgbuf *)msgp, meret, tipus, msgflg);
        printf("\n Visszatérés: %d, A fogadott üzenet:%s\n",rtn, msgp->mtext);
        rtn = msgctl(msgid,IPC_STAT,buf);

        // Üzenetsor adatokat lekerdezem, benne azt is, hány üzenet van még
    }
    exit (0);
}
```

marty@MartyPC: ~/Asztal

Fájl Szerkesztés Nézet Keresés Terminál Súgó

```
marty@MartyPC:~/Asztal$ cc msgcreate.c -o main.out
marty@MartyPC:~/Asztal$ ./main.out

Az msgid 2, 2 :
Az 1. msgsnd visszaadott 0-t
A kiküldött üzenet: Az egyik üzenet
A 2. msgsnd visszaadott 0-t
A kiküldött üzenet: Másik üzenet
marty@MartyPC:~/Asztal$ cc msgrcv.c -o main.out
marty@MartyPC:~/Asztal$ ./main.out

Az üzenet ID: 2
Az üzenetek szama: 2
Visszatérés: 17, A fogadott üzenet:Az egyik üzenet

Visszatérés: 15, A fogadott üzenet:Másik üzenet
marty@MartyPC:~/Asztal$
```

msgctl.c (~/Asztal)

Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó

msgcreate.c msgrcv.c msgctl.c

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <stdlib.h>
#define MSGKEY 654321L

int main()
{
    int msgid, msgflg, rtn;
    key_t kulcs;
    kulcs = MSGKEY;
    msgflg = 00666 | IPC_CREAT;
    msgid = msgget(kulcs, msgflg);

    rtn = msgctl(msgid, IPC_RMID, NULL); // Törlöm az üzenetsort
    printf ("\n Visszért: %d \n", rtn);

    exit (0);
}
```

marty@MartyPC: ~/Asztal

Fájl Szerkesztés Nézet Keresés Terminál Súgó

```
marty@MartyPC:~/Asztal$ cc msgcreate.c -o main.out
marty@MartyPC:~/Asztal$ ./main.out

Az msgid 2, 2 :
Az 1. msgsnd visszaadott 0-t
A kiküldött üzenet: Az egyik üzenet
A 2. msgsnd visszaadott 0-t
A kiküldött üzenet: Másik üzenet
marty@MartyPC:~/Asztal$ cc msgrcv.c -o main.out
marty@MartyPC:~/Asztal$ ./main.out

Az üzenet ID: 2
Az üzenetek szama: 2
Visszatérés: 17, A fogadott üzenet:Az egyik üzenet

Visszatérés: 15, A fogadott üzenet:Másik üzenet
marty@MartyPC:~/Asztal$ cc msgctl.c -o main.out
marty@MartyPC:~/Asztal$ ./main.out

Visszért: 0
marty@MartyPC:~/Asztal$
```



```
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <string.h>
#include <stdlib.h>
#define MSGKEY 654321L

struct msghbuf {
    long mtype;
    char mtext[512];
} sndbuf, *msgp; // Üzenet buffer és pointere

struct msqid_ds ds, *buf;

int main()
{
    int msgid; // Üzenetsor azonosító
    key_t key; // Kulcs az üzenetsorhoz
    int msgflg; // flag a kreáciohoz, üzenetküldéshez
    int rtn, msgs; // return es méret

    key = MSGKEY;
    msgflg = 00666 | IPC_CREAT;
    msgid = msgget( key, msgflg );
    if ( msgid == -1 ) {
        perror("\n Az msgget rendszerhívás sikertelen!");
        exit(-1);
    }

    msgp = &sndbuf;
    msgp->mtype = 1; // Text típus
    strcpy(msgp->mtext,"Az egyik üzenet");
    msgs = strlen(msgp->mtext) + 1; // Az üzenet hossza

    rtn = msgsnd(msgid,(struct msghbuf *) msgp, msgs, msgflg);

    strcpy(msgp->mtext,"Másik üzenet");
    msgs = strlen(msgp->mtext) + 1;
    rtn = msgsnd(msgid,(struct msghbuf *) msgp, msgs, msgflg);

    if (msgp->mtext == "exit"){ //Ha exit az üzenet, kilép
        exit(0);
    }

    key_t kulcs; // Kulcs az üzenetsorhoz
    int tipus; // Típusa
    int meret; // Méret

    kulcs = MSGKEY; // Beállítom a kulcsot
    msgflg = 00666 | IPC_CREAT | MSG_NOERROR;
```

marty@MartyPC: ~/Asztal

Fájl Szerkesztés Nézet Keresés Terminál Súgó

Válasszon a menüből
0. Üzenet darabszám
1. 1 üzenet kiolvasása
2. Üzenetsor megszüntetése
3. Kilépés
0

Az üzenetek száma: 11
marty@MartyPC:~/Asztal\$./main.out

Az üzenet ID: 0

Válasszon a menüből
0. Üzenet darabszám
1. 1 üzenet kiolvasása
2. Üzenetsor megszüntetése
3. Kilépés
1

A kiküldött üzenet: Másik üzenet
marty@MartyPC:~/Asztal\$

Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó

```
gyak10_2.c x
rtn = msgsnd(msgid,(struct msgb * ) msgp, msgsz, msgflg);

if (msgp->mtext == "exit"){ //Ha exit az üzenet, kilép
    exit(0);
}

key_t kulcs;      // Kulcs az üzenetsorhoz
int tipus;        // Típusa
int meret;        // Méret

kulcs = MSGKEY;   // Beállítom a kulcsot
msgflg = 00666 | IPC_CREAT | MSG_NOERROR;

msgid = msgget(kulcs, msgflg);
if (msgid == -1) {
    perror("\n Az msgget rendszerhívás sikertelen!");
    exit(-1);
}
printf("\n Az üzenet ID: %d\n\n",msgid);

msgp = &sndbuf;
buf = &ds;
meret = 20;
tipus = 0;
rtn = msgctl(msgid,IPC_STAT,buf);

rtn = msgrcv(msgid,(struct msgbuf *)msgp, meret, tipus, msgflg);
rtn = msgctl(msgid,IPC_STAT,buf);

do {
    printf("\n Válasszon a menüből");
    printf("\n 0. Üzenet darabszám");
    printf("\n 1. 1 üzenet kiolvasása");
    printf("\n 2. Üzenetsor megszüntetése");
    printf("\n 3. Kilépés");
    scanf("%d", &msgid);
}while(msgid < 0 && msgid > 1);

switch(msgid)
{
    case 0:
        printf("\n Az üzenetek száma: %ld\n",buf->msg_qnum);
        break;

    case 1:
        printf("\n A kiküldött üzenet: %s\n", msgp->mtext);
        break;

    case 2:
        rtn = msgctl(msgid, IPC_RMID, NULL);
        printf("\n Üzenetsor megszüntetve\n");
        break;

    case 3:
        exit(0);
}
```

marty@MartyPC: ~/Asztal

Fájl Szerkesztés Nézet Keresés Terminál Súgó

Az üzenet ID: 0

Válasszon a menüből

0. Üzenet darabszám
1. 1 üzenet kiolvasása
2. Üzenetsor megszüntetése
3. Kilépés

2

Üzenetsor megszüntetve

marty@MartyPC:~/Asztal\$./main.out

Az üzenet ID: 0

Válasszon a menüből

0. Üzenet darabszám
1. 1 üzenet kiolvasása
2. Üzenetsor megszüntetése
3. Kilépés

3

marty@MartyPC:~/Asztal\$



```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#define SHMKEY 13
#include <string.h>

struct str {
    int lng;
    char text[512-sizeof(int)];
} *segment;

int main()
{
    int shmid, flag, rtn, choose;
    int size = 512;
    key_t key;

    key = SHMKEY;
    flag = 0;

    if((shmid = shmget(key, size, flag)) < 0)
    {
        flag = 00666 | IPC_CREAT;
        if((shmid = shmget(key, size, flag)) < 0)
        {
            perror("\nSikertelen shmget!");
            return -1;
        }
    }

    printf("\nAz azonosító : %d\n", shmid);

    flag = 00666 | SHM_RND;
    segment = (struct str *)shmat(shmid, NULL, flag);

    if(segment == (void *) - 1)
    {
        perror("\n Sikertelen csatlakozas");
        return -1;
    }

    if(strlen(segment->text) > 0)
    {
        printf("\nRégi szöveg : %s", segment->text);
    }

    printf("\nÚj szöveget : ");
    scanf("%s", segment->text);
    printf("\nAz új szöveg: %s \n", segment->text);

    shmdt(segment);
    struct shmid_ds shmid_ds, *buf;
    buf = &shmid_ds;
    flag = 0;

    do {
```

```
marty@MartyPC:~/Asztal$ cc gyak10_3.c -o main.out
marty@MartyPC:~/Asztal$ ./main.out
```

Az azonosító : 32786

Régi szöveg : asd
Új szöveget : █

gyak10_3.c (~/Asztal)

Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó

gyak10_3.c x gyak10_2.c x

```
int((snmid = shmget(key, size, rtag)) < 0)
{
    flag = 00666 | IPC_CREAT;
    if((shmid = shmget(key, size, flag)) < 0)
    {
        perror("\nSikertelen shmget!");
        return -1;
    }

    printf("\nAz azonosító : %d\n", shmid);

    flag = 00666 | SHM_RND;
    segment = (struct str *)shmat(shmid, NULL, flag);

    if(segment == (void *) - 1)
    {
        perror("\n Sikertelen csatlakozas");
        return -1;
    }

    if(strlen(segment->text) > 0)
    {
        printf("\nRégi szöveg : %s", segment->text);
    }

    printf("\nÚj szöveget : ");
    scanf("%s", segment->text);
    printf("\nAz új szöveg: %s \n", segment->text);

    shmdt(segment);
    struct shmid_ds shmid_ds, *buf;
    buf = &shmid_ds;
    flag = 0;

    do {
        printf("Válasszon a menüből\n");
        printf("0. státusz lekérése\n");
        printf("1. szegmens törlése\n");
        scanf("%d", &choose);

    }while(choose != 0 && choose != 1);

    switch(choose){
        case 0:
            rtn = shmctl(shmid, IPC_STAT, buf);
            printf("\nMéret: %ld", buf->shm_segsz);
            printf("\nPID: %d\n", buf->shm_lpid);
            break;

        case 1:
            rtn = shmctl(shmid, IPC_RMID, NULL);
            printf("\nSzegmens megszüntetve!\n");
            break;
    }

    return 0;
}
```

marty@MartyPC: ~/Asztal

Fájl Szerkesztés Nézet Keresés Terminál Súgó

```
Az új szöveg: jó
Válasszon a menüből
0. státusz lekérése
1. szegmens törlése
0

Méret: 512
PID: 6071
marty@MartyPC:~/Asztal$ ./main.out

Az azonosító : 32786

Régi szöveg : jó
Új szöveget : 1

Az új szöveg: 1
Válasszon a menüből
0. státusz lekérése
1. szegmens törlése
1

Szegmens megszüntetve!
marty@MartyPC:~/Asztal$
```

shmcreate.c (~/Asztal)

Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó

shmcreate.c x shmctl.c x shmop.c x

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <stdlib.h>
#define SHMKEY 13

int main()
{
    int shmid;
    key_t kulcs;
    int meret = 512;
    int flag;

    kulcs = SHMKEY;
    flag = 0;
    if((shmid = shmget(kulcs, meret, flag)) < 0)
    {
        //Szegmens létrehozása
        flag = 0066 | IPC_CREAT;
        if((shmid = shmget(kulcs, meret, flag)) < 0)
        {
            perror("\n Sikertelen volt az shmget!\n");
            exit(-1);
        }
    }
    //Ha már van szegmens
    printf("Az azonosító: %d\n", shmid);
    return 0;
}
```

marty@MartyPC: ~/Asztal

Fájl Szerkesztés Nézet Keresés Terminál Súgó

```
marty@MartyPC:~/Asztal$ cc shmcreate.c -o main.out
marty@MartyPC:~/Asztal$ ./main.out
Az azonosító: 32786
marty@MartyPC:~/Asztal$
```

shmctl.c (~/Asztal)

Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó

shmcreate.c shmctl.c shmop.c

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#define SHMKEY 13

int main()
{
    int shmid;
    key_t kulcs;
    int meret = 512;
    int flag, rtn, sm;
    struct shmid_ds shmid_ds, *buf;

    buf = &shmid_ds;
    kulcs = SHMKEY;
    flag = 0;

    if((shmid = shmget(kulcs, meret, flag)) < 0)
    {
        perror("\nSikertelen volt az shmget!\n");
        exit(-1);
    }

    //Státusz lekérdezése
    rtn = shmctl(shmid, IPC_STAT, buf);
    printf("\n Méret: %ld", buf->shm_segsz);
    printf("\n PID: %d\n", buf->shm_lpid);

    exit(0);
}
```

marty@MartyPC:~/Asztal

Fájl Szerkesztés Nézet Keresés Terminál Súgó

```
marty@MartyPC:~/Asztal$ cc shmctl.c -o main.out
marty@MartyPC:~/Asztal$ ./main.out

Méret: 512
PID: 5169
marty@MartyPC:~/Asztal$
```

shmop.c (~/Asztal)

Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó

shmcreate.c x shmctl.c x shmop.c x

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <string.h>
#include <stdlib.h>
#define SHMKEY 13

int main()
{
    int shmid;
    key_t kulcs;
    int meret = 512;
    int flag;

    struct str {
        int lng;
        char text[512-sizeof(int)];
    } *segment;

    kulcs = SHMKEY;
    flag = 0;

    if((shmid = shmget(kulcs, meret, flag)) < 0)
    {
        perror("\n Sikertelen volt az shmget!\n");
        exit(-1);
    }

    flag = 00666 | SHM_RND;
    segment = (struct str *)shmat(shmid, NULL, flag);

    if(segment == (void *) - 1)
    {
        perror("\n Sikertelen csatlakozás!");
        exit(-1);
    }

    if(strlen(segment->text) > 0)
    {
        printf("\n A régi szöveg : %s", segment->text);
    }

    printf("\n Adja meg az új szöveget : ");
    scanf("%s", segment->text);
    printf("\n\n Az új szöveg : %s \n", segment->text);

    shmdt(segment);
    exit(0);
}
```

marty@MartyPC: ~/Asztal

Fájl Szerkesztés Nézet Keresés Terminál Súgó

```
marty@MartyPC:~/Asztal$ cc shmop.c -o main.out
marty@MartyPC:~/Asztal$ ./main.out

A régi szöveg : asd
Adja meg az új szöveget : Működik

Az új szöveg : Működik
marty@MartyPC:~/Asztal$
```