

# Operációs rendszerek Bsc

10. Gyak.

2022. 04. 12.

**Készítette:**

Flaskó Lilian Laura  
Programtervező informatikus  
GCNS8S

**Miskolc, 2022.05**

## 1.feladat:

1. lépés				2. lépés							
MAX IGÉNY				FOGLAL				Kielégítetlen igények (max-foglal)			
R1	R2	R3		R1	R2	R3		R1	R2	R3	
p0	7	5	3	0	1	0	7	0	4	3	
p1	3	2	2	2	0	0	1	2	2	2	
p2	9	0	2	3	0	2	6	0	0	0	
p3	2	2	2	2	1	1	0	1	1	1	
p4	4	3	3	3	3	2	1	0	1	1	
				foglaltak	10	5	5	készlet igény			
				összesen	10	5	7	R1	R2	R3	
				készlet szabad eg. Szám	0	0	2	-7	-4	-1	--
								-1	-2	0	--
								-6	0	2	--
								0	-1	1	--
								-1	0	1	--
				P0 (0,2,0) kérése							
1. lépés				2. lépés							
MAX IGÉNY				FOGLAL				Kielégítetlen igények (max-foglal)			
R1	R2	R3		R1	R2	R3		R1	R2	R3	
p0	7	5	3	0	3	0	7	2	2	3	
p1	3	2	2	2	0	0	1	2	2	2	
p2	9	0	2	3	0	2	6	0	0	0	
p3	2	2	2	2	1	1	0	1	1	1	
p4	4	3	3	0	0	2	4	3	1	1	
				foglaltak	7	4	5	készlet igény			
				összesen	10	5	7	R1	R2	R3	
				készlet szabad eg. Szám	3	1	2	-4	-1	-1	--
								2	-1	0	--
								-3	1	2	--
								3	0	1	--
								-1	-2	1	--
				runnable							
				Nem biztonságos a rendszer							

## 2.feladat

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <string.h>
#include <stdlib.h>
#define MSGKEY 646464L

struct msgbuf1 {
    long mtype;
    char mtext[512];
} sndbuf, *msgp;

struct msqid_ds ds, *buf;

int main()
{
    int msgid;
    key_t key;
    int msgflg;
    int rtn, msgsz;

    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;
    strcpy(msgp->mtext, "Flasko");
    msgsz = strlen(msgp->mtext) + 1;

    rtn = msgsnd(msgid, (struct msgbuf *) msgp, msgsz, msgflg);

    strcpy(msgp->mtext, "Lilian Laura");
    msgsz = strlen(msgp->mtext) + 1;
    rtn = msgsnd(msgid, (struct msgbuf *) msgp, msgsz, msgflg);

    if (msgp->mtext == "exit"){
        exit(0);
    }
}
```

```
gyak10_5.c (~/Asztal/GCNS850412)

Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó

gyak10_4.c x gyak10_5.c x

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 shm_id ds shm_id_ds, *buf;
buf = &shm_id_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("\nMét: %ld", buf->shm_segsize);
    printf("\nPID: %d\n", buf->shm_lpid);
}

C Tabulátorszélesség: 4 80. sor, 2. oszlop BESZ
máj. 9, h. 03:58
```

```
msgcreate.c (~/Asztal/GCNS850412)

Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó

gyak10_4.c x gyak10_5.c x msgcreate.c x

#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <string.h>

// kulcs
#define MSGKEY 646464L

struct msgbuf1 {
    long mtype;
    char mtext[312];
} sendbuf, *msgp;

int main()
{
    int msgid;
    key_t key;
    int msgflg;
    int ret, msgsz;

    key = MSGKEY;
    msgflg = 00666 | IPC_CREAT;
    msgid = msgget(key, msgflg);
    if (msgid == -1) {
        perror(": msgget() nem sikerült!");
        exit(-1);
    }

    printf("Az msgid létrejött: %d, %x\n", msgid, msgid);

    msgp = &sendbuf;
    msgp->mtype = 1;
    strcpy(msgp->mtext, "Flasko");

C Tabulátorszélesség: 4 46. sor, 1. oszlop BESZ
máj. 9, h. 03:58
```

```
msgctl.c (~/Asztal/GCNS850412)

Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó

gyak10_4.c x gyak10_5.c x msgcreate.c x msgctl.c x

#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>

// Kulcs
#define MSGKEY 646464L

int main()
{
    int msgid, msgflg, ret;
    key_t key;

    key = MSGKEY;
    msgflg = 00666 | IPC_CREAT;

    msgid = msgget(key, msgflg);

    ret = msgctl(msgid, IPC_RMID, NULL);
    printf ("Visszatért: %d\n", ret);

    exit(0);
}
```

```
msgrcv.c (~/.Asztal/GCNS850412)

Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó

gyak10_4.c x gyak10_5.c x msgcreate.c x msgctl.c x msgrcv.c x

#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>

// kulcs
#define MSGKEY 646464L

struct msgbuf {
    long mtype;
    char mtext[512];
} recvbuf, *msgp;

struct msqid_ds, *buf;

int main()
{
    int msgid;
    key_t key;
    int mtype, msgflg;
    int ret, msgsz;

    key = MSGKEY;
    msgflg = 000066 | IPC_CREAT | MSG_NOERROR;

    msgid = msgget(key, msgflg);
    if (msgid == -1) {
        perror("msgget() nem sikerült");
        exit(-1);
    }

    printf("Az msgid: %d\n", msgid);

    msgp = &recvbuf;
```

```
named.c (~/.Asztal/GCNS850412)

Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó

gyak10_4.c x gyak10_5.c x msgcreate.c x msgctl.c x msgrcv.c x named.c x

#include <sys/wait.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/file.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

int main()
{
    int fd, ret;
    char buf[32];

    buf[0]=0;
    pid_t cpid;
    ret=mknfifo("tröfkp",000666);
    if (ret == -1) {
        perror("mknfifo()");
        exit(-1);
    }
    cpid = fork();
    if (cpid == -1) {
        perror("fork");
        exit(-1);
    }
    if (cpid == 0) { /*gyerek processz*/
        fd=open("tröfkp",O_RDWR);
        if (fd == -1) {
            perror("open() hiba!");
            exit(-1);
        }
    }
```

```
shmcreate.c (~/.Asztal/GCNS850412)

Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó

gyak10_4.c x gyak10_5.c x msgcreate.c x msgctl.c x msgrcv.c x named.c x shmcreate.c x 2. mentendő dokumentum 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)
    {
        flag = 000666 | IPC_CREAT;
        if ((shmid = shmget(kulcs, meret, flag)) < 0)
        {
            perror("\n Sikertelen volt az shmget!\n");
            exit(-1);
        }
    }
    printf("Azonosító: %d\n", shmid);
    return 0;
}
```

```
shmctl.c (~/Asztal/GCNS850412)

Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó

gyak10_4.c x gyak10_5.c x msgcreate.c x msgctl.c x msgrcv.c x named.c x shmcreate.c x shmctl.c x

#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("\n Sikertelen volt az shmget!\n");
        exit(-1);
    }

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

    exit(0);
}
```

```
GCNS85_name.c (~/Asztal/GCNS850412)

Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó

gyak10_4.c x gyak10_5.c x msgcreate.c x msgctl.c x msgrcv.c x named.c x shmcreate.c x shmctl.c x GCNS85_name.c x

#include <sys/wait.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/file.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

int main()
{
    int fd, ret;
    char buf[32];

    buf[0]=0;
    pid_t cpid;
    ret=mkfifo("tr6fkp",0666);
    if (ret == -1) {
        perror("mkfifo()");
        exit(-1);
    }
    cpid = fork();
    if (cpid == -1) {
        perror("fork");
        exit(-1);
    }
    if(cpid == 0){ /*gyerek processz*/
        fd=open("tr6fkp",O_RDWR);
        if (fd == -1) {
            perror("open() hiba!");
            exit(-1);
        }
    }
```

```
GCNS85_unname.c (~/Asztal/GCNS850412)

Fájl Szerkesztés Nézet Keresés Eszközök Dokumentumok Súgó

gyak10_4.c x gyak10_5.c x msgcreate.c x msgctl.c x msgrcv.c x named.c x shmcreate.c x shmctl.c x GCNS85_name.c x GCNS85_unname.c x

#include <sys/wait.h>
#include <unistd.h>

int main(){
    int fd[2];
    pid_t wpid;
    int status = 0;
    if (pipe(fd) == -1) {
        fprintf(stderr, "Pipe failed");
        return 1;
    }

    pid_t p = fork();
    if (p < 0) {
        fprintf(stderr, "fork Failed");
        return 1;
    } else if (p>0){
        close(fd[1]);
        wpid = wait(&status);
        char input_str[30];
        read(fd[0],input_str,30);
        printf("Kapott String: %s",input_str);
    } else {
        close(fd[0]);
        char output_str[30] = "Flasko Lillian GCNS850";
        write(fd[1],output_str,strlen(output_str)+1);
        close(fd[1]);
        exit(0);
    }

    return 0;
}
```

