Operációs Rendszerek BSc

10.gyak.

2021.04.21

Készítette:

Orosz Dániel Bsc

Üzemmérnök-informatikus

C5S7FM

Miskolc, 2021

„1. Az előadáson bemutatott mintaprogram alapján készítse el a következő feladatot.

Adott egy rendszerbe az alábbi erőforrások: R (R1: 10; R2: 5; R3: 7)

A rendszerbe 5 processz van: P0, P1, P2, P3, P4

Kérdés: Kielégíthető-e P4 (3,3,0) ill. P0 (0,2,0) kérése úgy, hogy biztonságos legyen, holtpontmentesség szempontjából a rendszer - a következő kiinduló állapot alapján.

Igazolja a processzek végrehajtásának sorrendjét – számolással.”

R1 : 10	R2 : 5	R3 : 7									
P4 (3, 3, 0)											
P0 (0, 2, 0)											
	1. lépés			2. lépés							
	Max igény			Foglal				Igény			
	R1	R2	R3	R1	R2	R3		R1	R2	R3	R1 erőforrások száma: 10 - 7 = 3
P0	7	5	3	0	1	0		7	4	3	
P1	3	2	2	2	0	0		1	2	2	R2 erőforrások száma: 5 - 2 = 3
P2	9	0	2	3	0	2		6	0	0	
P3	2	2	2	2	1	1		0	1	1	R3 erőforrások száma: 7 - 5 = 2
P4	4	3	3	0	0	2		4	3	1	
				7	2	5					Szabad erőforrások száma:
MAXr = (10, 5, 7)											Készlet: (3, 3, 2)
KÉSZLET / SZABAD = (10, 5, 7) - (7, 2, 5) = (3, 3, 2)											
Vizsgálat, hogy Igény <= KÉSZLET/SZABAD											
P4 SZABAD = (3, 3, 2) + (3, 3, 0) = (6, 6, 2) <- új készlet											
P0 SZABAD = (6, 6, 2) + (0, 2, 0) = (6, 8, 2)											

A P4-es processzre teljesül a feltéttel, hogy kevesebb erőforrást igényel mint amennyi szabad van. A P0-ás processzre nem teljesül ez a feltétel ezért nem lesz biztonságos a rendszer.

2. Gyakorló feladat: Először tanulmányozzák Vadász Dénes: Operációs rendszer jegyzet, a témához kapcsolódó fejezetét (5.3)., azaz

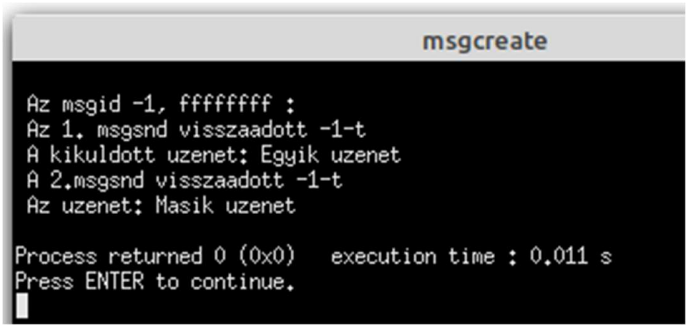
Írjanak három C nyelvű programot, ahol készít egy üzenetsort és ebbe két üzenetet tesz bele – msgcreate.c, majd olvassa ki az üzenetet - msgrcv.c, majd szüntesse meg az üzenetsort (takarít) - msgctl.c.

A futtatás eredményét is tartalmazza a jegyzőkönyv.

Mentés: msgcreate.c; msgrcv.c; msgctl.c.

msgcreate.c:

```c
int main()
{
    int id;
    key_t key;
    int msgflag;
    int rtn, msgsize;
    int msgid;
    int msgflg;

    key = MSGKEY;
    msgflag = 00666 | IPC_CREAT;
    msgid = msgget( key, msgflg);
    if ( id == -1) {
        perror("\n Hiba az msgget hivasnal\n");
        exit(-1);
    }

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

    msgp = &sndbuf;
    msgp->mtype = 1;
    strcpy(msgp->mtext,"Egyik uzenet");
    msgsize = strlen(msgp->mtext) + 1;

    rtn = msgsnd(id,(struct msgbuf *) msgp, msgsize, msgflag);
    printf("\n Az 1. msgsnd visszaadott %d-t", rtn);
    printf("\n A kikuldott uzenet: %s", msgp->mtext);

    strcpy(msgp->mtext,"Masik uzenet");
    msgsize = strlen(msgp->mtext) + 1;
    rtn = msgsnd(id,(struct msgbuf *) msgp, msgsize, msgflag);
    printf("\n A 2.msgsnd visszaadott %d-t", rtn);
    printf("\n Az uzenet: %s", msgp->mtext);
    printf("\n");

    exit (0);
}
```



msgrcv.c:

```c
struct msgbuf1 {
    long mtype;
    char mtext[512];
} rcvbuf, *msgp;

 struct msqid_ds ds, *buf;

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

    key = MSGKEY;
    msgflg = 00666 | IPC_CREAT | MSG_NOERROR;

    msgid = msgget( key, msgflg);
    if ( msgid == -1) {
        perror("\n A hivas sikertelen volt \n");
        exit(-1);
    }
    printf("\n Az msgid: %d",msgid);

    msgp = &rcvbuf;
    buf = &ds;
    msgsz = 20;
    mtype = 0;
    rtn = msgctl(msgid,IPC_STAT,buf);
    printf("\n Az uzenetek szama: %d",buf->msg_qnum);

    while (buf->msg_qnum) {
        rtn = msgrcv(msgid,(struct msgbuf *)msgp, msgsz, mtype, msgflg);
        printf("\n Az rtn: %d,  a vett uzenet:%s\n",rtn, msgp->mtext);
        rtn = msgctl(msgid,IPC_STAT,buf);
    }

    exit (0);
}
```

```
msgrvc

 Az msgid: 0
 Az uzenetek szama: 0
Process returned 0 (0x0)   execution time : 0.010 s
Press ENTER to continue.
```

msgctl.c:

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

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

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

    exit (0);
}
```

```
msgctl

Visszatert: 0

Process returned 0 (0x0)   execution time : 0.002 s
Press ENTER to continue.
```

2a. Írjon egy C nyelvű programot, melyben

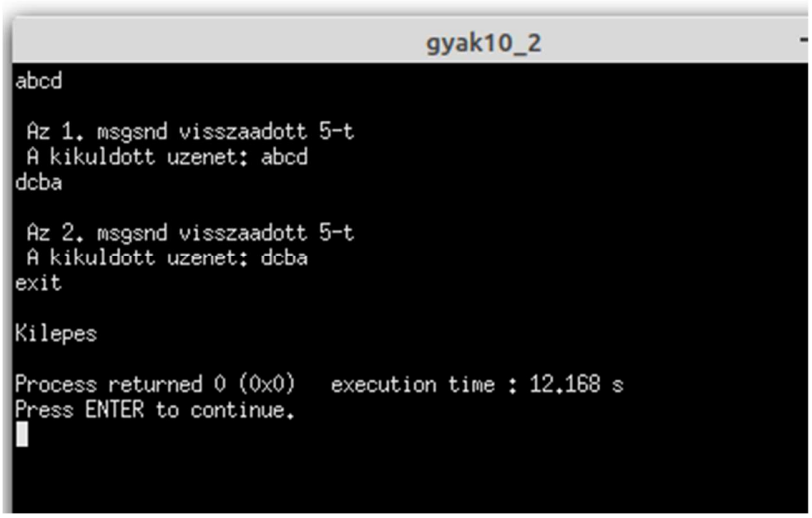
• az egyik processz létrehozza az üzenetsort, és szövegeket küld bele, exit üzenetre kilép,

• másik processzben lehet választani a feladatok közül: üzenetek darabszámának lekérdezése, 1 üzenet kiolvasása, összes üzenet kiolvasása, üzenetsor megszüntetése, kilépés.

Mentés: gyak10_2.c

A futtatás eredményét is tartalmazza a jegyzőkönyv.

```c
int main()
{
    int id;
    key_t key;
    int flag;
    int rtn, size;
    int ok = 1, count = 1;

    char teszt[256];
    key = MSGKEY;
    flag = 00666 | IPC_CREAT;
    id = msgget( key, flag);
    if ( id == -1) {
        perror("Az msgget hivas sikertelen volt");
        exit(-1);
    }

    do {
        scanf("%s", teszt);
        msgp = &sndbuf;
        msgp->mtype = 1;
        strcpy(msgp->mtext,teszt);
        size = strlen(msgp->mtext) + 1;

        if(strcmp("exit",teszt) != 0) {
            rtn = msgsnd(id,(struct msgbuf *) msgp, size, flag);
            printf("\n Az %d. msgsnd visszaadott %d-t", count);
            printf("\n A kikuldott uzenet: %s\n", msgp->mtext);
            count++;
        }
        else
        {
            ok = 0;
            printf("\nKilepes\n");
        }

    } while(ok == 1);
    return 0;
}
```


gyak10_2
abcd
Az 1. msgsnd visszaadott 5-t
A kikuldott uzenet: abcd
dcba
Az 2. msgsnd visszaadott 5-t
A kikuldott uzenet: dcba
exit
Kilepes
Process returned 0 (0x0) execution time : 12.168 s
Press ENTER to continue.