

# **Operációs rendszerek BSc**

11.gyak.

2021. 04.21.

**Készítette:**

Tóth József BProf  
Üzemtechnikus-  
informatikus alapszak  
WI2GDP

**Miskolc, 2021**

- 1. feladat** - Adott egy rendszerbe az alábbi erőforrások: R (R1: 10; R2: 5; R3: 7) A rendszerbe 5 processz van: P0, P1, P2, P3, P4  
 Kérdés: Teljesíthető-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.

Az összes osztály -erőforrások száma: (10, 5, 7)											
Kiinduló állapot											
	1. lépés				2. lépés						
	MAX. Igény				Foglal				MÉG		
	R1	R2	R3		R1	R2	R3		R1	R2	R3
P0	7	5	3		0	1	0		7	4	3
P1	3	2	2		2	0	0		1	2	2
P2	9	0	2		3	0	2		6	0	0
P3	2	2	2		2	1	1		0	1	1
P4	4	3	3		0	0	2		4	3	1
MAXr = [10, 5, 7]											
SZABAD = [10, 5, 7] - [7, 2, 5] = [3, 3, 2]											
Most megnézzük, hogy a MÉG[i] <= SZABAD feltétel igaz-e a P4 és a P0 processzekre											
P4 SZABAD = [3,3,2] + [3,3,0] = [6,6,2]											
P0 SZABAD = [6,6,2] + [0,2,0] = [6,8,2]											

Egyedül a P4-es processzre volt igaz a feltétel, hogy kevesebb erőforrást kér, mint amennyi szabad. A P0-ás processz azonban nem teljesül, így ezek feltételek alapján nem lesz biztonságos a rendszer.

**2. 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:**

```
int main()
{
    int id;        /* üzenetsor azonosító */
    key_t key;     /* kulcs az üzenetsorhoz */
    int flag;      /* flag a kreációhoz, üzenetküldéshez */
    int rtn, size; /* return és méret */

    key = MSGKEY;
    flag = 00666 | IPC_CREAT;
    msgid = msgget( key, msgflag);
    if ( id == -1) {
        perror( "Az msgget hívás nem valósult meg");
        exit(-1);
    }

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

    msgp = &sndbuf;
    msgp->mtype = 1;          /* típus = text */
    strcpy(msgp->mtext,"Egyik üzenet");
    size = strlen(msgp->mtext) + 1; /* az üzenet hossza */

    /* és elküldöm: */

    rtn = msgsnd(id,(struct msgbuf *) msgp, size, flag);
    printf( format: "\n Az 1. msgsnd visszaadott %d-t", rtn);
    printf( format: "\n A kiküldött üzenet: %s", msgp->mtext);

    strcpy(msgp->mtext,"Masik üzenet");
    size = strlen(msgp->mtext) + 1;
    rtn = msgsnd(id,(struct msgbuf *) msgp, size, flag);
    printf( format: "\n A 2. msgsnd visszaadott %d-t", rtn);
    printf( format: "\n Az üzenet: %s", msgp->mtext);
}
```

```

joseph@joseph-virtual-machine:~/Documents/WI2GDPGyak/WI2GDPOsGyak/WI2GDP_0421$ ./create

Az msgid 2, 2 :
Az 1. msgsnd visszaadott 0-t
A kikuldott uzenet:Egyik uzenet
A 2. msgsnd visszaadott 0-t
A kikuldott uzenet:Masik uzenet
joseph@joseph-virtual-machine:~/Documents/WI2GDPGyak/WI2GDPOsGyak/WI2GDP_0421$

```

### msgrcv.c:

```

main()
{
    int msgid;      /* uzenetsor azonosito */
    key_t key;      /* kulcs az uzenetsorhoz */
    int mtype, msgflg; /* tipus, flag */
    int rtn, msgsz;  /* return es meret */

    key = MSGKEY;    /* beallitom a kulcsot */
    msgflg = 00666 | IPC_CREAT | MSG_NOERROR;

    msgid = msgget( key, msgflg);
    if ( msgid == -1) {
        perror( $: "\n A hivas nem valosult meg");
        exit(-1);
    }
    printf( format: "\n Az msgid: %d",msgid);

    msgp = &rcvbuf; /* uzenetfogado buffer cime */
    buf = &ds;       /* uzenetsor jellemzo adataihoz */
    msgsz = 20;       /* max hossz */
    mtype = 0;        /* minden tipust varok */
    rtn = msgctl(msgid,IPC_STAT,buf); /* uzenetsor adatokat lekerdezem */
    printf( format: "\n Az uzenetek szama: %d",buf->msg_qnum);

    while (buf->msg_qnum) { /* van-e uzenet?*/
        /* veszem a kovetkezo uzenetet: */
        rtn = msgrcv(msgid,(struct msgbuf *)msgp, msgsz, mtype, msgflg);
        printf( format: "\n Az rtn: %d, a vett uzenet:%s\n",rtn, msgp->mtext);
        rtn = msgctl(msgid,IPC_STAT,buf); /* uzenetsor adatokat lekerdezem,
                                           benne azt is, hany uzenet van meg */
    }

    exit (0);
}

```

```

joseph@joseph-virtual-machine:~/Documents/WI2GDPGyak/WI2GDPOsGyak/WI2GDP_0421$ ./read

Az msgid: 3
Az uzenetek szama: 2
Az rtn: 13, a vett uzenet:Egyik uzenet

Az rtn: 13, a vett uzenet:Masik uzenet
joseph@joseph-virtual-machine:~/Documents/WI2GDPGyak/WI2GDPOsGyak/WI2GDP_0421$

```

***msgctl.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, buf: NULL); /* torlom az uzenetsort */
    printf ( format: "\n Vissztert: %d\n", rtn);

    exit (0);
}

joseph@joseph-virtual-machine:~/Documents/WI2GDPGyak/WI2GDPOsGyak/WI2GDP_0421$ ./clear
Vissztert: 0joseph@joseph-virtual-machine:~/Documents/WI2GDPGyak/WI2GDPOsGyak/WI2GDP_0421$
```

### **3. feladat - 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***

```

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( "\n Az msgget hivas nem valosult meg");
        exit(-1);
    }

    do {
        scanf( format: "%s", teszt);
        msgp = &sndbuf;
        msgp->mtype = 1;          /* tipus = text */
        strcpy(msgp->mtext,teszt);
        size = strlen(msgp->mtext) + 1; /* az uzenet hossza */

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

```

```

joseph@joseph-virtual-machine:~/Documents/WI2GDPGyak/WI2GDP0sGyak/WI2GDP_0421$ ./ask
asd

Az 1. msgsnd visszaadott 4-t
A kikuldott uzenet: asd
ds

Az 2. msgsnd visszaadott 3-t
A kikuldott uzenet: ds
exit

Kilepes
joseph@joseph-virtual-machine:~/Documents/WI2GDPGyak/WI2GDP0sGyak/WI2GDP_0421$

```