

Jegyzőkönyv

Operációs rendszerek BSC

2022. tavasz féléves feladat


Készítette: **Mezei Gréta**

Neptunkód: **WY9VBO**

1. feladat

Írjon C nyelvű programokat, ami létrehoz egy üzenetsort. Ebbe az üzenetsorba betesz egy másodpercenként egy változót. Ennek a változónak a kezdőértéke legyen az aktuális processz azonosítója. A változó mindig beírás után növekedjen 2-vel. A másik program pedig: kiolvas.

Először a program kiír, majd elküldi az üzenetet az olvas programnak.

 jerry.iit.uni-miskolc.hu - PuTTY

```
GNU nano 2.7.4
// ír
#include <stdio.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#define MAX 10

// Struktúra az üzenetsorhoz
struct msg_buffer {
    long msg_tipus;
    char msg_uzenet[100];
} uzenet;

int main()
{
    key_t kulcs;
    int msgid, szam;

    // Kulcsgenerálás
    kulcs = ftok("beadando", 10);

    // Az msget() megcsinálja az üzenetsort,
    // az msgid pedig ennek az üzenetsornak az azonosítóját tárolja
    msgid = msgget(kulcs, 0666 | IPC_CREAT);
    uzenet.msg_tipus = 1;

    szam=getpid();
    printf("Kezdoérték: %d\n\n", szam);

    while (szam < 100000) {
        printf("Adat irasa folyamatban...\n");
        sprintf(uzenet.msg_uzenet, "%d", szam);

        // Az msgsnd() elküldi az üzenetet
        msgsnd(msgid, &uzenet, sizeof(uzenet), 0);
    }
}
```

```

        szam=szam+2;
        sleep(1);
    }

    return 0;
}

```

Az olvas program pedig beolvassa, majd kiírja a kapott üzenetet.

```

// Olvas
#include <stdio.h>
#include <sys/ipc.h>
#include <sys/msg.h>
// Struktúra az üzenetsorhoz
struct msg_buffer {
    long msg_típus;
    char msg_uzenet[100];
} uzenet;

int main()
{
    key_t kulcs;
    int msgid, beszam;

    // Kulcsgenerálás
    kulcs = ftok("beadando", 10);

    // Az msgget() megcsinálja az üzenetsort,
    // az msgid pedig ennek az üzenetsornak az azonosítóját tárolja
    msgid = msgget(kulcs, 0666 | IPC_CREAT);

    // A kapott üzenetet (string) számmá alakítjuk (int)
    beszam=atoi(uzenet.msg_uzenet);

    // Amíg az üzenet kisebb mint 100000, folytatjuk az olvasást (ugyan ez van az írásnál is, így egyszerre fejezik be)
    while (beszam < 100000) {

        beszam=atoi(uzenet.msg_uzenet);

        // Az msgrcv() beolvassa az adott azonosítójú üzenetsorból az üzenetet
        msgrcv(msgid, &uzenet, sizeof(uzenet), 1, 0);

        // Kiíratjuk a kapott üzenetet
        printf("Beerkezett adat: %s \n", uzenet.msg_uzenet);

        beszam=atoi(uzenet.msg_uzenet);

        // Az msgrcv() beolvassa az adott azonosítójú üzenetsorból az üzenetet
        msgrcv(msgid, &uzenet, sizeof(uzenet), 1, 0);

        // Kiíratjuk a kapott üzenetet
        printf("Beerkezett adat: %s \n", uzenet.msg_uzenet);

    }

    // Amint a while ciklus befejeződik, töröljük az üzenetsort
    msgctl(msgid, IPC_RMID, NULL);

    return 0;
}

```

Az eredmény:

[illegible]

2.feladat

Adott egy igény szerinti lapozást használó számítógéprendszer, melyben futás közben egy processz számára a következő laphivatkozással lehet hivatkozni: 1,2,3,4,0,2,5,1,2,3,4,5,1,2.

Memóriakeret: 3, ill. 4 memóriakeret. Készítse el a laphivatkozások betöltését külön-külön táblázatba 3, ill. 4 memóriakeret esetén.

Mennyi laphiba keletkezik az alábbi algoritmusok esetén. SC, OPT?

Hasonlítsa össze és magyarázza az eredményeket!

[illegible]

18																
19	SC	1	2	3	4	0	2	5	1	2	3	4	5	1	2	
20	1.lap	1,0	1,0	1,0	1,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	5,0	5,0	5,0	
21	2.lap		2,0	2,0	2,0	2,0	2,1	2,1	2,1	2,1	2,0	2,0	2,0	1,0	1,0	
22	3.lap			3,0	3,0	3,0	3,0	5,0	5,0	5,0	3,0	3,0	3,0	3,0	2,0	
23	4.lap				4,0	4,0	4,0	4,0	1,0	1,0	1,0	4,0	4,0	4,0	4,0	
24	Laphiba	1	1	1	1	1		1	1		1	1	1	1	1	
25	Ö. Laphiba	12														
26																
27																
28	OPT	1	2	3	4	0	2	5	1	2	3	4	5	1	2	
29	1.lap	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
30	2.lap		2	2	2	2	2	2	2	2	2	2	2	2	2	
31	3.lap			3	3	3	3	3	3	3	3	4	4	4	4	
32	4.lap				4	0	0	5	5	5	5	5	5	5	5	
33	Laphiba	1	1	1	1	1		1				1				
34	Ö. Laphiba	7														
35																

SC-nél mind a két esetben 12 laphiba
volt, míg OPT-nál 3 memóriakeret esetén
9, 4 memóriakeretnél pedig 7.