

1. Feladat

Szabad: 30k, 35k, 15k, 25k, 75k, 45k							
Igény: 39k, 40k, 33k, 20k, 21k							
Foglalási igény		Memória terület, szabad terület					
		30	35	15	25	75	45
	39	30	35	15	25	75	45
	40	30	35	15	25	75	45
	33	30	35	15	25	75	45
	20	30	35	15	25	75	45
	21	30	35	15	25	75	45
First Fit							
Foglalási igény		Memória terület, szabad terület					
		30	35	15	25	75	45
	39	30	35	15	25	36,39	45
	40	30	35	15	25	75	5, 40
	33	30	2, 33	15	25	75	45
	20	10, 20	35	15	25	75	45
	21	30	35	15	4, 21	75	45
Next Fit							
Foglalási igény		Memória terület, szabad terület					
		30	35	15	25	75	45
	39	30	35	15	25	39, 36	45
	40	30	35	15	25	75	40, 5
	33	30	33, 2	15	25	75	45
	20	30	35	15	20, 5	75	45
	21	30	35	15	25	21, 15	45
Best Fit							
Foglalási igény		Memória terület, szabad terület					
		30	35	15	25	75	45
	39	30	35	15	25	75	39, 6
	40	30	35	15	25	40, 35	45
	33	30	33, 2	15	25	75	45
	20	30	35	15	20, 5	75	45
	21	21, 9	35	15	25	75	45
Worst Fit							
Foglalási igény		Memória terület, szabad terület					
		30	35	15	25	75	45
	39	30	35	15	25	39, 36	45
	40	30	35	15	25	75	40, 5
	33	30	35	15	25	33, 3	40, 5
	20	30	20, 15	15	25	33, 3	40, 5
	21	21, 9	35	15	25	33, 3	40, 5

2. Feladat

```
mozes@mozes-VirtualBox:~/Asztal/Csdolgozok/04_29$ cc semset.c -o semset.out
mozes@mozes-VirtualBox:~/Asztal/Csdolgozok/04_29$ ./semset.out
```

```
Az azonosító: 0
Kérem a semval erteket: 3
```

```
Visszatérési érték: 0
Semval értéke: 3
```

```
#include <stdlib.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#define SEMKEY 0x12

    int semid,
        nsems,
        semnum,
        rtn;

    int semflg;
    struct sembuf sembuf, *sop;
    union semun;

    int cmd;

int main()
{
    int arg;

    nsems = 1;
    semflg = 00666 | IPC_CREAT;
    semid = semget (SEMKEY, nsems, semflg);
    if (semid < 0 ) {perror("semget() hiba!\n"); exit(0);}
    else printf("\n Az azonosító: %d\n",semid);

    printf ("Kérem a semval erteket: ");
    semnum = 0;
    cmd = SETVAL;
    scanf("%d", &arg);
    rtn = semctl(semid, semnum, cmd, arg);
    printf("\nVisszatérési érték: %d\nSemval értéke: %d\n",rtn, arg);

    return 0;
}
```

```
mozes@mozes-VirtualBox:~/Asztal/Csdolgozok/04_29$ cc semval.c -o semval.out
mozes@mozes-VirtualBox:~/Asztal/Csdolgozok/04_29$ ./semval.out
azonosító: 0
semval kiolvasott értéke: 3
mozes@mozes-VirtualBox:~/Asztal/Csdolgozok/04_29$
```

```
Megnyitás  semval.c  Mentés
~/Asztal/Csdolgozok/04_29

3 #include <sys/types.h>
4 #include <sys/ipc.h>
5 #include <sys/sem.h>
6
7 #define SEMKEY 0x12|
8
9     int semid,nsems,rtn;
10    int semflg;
11    struct sembuf sembuf, *sop;
12    union semun arg;
13
14    union semun {
15
16        int val;
17        struct semid_ds *buf;
18        unsigned short int *array;
19    };
20
21    int cmd;
22
23 int main()
24
25 {
26     nsems = 1;
27     semflg = 00666 | IPC_CREAT;
28     semid = semget (SEMKEY, nsems, semflg);
29     if (semid < 0 ) {perror("semget() hiba!\n"); exit(0);}
30     else printf("azonosító: %d\n",semid);
31     cmd = GETVAL;
32     rtn = semctl(semid,0, cmd, NULL);
33     printf("semval kiolvasott értéke: %d ",rtn);
34     printf("\n");
35     return 0;
36 }
37
```

```
mozes@mozes-VirtualBox:~/Asztal/Csdolgok/04_29$ cc semup.c -o semup.out
mozes@mozes-VirtualBox:~/Asztal/Csdolgok/04_29$ ./semup.out
Azonosító: 0

Visszatérési érték: 0
mozes@mozes-VirtualBox:~/Asztal/Csdolgok/04_29$
```

```
Megnyitás  semup.c  Mentés
~/Asztal/Csdolgok/04_29

1#include <stdlib.h>
2#include <stdio.h>
3#include <sys/types.h>
4#include <sys/ipc.h>
5#include <sys/sem.h>
6
7#define SEMKEY 0x12
8
9    int semid, nsems, rtn;
10    unsigned nsops;
11    int semflg;
12    struct sembuf sembuf, *sop;
13
14int main()
15{
16    nsems = 1;
17    semflg = 00666 | IPC_CREAT;
18    semid = semget (SEMKEY, nsems, semflg);
19    if (semid < 0 ) {perror("semget() hiba!\n"); exit(0);}
20    else printf("Azonosító: %d\n",semid);
21    printf ("\n");
22
23    nsops = 1;
24    sembuf.sem_num = 0;
25    sembuf.sem_op = 1;
26    sembuf.sem_flg = 0666;
27    sop = &sembuf;
28    rtn = semop(semid, sop, nsops);
29    printf("Visszatérési érték: %d\n",rtn);
30
31    return 0;
32}
33
34
```

```
mozes@mozes-VirtualBox:~/Asztal/Csdolgok/04_29$ cc semkill.c -o semkill.out
mozes@mozes-VirtualBox:~/Asztal/Csdolgok/04_29$ ./semkill.out
semid értéke: 1
Kill visszatérés: 0
mozes@mozes-VirtualBox:~/Asztal/Csdolgok/04_29$
```

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#define SEMKEY 0x10

int semid, nsems, rtn;
int semflg;
struct sembuf sembuf, *sop;
union semun {
    int val;
    struct semid_ds *buf;
    unsigned short int *array;
};
int cmd;

int main()
{
    int arg;

    nsems = 1;
    semflg = 00666 | IPC_CREAT;
    semid = semget (SEMKEY, nsems, semflg);
    if (semid < 0 ) {perror("semget() hiba!\n"); exit(0);}
    else printf("semid értéke: %d\n",semid);
    cmd = IPC_RMID;
    rtn = semctl(semid, 0, cmd, arg);
    printf("Kill visszatérés: %d\n",rtn);

    return 0;
}
```

2.a Feladat

```
mozes@mozes-VirtualBox:~/Asztal/Csdolgok/04_29$ cc -pthread gyak11_2.c -o gyak11_2.out3
mozes@mozes-VirtualBox:~/Asztal/Csdolgok/04_29$ ./gyak11_2.out
bash: ./gyak11_2.out: Nincs ilyen fájl vagy könyvtár
mozes@mozes-VirtualBox:~/Asztal/Csdolgok/04_29$ ./gyak11_2.out3
A pдем: 4805
Várákozás...
A pдем: 4805
Várákozás...
A pдем: 4805
Várákozás...
mozes@mozes-VirtualBox:~/Asztal/Csdolgok/04_29$
```

```
Megnyitás  ▾  [+]  gyak11_2.c  ~/Asztal/Csdolgok/04_29  Mentés  ≡
1 #include <stdio.h>
2 #include <pthread.h>
3 #include <semaphore.h>
4 #include <unistd.h>
5
6 sem_t m;
7
8 void* thread(void* arg)
9
10 {
11
12     sem_wait(&m);
13     printf("A pдем: %d\nVárákozás...\n", getpid());
14     sleep(3);
15     sem_post(&m);
16 }
17
18 int main()
19
20 {
21     sem_init(&m, 0, 1);
22     pthread_t t1,t2,t3; // 3 feladat
23     pthread_create(&t1, NULL,thread,NULL);
24     pthread_create(&t2, NULL,thread,NULL);
25     pthread_create(&t3, NULL,thread,NULL);
26
27     pthread_join(t1, NULL);
28     pthread_join(t2, NULL);
29     pthread_join(t3, NULL);
30
31     sem_destroy(&m);
32
33     return 0;
34 }
```