

1,

First fit		Memóriaterület-szabad terület						
Foglalt igény		30	35	15	25	75	45	
39	30	30	35	15	25	75	45	
40	30	30	35	15	25	75	45	
33	30	30	35	15	25	75	45	
20	30	30	35	15	25	75	45	
21	30	30	35	15	25	75	45	

worst fit		Memóriaterület-szabad terület						
Foglalt igény		30	35	15	25	75	45	
39	30	30	35	15	25	39 36	45	
40	30	30	35	15	25	75	40 5	
33	30	30	35	15	25	33 3	40 5	
20	30	30	20 15	15	25	33 3	40 5	
21	21 9	35	15	25	33 3	40 5		

1 lefoglalja a legnagyobbat
mindig a legnagyobbat foglalja le

next fit		Memóriaterület-szabad terület						
Foglalt igény		30	35	15	25	39 36	75	45
39	30	30	35	15	25	39 36	75	45
40	30	30	35	15	25	75	40 5	
33	30	30	33 2	15	25	75	45	
20	30	30	35	15	20 5	75	45	
21	30	30	35	15	25	21 15	45	

keres egyet lefoglal majd utána elindul
onnan tovább és ha elfogy a szabad terület
és újra kezd

best fit		Memóriaterület-szabad terület						
Foglalt igény		30	35	15	25	75	45	
39	30	30	35	15	25	75	39 6	
40	30	30	35	15	25	40 35	45	
33	30	33 2	15	25	75	45		
20	30	30	35	15	20 5	75	45	
21	21 9	35	15	25	75	45		

azokat keresi ahol a legkevesebb
maradék szabad területet hagyja

2,

semset.c

```

oraveczaron@oraveczaron-VirtualBox: ~/Desktop
File Edit View Search Terminal Help
oraveczaron@oraveczaron-VirtualBox:~$ cc semset.c -o main.out
cc: error: semset.c: No such file or directory
cc: fatal error: no input files
compilation terminated.
oraveczaron@oraveczaron-VirtualBox:~$ cd Desktop
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ cc semset.c -o main.out
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ ./main.out
Az ID avagy azonosító: 0
Kérem a semval értéket: 128

Visszatérési érték: 0 , semval érték: 128
oraveczaron@oraveczaron-VirtualBox:~/Desktop$

```

```

File Edit View Search Tools Documents Help
semset.c x semval.c x semkill.c x semup.c x
#include <stdlib.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#define SEMKEY 0x12

int semID;
int nsems;
int semnum;
int r;
int semflag;
int cmd;

struct sembuf sembuf, *sop;
union semun;

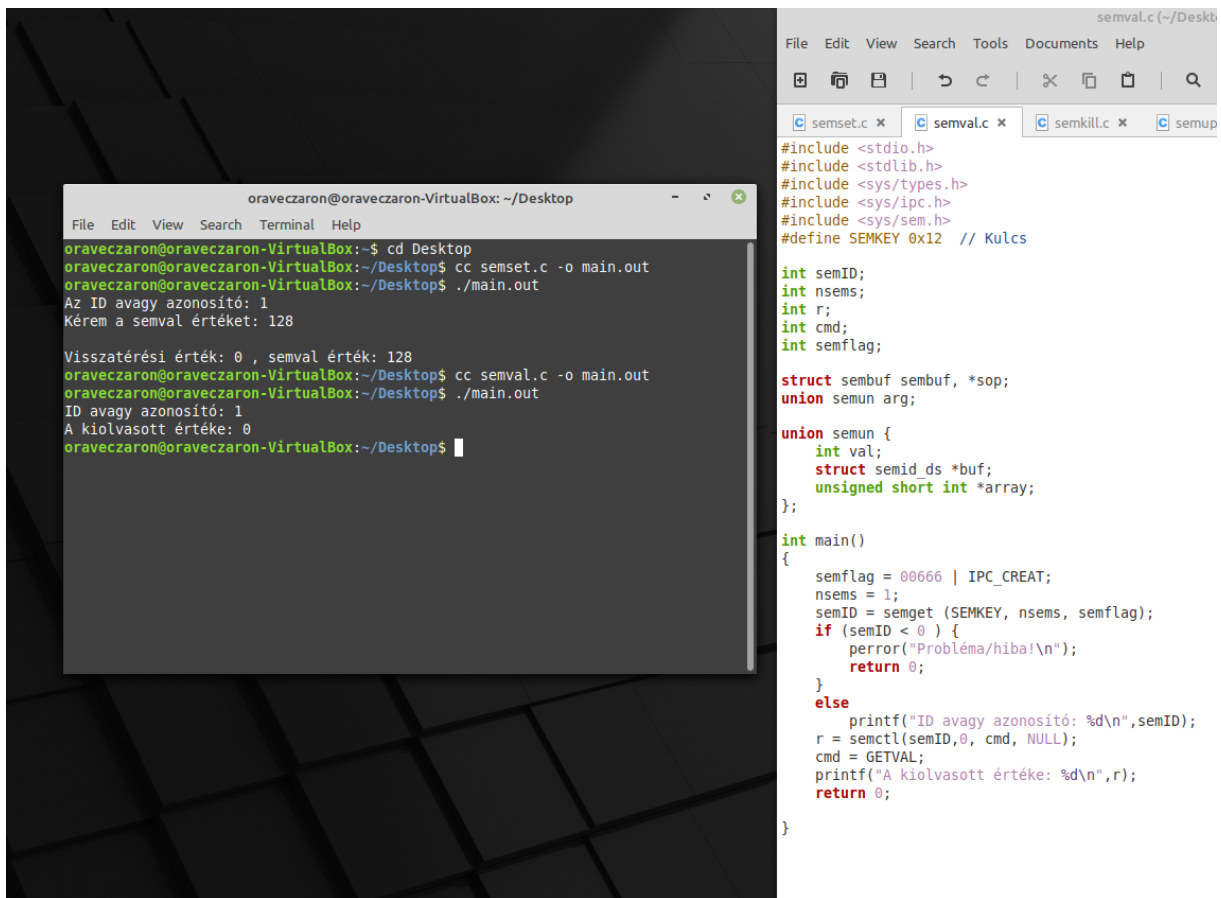
int main()
{
    int arg;
    nsems = 1;
    semflag = 00666 | IPC_CREAT;
    semID = semget (SEMKEY, nsems, semflag);
    if (semID < 0 ) {
        perror("Probléma/hiba!\n");
        return 0;
    }
    else
        printf("Az ID avagy azonosító: %d\n",semID);
    printf ("Kérem a semval értéket: ");

    semnum = 0;
    cmd = SETVAL;
    scanf("%d", &arg);
    r = semctl(semID, semnum, cmd, arg);
    printf("\nVisszatérési érték: %d , semval érték: %d\n",r, arg);

    return 0;
}

```

semval.c



The image shows a terminal window and a code editor side-by-side. The terminal window, titled 'oraveczaron@oraveczaron-VirtualBox: ~/Desktop', shows the execution of the 'semval.c' program. The user enters 'cd Desktop', 'cc semset.c -o main.out', and './main.out'. The program prompts for an ID (128) and returns a value of 0. The user then compiles 'semval.c' and runs it, which prompts for an ID (1) and returns a value of 0. The code editor, titled 'semval.c (~/Desk...', shows the source code of 'semval.c'. It includes headers for stdio, stdlib, sys/types, sys/ipc, and sys/sem. It defines SEMKEY as 0x12. The code declares variables for semID, nsems, r, cmd, and semflag. It defines a struct sembuf and a union semun. The main function sets semflag to 00666 | IPC_CREAT, nsems to 1, and semID to semget(SEMKEY, nsems, semflag). It checks if semID is less than 0 and returns 0 if so. Otherwise, it prints the ID, sets r to semctl(semID, 0, cmd, NULL), sets cmd to GETVAL, and prints the value of r.

```
oraveczaron@oraveczaron-VirtualBox: ~/Desktop
File Edit View Search Terminal Help
oraveczaron@oraveczaron-VirtualBox:~$ cd Desktop
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ cc semset.c -o main.out
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ ./main.out
Az ID avagy azonosító: 1
Kérem a semval értéket: 128

Visszatérési érték: 0 , semval érték: 128
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ cc semval.c -o main.out
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ ./main.out
ID avagy azonosító: 1
A kiolvasott értéke: 0
oraveczaron@oraveczaron-VirtualBox:~/Desktop$
```

```
semval.c (~/Desk...
File Edit View Search Tools Documents Help
semset.c x semval.c x semkill.c x semup
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#define SEMKEY 0x12 // Kulcs

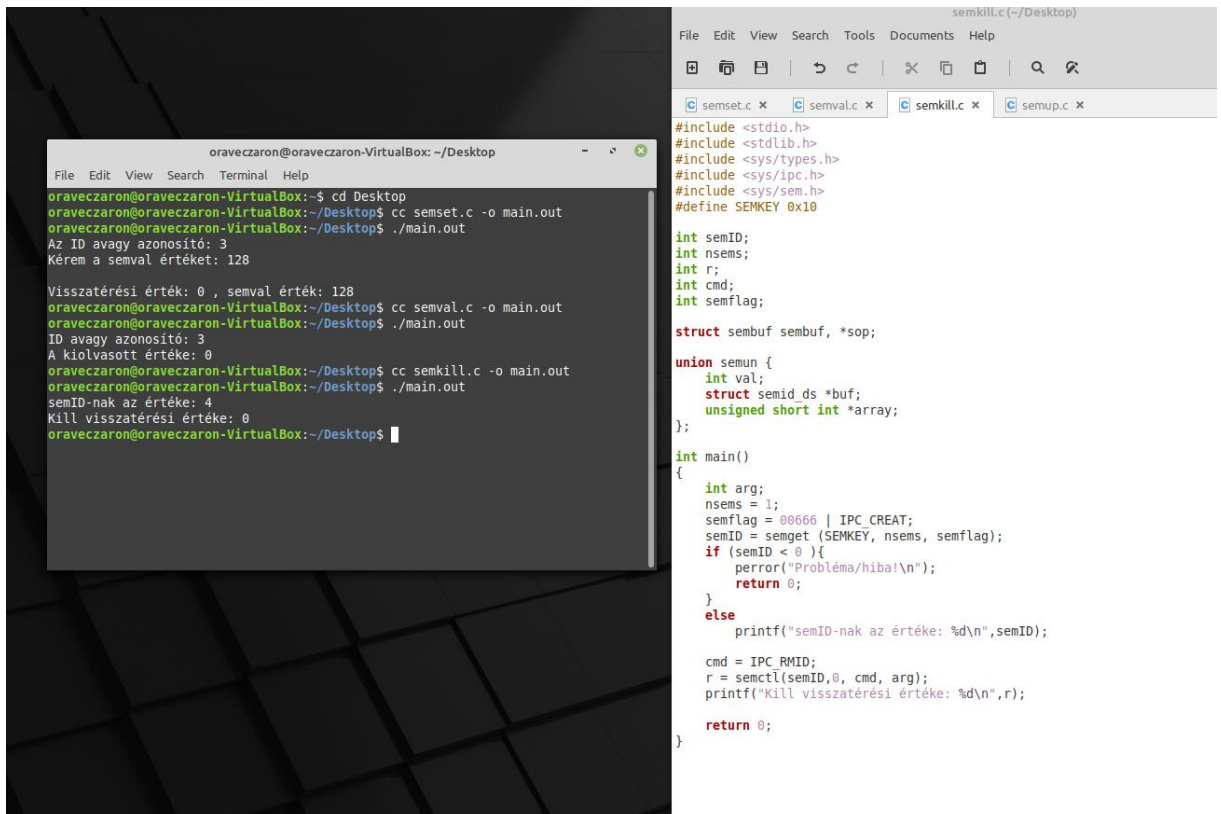
int semID;
int nsems;
int r;
int cmd;
int semflag;

struct sembuf sembuf, *sop;
union semun arg;

union semun {
    int val;
    struct semid_ds *buf;
    unsigned short int *array;
};

int main()
{
    semflag = 00666 | IPC_CREAT;
    nsems = 1;
    semID = semget (SEMKEY, nsems, semflag);
    if (semID < 0 ) {
        perror("Probléma/hiba!\n");
        return 0;
    }
    else
        printf("ID avagy azonosító: %d\n",semID);
    r = semctl(semID,0, cmd, NULL);
    cmd = GETVAL;
    printf("A kiolvasott értéke: %d\n",r);
    return 0;
}
```

semkill.c



The image shows a terminal window and a code editor side-by-side. The terminal window, titled 'oraveczaron@oraveczaron-VirtualBox: ~/Desktop', shows the execution of the 'semkill.c' program. The user enters 'cd Desktop', 'cc semset.c -o main.out', and './main.out'. The program prompts for an ID (3) and returns a value of 0. The user then compiles 'semkill.c' and runs it, which prompts for an ID (3) and returns a value of 0. The code editor, titled 'semkill.c (~/Desktop)', shows the source code of 'semkill.c'. It includes headers for stdio, stdlib, sys/types, sys/ipc, and sys/sem. It defines SEMKEY as 0x10. The code declares variables for semID, nsems, r, cmd, and semflag. It defines a struct sembuf and a union semun. The main function sets semflag to 00666 | IPC_CREAT, nsems to 1, and semID to semget(SEMKEY, nsems, semflag). It checks if semID is less than 0 and returns 0 if so. Otherwise, it prints the ID, sets r to semctl(semID, 0, cmd, arg), sets cmd to IPC_RMID, and prints the value of r.

```
oraveczaron@oraveczaron-VirtualBox: ~/Desktop
File Edit View Search Terminal Help
oraveczaron@oraveczaron-VirtualBox:~$ cd Desktop
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ cc semset.c -o main.out
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ ./main.out
Az ID avagy azonosító: 3
Kérem a semval értéket: 128

Visszatérési érték: 0 , semval érték: 128
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ cc semval.c -o main.out
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ ./main.out
ID avagy azonosító: 3
A kiolvasott értéke: 0
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ cc semkill.c -o main.out
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ ./main.out
semID-nak az értéke: 4
Kill visszatérési értéke: 0
oraveczaron@oraveczaron-VirtualBox:~/Desktop$
```

```
semkill.c (~/Desktop)
File Edit View Search Tools Documents Help
semset.c x semval.c x semkill.c x semup.c x
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#define SEMKEY 0x10

int semID;
int nsems;
int r;
int cmd;
int semflag;

struct sembuf sembuf, *sop;
union semun {
    int val;
    struct semid_ds *buf;
    unsigned short int *array;
};

int main()
{
    int arg;
    nsems = 1;
    semflag = 00666 | IPC_CREAT;
    semID = semget (SEMKEY, nsems, semflag);
    if (semID < 0 ){
        perror("Probléma/hiba!\n");
        return 0;
    }
    else
        printf("semID-nak az értéke: %d\n",semID);

    cmd = IPC_RMID;
    r = semctl(semID,0, cmd, arg);
    printf("Kill visszatérési értéke: %d\n",r);

    return 0;
}
```

semup.c

```
oraveczaron@oraveczaron-VirtualBox: ~/Desktop
File Edit View Search Terminal Help

oraveczaron@oraveczaron-VirtualBox:~$ cd Desktop
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ cc semset.c -o main.out
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ ./main.out
Az ID avagy azonosító: 3
Kérem a semval értéket: 128

Visszatérési érték: 0 , semval érték: 128
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ cc semval.c -o main.out
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ ./main.out
ID avagy azonosító: 3
A kiolvasott értéke: 0
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ cc semkill.c -o main.out
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ ./main.out
semID-nak az értéke: 4
Kill visszatérési értéke: 0
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ cc semup.c -o main.out
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ ./main.out
ID avagy azonosító: 5
Visszatérési érték: 0
oraveczaron@oraveczaron-VirtualBox:~/Desktop$
```

```
semup.c (~/Desktop)
File Edit View Search Tools Documents Help

semset.c x semval.c x semkill.c x semup.c x

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

int semID;
int nsems;
int r;
int semflag;
unsigned nsops;

struct sembuf sembuf, *sop;

int main()
{
    nsems = 1;
    semflag = 00666 | IPC_CREAT;
    semID = semget (SEMKEY, nsems, semflag);
    if (semID < 0){
        perror("Probléma/hiba!\n");
        return 0;
    }
    else
        printf("ID avagy azonosító: %d\n",semID);

    nsops = 1;
    sembuf.sem_flg = 0666;
    sembuf.sem_op = 1;
    sembuf.sem_num = 0;
    sop = &sembuf;
    r = semop(semID, sop, nsops);
    printf("Visszatérési érték: %d\n",r);

    return 0;
}
```

2a,

gyak11_2.c

```
oraveczaron@oraveczaron-VirtualBox:~$ cd Desktop
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ cc -pthread gyak11_2.c -o main.out
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ ./main.out
A pidem: 2400
A pidem: 2400
A pidem: 2400
oraveczaron@oraveczaron-VirtualBox:~/Desktop$
```

```
gyak11_2.c x
#include <stdio.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>

sem_t mutex;

void* thread(void* arg)
{
    sem_wait(&mutex);
    printf("A pidem: %d\n", getpid());
    sleep(3);
    sem_post(&mutex);
}

int main()
{
    sem_init(&mutex, 0, 2);
    pthread_t t1,t2,t3;
    pthread_create(&t1,NULL,thread,NULL);
    pthread_create(&t2,NULL,thread,NULL);
    pthread_create(&t3,NULL,thread,NULL);

    pthread_join(t1,NULL);
    pthread_join(t2,NULL);
    pthread_join(t3,NULL);
    sem_destroy(&mutex);
    return 0;
}
```