

1.Feladat:

. Adott egy rendszer (foglalási stratégiák), melyben a következő

☐ Szabad területek: 30k, 35k, 15k, 25k, 75k, 45k és

📌 Foglalási igények: 39k, 40k, 33k, 20k, 21k állnak rendelkezésre.

Határozza meg változó partíció esetén a következő algoritmusok felhasználásával: first fit, next fit, best fit, worst fit a foglalási igényeknek megfelelő helyfoglalást!

[illegible]

2.Feladat:

A feladat megoldásához először tanulmányozza Vadász Dénes:

Operációs rendszer jegyzet, a témához kapcsolódó fejezetét (6.4.), azaz

Írjon C nyelvű programokat, ahol

☐ kreál/azonosít szemafor készletet, benne N szemafor-t. A kezdő értéket 0-ra állítja – semset.c,

🔍 kérdezze le és írja ki a pillanatnyi szemafor értéket – semval.c

☐ szüntesse meg a példácskák szemafor készletét – semkill.c

☐ `sembuf.sem_op=1` értékkel inkrementálja a szemafor – `semup.c`

- semset.c:

```

1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/ipc.h>
4  #include <sys/sem.h>
5  //kulcs definíciója:
6  #define SEMKEY NX12YCH
7      int semid,      semaphor set azonosító
8          numofsems,  semaphorok száma a készletben
9          semnum,     semapho szám a set-en belül
10         rtn;        visszatérési érték
11
12     int semflg; // flag
13     struct sembuf sembuf, *sop;
14     union semunio{int val;
15     struct semid_ds *buf;
16     unsigned short *array;}arg;
17     int cmd;      // semctl-nek azonos parancs
18
19 int main()
20 {
21
22
23     nsems = 1;
24     semflg = 00666 | IPC_CREAT;
25     semid = semget (SEMKEY, numofsems, semflg);
26     if (semid < 0 ) {perror(" hibas a semafor-szám"); exit(0);}
27     else printf("\n semid: %d ",semid);
28     printf ("\n kerem a semval értéket ");
29
30     semnum = 0;
31
32     cmd = SETVAL;
33     scanf("%d",&arg.val);
34     rtn = semctl(semid,semnum, cmd, arg);
35
36
37     printf("\n set   rtn: %d ,semval: %d ",rtn,arg.val);
38     printf("\n");
39
40     return 0;
41 }

```

– semval.c:

```

1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/ipc.h>
4  #include <sys/sem.h>
5  //kulcs definíció:
6  #define SEMKEY NX12YCH
7
8      int semid,numofsems,rtn;
9      int semflg;
10     union semun{int val;
11     struct semid_ds *buf;
12     unsigned short *array;}arg;
13     union semun arg;
14     int cmd;
15
16     main()
17     {
18
19
20         numofsems = 1;
21         semflg = 00666 | IPC_CREAT;
22         semid = semget (SEMKEY, numofsems, semflg);
23         if (semid < 0 ) {perror(" hibas a semafororszam"); exit(0);}
24         else printf("\n semid: %d ",semid);
25         printf ("\n");
26
27         cmd = GETVAL;
28         rtn = semctl(semid,0, cmd, NULL);
29
30         printf("\n semval: %d ",rtn);
31         printf("\n");
32
33     }
34

```

– semkill.c:

```

1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/ipc.h>
4  #include <sys/sem.h>
5  //kulcs definíció:
6  #define SEMKEY NX12YCH
7
8
9  int semid,numofsems,rtn;
10 int semflg;
11 struct sembuf sembuf, *sop;
12 union semunio{int val;
13 struct semid_ds *buf;
14 unsigned short *array;}arg;
15 int cmd;
16
17 main()
18 {
19
20
21     numsems = 1;
22     semflg = 00666 | IPC_CREAT;
23     semid = semget (SEMKEY, numofsems, semflg);
24     if (semid < 0 ) {perror(" hibas a semafororszam"); exit(0);}
25     else printf("\n semid: %d ",semid);
26     printf ("\n");
27
28     //megszüntetes
29     cmd = IPC_RMID;
30     rtn = semctl(semid,0, cmd, arg);
31
32     printf("\n kill rtn: %d ",rtn);
33     printf("Meggzunt.\n");
34
35 }
36

```

– semup.c:

```

1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/ipc.h>
4  #include <sys/sem.h>
5  //tulcs definialasa:
6  #define SEMKEY NX12YCH
7
8      int semid,numofsems,rtn;
9      unsigned nsops; // semop-ban ezzel adjuk meg, hogy semaphore strukturalal foglakozzon
10
11      int semflg;
12      struct sembuf sembuf, *sop;
13
14
15  main()
16  {
17
18
19      nsems = 1;
20      semflg = 00666 | IPC_CREAT;
21      semid = semget (SEMKEY, numofsems, semflg);
22      if (semid < 0 ) {perror(" hibas a semaforozas"); exit(0);}
23      else printf("\n semid: %d ",semid);
24      printf ("\n");
25
26      nsops = 1;
27      sembuf.sem_num = 0;
28      sembuf.sem_op = 1;
29      sembuf.sem_flg = 0666; // Flag beallitas
30      sop = &sembuf;
31      rtn = semop(semid, sop, nsops);
32
33      printf("\n utp rtn: %d ",rtn);
34      printf("\n");
35
36  }
37

```