

Operációs rendszerek BSc

11. gyakorlat

2021. május 03. (hétfő) 16:00 – 18:00

Készítette:

Szeli Márk

Gazdaságinformatikus

B8VNQ7

1. 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!

First fit	30	35	15	25	75	45
39	30	35	15	25	39, 36	45
40	30	35	15	25	39, 36	40, 5
33	30	33, 2	15	25	39, 36	40, 5
20	20, 10	33, 2	15	25	39, 36	40, 5
21	20, 10	33, 2	15	21, 4	39, 36	40, 5

Next fit	30	35	15	25	75	45
39	30	35	15	25	39, 36	45
40	30	35	15	25	39, 36	40, 5
33	30	33, 2	15	25	39, 36	40, 5
20	30	33, 2	15	20, 5	39, 36	40, 5
21	30	33, 2	15	20, 5	39, 21, 15	40, 5

Best fit	30	35	15	25	75	45
39	30	35	15	25	75	39, 6
40	30	35	15	25	40, 35	39, 6
33	30	33, 2	15	25	40, 35	39, 6
20	30	33, 2	15	20, 5	40, 35	39, 6
21	21, 9	33, 2	15	20, 5	40, 35	39, 6

Worst fit	30	35	15	25	75	45
39	30	35	15	25	39, 36	45
40	30	35	15	25	39, 36	40, 5
33	30	35	15	25	39, 33, 3	40, 5
20	30	20, 15	15	25	39, 33, 3	40, 5
21	21, 9	20, 15	15	25	39, 33, 3	40, 5

2. Gyakorló 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 példacskák szemafor készletét – *semkill.c*, *sembuf.sem_op=1* értékkel inkrementálja a szemafort – *semup.c*.

B8VNQ7_semset.c:

```
1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/ipc.h>
4  #include <sys/sem.h>
5  #define SEMKEY 123456
6
7      int semid;
8      int nsems;
9      int semnum;
10     int rtn;
11     int semflg;
12     struct sembuf sembuf, *sop;
13     union semun arg;
14     int cmd;
15 main()
16 {
17     nsems = 1;
18     semflg = 00666 | IPC_CREAT;
19     semid = semget (SEMKEY, nsems, semflg);
20     if (semid < 0 )
21     {
22         perror("semget hiba!\n");
23         exit(0);
24     }
25     else
26     {
27         printf("emid: %d\n",semid);
28     }
29     printf ("Kerem a semval erteket!\n");
30     semnum = 0;
31     cmd = SETVAL;
32     scanf("%d",&arg.val);
33     rtn = semctl(semid, semnum, cmd, arg);
34     printf("set rtn: %d, semval: %d\n",rtn, arg.val);
35 }
```

B8VNQ7_semval.c:

```
1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/ipc.h>
4  #include <sys/sem.h>
5  #define SEMKEY 123456L
6
7      int semid,nsems,rtn;
8      int semflg;
9      struct sembuf sembuf, *sop;
10     union semun arg;
11     int cmd;
12
13     main()
14     {
15         nsems = 1;
16         semflg = 00666 | IPC_CREAT;
17         semid = semget (SEMKEY, nsems, semflg);
18         if (semid < 0 )
19         {
20             perror("Semget hiba!\n");
21             exit(0);
22         }
23         else
24         {
25             printf("semid: %d\n", semid);
26         }
27
28         cmd = GETVAL;
29         rtn = semctl(semid, 0, cmd, NULL);
30
31         printf("semval: %d\n", rtn);
32     }
```

B8VNQ7_semkill.c:

```
1  ✓ #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/ipc.h>
4  #include <sys/sem.h>
5  #define SEMKEY 123456L
6
7      int semid,nsems,rtn;
8      int semflg;
9      struct sembuf sembuf, *sop;
10     union semun arg;
11     int cmd;
12
13  ✓ main()
14  {
15      nsems = 1;
16      semflg = 00666 | IPC_CREAT;
17      semid = semget (SEMKEY, nsems, semflg);
18  ✓  if (semid < 0 )
19      {
20          perror("Semget hiba!\n");
21          exit(0);
22      }
23  ✓  else
24      {
25          printf("semid: %d\n",semid);
26      }
27
28      cmd = IPC_RMID;
29      rtn = semctl(semid, 0, cmd, arg);
30
31      printf("kill rtn: %d\n",rtn);
32  }
```

B8VNQ7_semup.c:

```
1  ✓ #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/ipc.h>
4  #include <sys/sem.h>
5  #define SEMKEY 123456L
6
7      int semid, nsems, rtn;
8      unsigned nsops;
9      int semflg;
10     struct sembuf sembuf, *sop;
11
12  ✓ main()
13  {
14      nsems = 1;
15      semflg = 00666 | IPC_CREAT;
16      semid = semget (SEMKEY, nsems, semflg);
17  ✓   if (semid < 0 )
18      {
19          perror("Semget hiba!\n");
20          exit(0);
21      }
22
23  ✓   else
24      {
25          printf("semid: %d\n", semid);
26      }
27
28      nsops = 1;
29      sembuf.sem_num = 0;
30      sembuf.sem_op = 1;
31      sembuf.sem_flg = 0666;
32      sop = &sembuf;
33      rtn = semop(semid, sop, nsops);
34      printf("up rtn: %d\n", rtn);
35  }
```