

# **Operációs rendszerek BSc**

**11. Gyak.**

**2022. 04. 25.**

**Készítette:**

Martinák Mátyás Bsc

Programtervező informatikus

KLNSPG

**Miskolc, 2022**

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.

A rendszerben a memória 4 kbyte-os blokkokban kerül nyilvántartásra, ennél kisebb méretű

töredék igény esetén a teljes blokk lefoglalásra kerül.

Határozza meg változó méretű partíció esetén a következő algoritmusok felhasználásával:

first fit, next fit, best fit, worst fit a foglalási igényeknek megfelelő helyfoglalást – táblázatos

Formában (az ea. Bemutatott mintafeladat alapján)!

Hasonlítsa össze, hogy a teljes szabad memóriaterület hány százaléka vész el átlagosan az

egy-egy algoritmusok esetén! A kapott eredményeket ábrázolja oszlop diagrammal!

Magyarázza a kapott eredményeket és hogyan lehet az eredményeket javítani!

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

2. feladat - 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 szemafor-t – semup.c

semset.c:

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#define SEMKEY 123456L

int semid, nsems, semnum, rtn;
int semflg;
struct sembuf sembuf, *sop;
union semun arg;
int cmd;

int main()
{
    nsems = 1;
    semflg = 00666 | IPC_CREAT;
    semid = semget (SEMKEY, nsems, semflg);
    if (semid < 0 ) {perror(" semget hiba"); exit(0);}
    else printf("\n semid: %d ",semid);
    printf ("\n kerem a semval erteket ");

    semnum = 0;

    cmd = SETVAL;
    scanf("%d",&arg.val);
    rtn = semctl(semid,semnum, cmd, arg);

    printf("\n set   rtn: %d ,semval: %d ",rtn,arg.val);
    printf("\n");
}
```

semval.c:

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#define SEMKEY 123456L

int semid,nsems,rtn;
int semflg;
struct sembuf sembuf, *sop;
union semun arg;
int cmd;

int main()
{

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

    cmd = GETVAL;
    rtn = semctl(semid,0, cmd, NULL);

    printf("\n semval: %d ",rtn);
    printf("\n");

}
```

semkill.c:

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#define SEMKEY 123456L

int semid,nsems,rtn;
int semflg;
struct sembuf sembuf, *sop;
union semun arg;
int cmd;

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

    cmd = IPC_RMID;
    rtn = semctl(semid,0, cmd, arg);

    printf("\n kill rtn: %d ",rtn);
    printf("\n");
}
```

semup.c:

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#define SEMKEY 123456L

int semid,nsems,rtn;
unsigned nsops;
int semflg;
struct sembuf sembuf, *sop;

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

    nsops = 1;
    sembuf.sem_num = 0;
    sembuf.sem_op = 1;
    sembuf.sem_flg = 0666;
    sop = &sembuf;
    rtn = semop(semid, sop, nsops);
    printf("\n up rtn: %d ",rtn);
    printf("\n");
}
```