

Operációs Rendszerek BSc

11.gyak.

2021.04.28

Készítette:

Orosz Dániel Bsc

Üzemmérnök-informatikus

C5S7FM

Miskolc, 2021

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	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	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	35	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, 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	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, 33, 3	45
20	30	20, 15	15	25	75	45
21	21, 9	35	15	25	75	45

2. 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**

A futtatás eredményét is tartalmazza a jegyzőkönyv.

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

    semnum = 0; //0-i semaphort azonositasa

    cmd = SETVAL; //semaphor ertekek allitasa
    scanf("%d",&arg.val);
    rtn = semctl(semid,semnum, cmd, arg); // a semid-vel azonosított set 0-ik semaphorat

    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 //semget kulcs

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; //semctl visszaadja a current semaphore erteket. Itt az rtn-be.
    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 //semget kulcs

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; //megszuntetes parancsa
    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 //semget kulcs

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

int main()
{
    nsems = 1; //semaphore
    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; //egy operaci
    sembuf.sem_num = 0; //0-ik semaphore
    sembuf.sem_op = 1; //inkrementaciot kereses
    sembuf.sem_flg = 0666; //flag beallitas
    sop = &sembuf; //semop argumentum kérés
    rtn = semop(semid, sop, nsops);
    printf("\n up rtn: %d ",rtn);
    printf("\n");
}
```