

Operációs rendszerek BSc

11. Gyak.

2022. 04. 27.

Készítette:

Bodnár Máté

László Bsc

Mérnökinformatikus

GOVLWD

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 egyes algoritmusok esetén! A kapott eredményeket ábrázolja oszlop diagrammal! Magyarázza a kapott eredményeket és hogyan lehet az eredményeket javítani!

First fit

Memória terület mérete	Lefoglalt méret	Megmaradt terület	Process	Process	Igény
30k	20k	10k	P4	P1	39k
35k	33k	2k	P3	P2	40k
15k	--	--	--	P3	33k
25k	21k	4k	P5	P4	20k
75k	39k	36k	P1	P5	21k
45k	40k	5k	P2		

Next fit

NEXT FIT				Process	Igény
Memória terület mérete	Lefoglalt méret	Megmaradt terület	Process	P1	39k
30k	21k	9k	P5	P2	40k
35k	33k	2k	P3	P3	33k
15k	--	--	--	P4	20k
25k	20k	5k	P4	P5	21k
75k	39k	36k	P1		
45k	40k	5k	P2		

Best fit

BEST FIT				Process	Igény
Memória terület mérete	Lefoglalt méret	Megmaradt terület	Process	P1	39k
30k	21k	9k	P5	P2	40k
35k	33k	2k	P3	P3	33k
15k	--	--	--	P4	20k
25k	20k	5k	P4	P5	21k
75k	40k	35k	P2		
45k	39k	6k	P1		

Worst fit



OS [Fut] - Oracle VM VirtualBox

Fájl Gép Nézet Bevitel Eszközök Súlyó

File Edit View Search Terminal Help

```
mint@mint:~$ gcc semset.c -o semset
mint@mint:~$ ./semset
A Semid: 11
1. - set 0, semval: 0
2. - set 0, semval: 0
3. - set 0, semval: 0
mint@mint:~$
```



OS [Fut] - Oracle VM VirtualBox

Fájl Gép Nézet Bevitel Eszközök Súlyó

mint@mint:~

File Edit View Search Terminal Help

GNU nano 4.8

semval.c

```
#include <stdlib.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>

#define NUM_OF_SEMS 3

int main(void) {
    key_t key = ftok("semset.c", (int)'S');
    int flag = 00666 | IPC_CREAT;
    int num_of_semaphs = NUM_OF_SEMS;
    int sem_id = semget(key, num_of_semaphs, flag);

    if (sem_id < 0) {
        perror("SEMGET HIBA...");
        return -1;
    } else {
        printf("A Semid: %d\n", sem_id);
    }
    int command = GETVAL;

    for (int i = 0; i < num_of_semaphs; i++) {
        int current_sem_value = semctl(sem_id, i, command, NULL);
        printf("semval %d: %d\n", i, current_sem_value);
    }

    return 0;
}
```

Get Help
Exit

Write Out
Read File

Where Is
Replace

Cut Text
Paste Text

Justify
To Spell

Cur Pos
Go To Line

Undo
Redo

Mark Text
Copy Text

To Bracket
Where Was

Previous
Next

Back
Forward

Find Word
Next Word

Help
End

Aktiválja a Windows rendszert a

OS [Fut] - Oracle VM VirtualBox

Fájl Gép Nézet Bevitel Eszközök Sûgó

```
File Edit View Search Terminal Help
mint@mint:~$ gcc semset.c -o semset
mint@mint:~$ ./semset
A Semid: 11
1. - set 0, semval: 0
2. - set 0, semval: 0
3. - set 0, semval: 0
mint@mint:~$ nano semset.c
mint@mint:~$ nano semval.c
mint@mint:~$ gcc semval.c -o semval
mint@mint:~$ ./semval
A Semid: 11
semval 0: 0
semval 1: 0
semval 2: 0
mint@mint:~$
```

```
OS [Fut] - Oracle VM VirtualBox
Fájl Gép Nézet Bevitel Eszközök Sûgó
mint@mint: ~
File Edit View Search Terminal Help
GNU nano 4.8 semkill.c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#define NUM_OF_SEMS 3
union semun {
    sem_val;
    struct semid_ds *buf;
    unsigned short *array;
    struct semaphinfo * _buf;
};
int main(int id) {
    key_t key = ftok("semset.c", (id)'S');
    int flag = 0666 | IPC_CREAT;
    int num_of_semaphs = NUM_OF_SEMS;
    int sem_id = semget(key, num_of_semaphs, flag);
    if (sem_id < 0) {
        perror("SEMGET HIBA....");
        return -1;
    } else {
        printf("A Semid: %d\n", sem_id);
    }
    int command = IPC_RMID;
    union semun argument;
    int status = semctl(sem_id, 0, command, argument);
    printf("Semaphore kill...\n");
    return 0;
}
```

Read 37 Lines (Converted from DOS format)

Get Help Exit Write Out Read File Where Is Replace Cut Text Paste Text Justify To Spell Cur Pos Go To Line Undo Redo Mark Text Copy Text To Bracket Where Was Previous Next Back Forward Aktiválja a Windows rendszert a

OS [Fut] - Oracle VM VirtualBox

Fájl Gép Nézet Bevitel Eszközök Súgó

mint@mint: ~

```
File Edit View Search Terminal Help
mint@mint:~$ gcc semset.c -o semset
mint@mint:~$ ./semset
A Semid: 11
1. - set 0, semval: 0
2. - set 0, semval: 0
3. - set 0, semval: 0
mint@mint:~$ nano semset.c
mint@mint:~$ nano semval.c
mint@mint:~$ gcc semval.c -o semval
mint@mint:~$ ./semval
A Semid: 11
semval 0: 0
semval 1: 0
semval 2: 0
mint@mint:~$ nano semkill.c
mint@mint:~$ gcc semkill.c -o semkill
mint@mint:~$ ./semkill
A Semid: 11
Semaphore Kill...
mint@mint:~$
```

Aktiválja a Windowst
Aktiválja a Windows rendszert a Gépházában

OS [Fut] - Oracle VM VirtualBox

Fájl Gép Nézet Bevitel Eszközök Súgó

mint@mint: ~

```
File Edit View Search Terminal Help
GNU nano 4.8 semup.c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>

#define NUM_OF_SEMS 3

int main(void) {
    key_t key = ftok("semset.c", (int)'S');
    int flag = 0666 | IPC_CREAT;
    int num_of_semaphs = NUM_OF_SEMS;
    int sem_id = semget(key, num_of_semaphs, flag);

    if (sem_id < 0) {
        perror("SEMGET HIBA ...");
        return -1;
    } else {
        printf("A Semid: %d\n", sem_id);
    }

    struct sembuf sem_op_buf;
    sem_op_buf.sem_op = 1;
    sem_op_buf.sem_flg = 0666;
    printf("Semaph. number: ");
    scanf("%hu", &sem_op_buf.sem_num);

    int rtn = semop(sem_id, &sem_op_buf, 1);
    printf("up rtn: %d\n", rtn);

    return 0;
}
```

Get Help Exit Write Out Read File Where Is Replace Cut Text Paste Text Justify To Spell Cur Pos Go To Line Undo Redo Mark Text Copy Text To Bracket Where Was Previous Next Back Forward Aktiválja a Windowst

```
mint@mint:~$ gcc semset.c -o semset
mint@mint:~$ ./semset
A Semid: 11
1. - set 0, semval: 0
2. - set 0, semval: 0
3. - set 0, semval: 0
mint@mint:~$ nano semset.c
mint@mint:~$ nano semval.c
mint@mint:~$ gcc semval.c -o semval
mint@mint:~$ ./semval
A Semid: 11
semval 0: 0
semval 1: 0
semval 2: 0
mint@mint:~$ nano semkill.c
mint@mint:~$ gcc semkill.c -o semkill
mint@mint:~$ ./semkill
A Semid: 11
Semaphore kill...
mint@mint:~$ nano semkill.c
mint@mint:~$ nano semup.c
mint@mint:~$ gcc semup.c -o semup
mint@mint:~$ ./semup
A Semid: 12
Semaph. number: 2
up rtn: 0
mint@mint:~$
```

