

Operációs rendszerek BSc

11. Gyak.

2022. 04. 07.

Készítette:

Varga Bence

Mérnökinformatika

CKFEC9

Miskolc, 2022

1. feladat –

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

Foglalási igények: 39k, 40k, 33k, 20k, 21k

first fit						
	Memória terület - szabad terület					
Foglalási igény	30	35	15	25	75	45
39					36 (75 - 39)	
40						5 (45 - 40)
33		2 (35 - 33)				
20				5 (25 - 20)		
21	9 (30 - 31)					
next fit						
	Memória terület - szabad terület					
Foglalási igény	30	35	15	25	75	45
39					36 (75 - 39)	
40						5 (45 - 40)
33		2 (35 - 33)				
20				5 (25 - 20)		
21					15 (36 - 21)	
best fit						
	Memória terület - szabad terület					
Foglalási igény	30	35	15	25	75	45
39						6 (45 - 39)
40					35 (75 - 40)	
33		2 (35 - 33)				
20				5 (25 - 20)		
21	9 (30 - 31)					
worst fit						
	Memória terület - szabad terület					
Foglalási igény	30	35	15	25	75	45
39					36 (75 - 39)	
40						5 (45 - 40)
33					3 (36 - 33)	
20		15 (35 - 20)				
21	9 (30 - 31)					

2. feladat –

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

#define KEY 77777L

void up(int);
void down(int);

void main()
{
    int semID = semget(KEY, 0, 0);

    if (semID == -1)
    {
        perror("Nem sikerult megnyitni\n");
        exit(-1);
    }

    //belepesi szakasz
    printf("Kritikus szakasz\n");
    down(semID);
    sleep(3);
    printf("pid : %d\n", getpid());
    printf("%d \n", semctl(semID, 0, GETVAL));
    up(semID);
    printf("kritikus szakasz vege\n");
}
```

```
void up(int semId) {
    struct sembuf buffer;
    buffer.sem_num = 0;
    buffer.sem_op = 1;
    buffer.sem_flg = 0;

    semop(semId, &buffer, 1);
}

void down(int semId) {
    struct sembuf buffer;
    buffer.sem_num = 0;
    buffer.sem_op = -1;
    buffer.sem_flg = 0;

    semop(semId, &buffer, 1);
}
```

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

#define KEY 77777L

void main() {
    int semID = semget(KEY, 0, 0);

    if (semID == -1)
    {
        perror("Nem sikerult megnyitni\n");
        exit(-1);
    }

    if (semctl(semID, 0, IPC_RMID) == -1)
    {
        perror("Nem sikerult torolni\n");
        exit(-1);
    }

    printf("Torolve\n");
}
```