

# JEGYZŐKÖNYV

Operációs rendszerek BSC  
2022. tavasz féléves feladat

Készítette: **Bodnár Máté László**

Neptunkód: **GOVLWD**

### Feladat Leírása:

23. Adott az alábbi terhelés esetén a rendszer. Határozza meg az indulás, befejezés, várakozás/átlagos várakozás és körülfordulás/átlagos körülfordulás, válasz/átlagos válaszidő és a CPU kihasználtság értékeit az FCFS ütemezési algoritmusok mellett! (cs: 0,1ms; sch: 0,1ms)

	P1	P2	P3	P4	P5
Érkezés	0	1	3	9	12
CPU idő	3	5	2	5	5
Indulás					
Befejezés					
Várakozás					

Ábrázolja Gantt diagram segítségével az aktív/várakozó processzek futásának menetét.

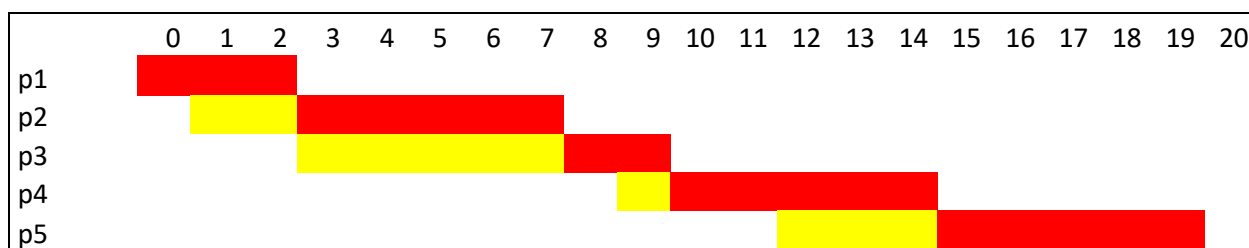
A feladat elkészítésének a lépései:

1. Kiszámítjuk az indulás, befejezés, várakozás, válaszidőt CPU kihasználtságot és a körülfordulást (ezek kiszámítása az excel táblázatban található)
2. A Körülfordulási, Válaszidők és a Várakozások átlagát kiszámoljuk.
3. Gantt diagramm elkészítése

Az eredmény:

FCFS	P1	P2	P3	P4	P5
Érkezés	0	1	3	9	12
CPU idő	3	5	2	5	5
Indulás	0	3	8	10	15
Befejezés	3	8	10	15	20
Várakozás	0	2	5	1	3
Körülford. Idő	3	7	7	6	8
Válaszidő	0	2	5	1	3

CPU usage	97,56%
Körülfordulási idő átl.	6,2 ms
Várakozások átlaga	2,2 ms
Válaszidők átlaga	2,2 ms



### IPC-s feladat leírása:

13. Írjon C nyelvű programokat, ami létrehoz egy osztott memória szegmenst az egyik program ír bele és vár pár másodpercet bináris szemafor segítségével "védi" az írást a másik program pedig kiolvassa belőle

A feladat elkészítésének lépései:

1. program

1. lépés: megnézzük, hogy van-e már SHMKEY kulcsú és „size” méretű szegmens, ha nincs el kell készíteni ezt egy if else –vel oldottam meg.

2. lépés: Megnézzük hol fut a kritikus és a nem kritikus szakasz.

3. lépés: kiíratunk valamilyen szöveget

4. lépés: Megszüntetjük a szemafort.

2. program

1. lépés: rákapcsolódunk

2. lépés: miután rákapcsolódtunk kiíratjuk a régi szöveget, majd töröljük a szegmenst.

# A Futtatás eredménye:

OS [Fut] - Oracle VM VirtualBox

FájlGépNézetBevitelEszközökSúgó

mint@mint: ~/beadando

FileEditViewSearchTerminalHelp

GNU nano 4.813.c

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

#define SHMKEY 69L
#define KEY 15L
#define PERM 00666

int main()
{
    int shmid;          /* osztott mem azonosító */
    key_t key;          /* kulcs a shm-hez */
    int size=512;       /* Az osztott szegmensnek merete byte-ban megadva */
    int shmflg;         /* flag a jellezőkhöz */

    struct vmi {
        int hossz;
        char szoveg[512-128*4]; // Az egész merete 512, ezért a hossz változatot levonjuk a tombból
    } *segm;
    key = SHMKEY;

    /* Megnézzük, van-e már SHMKEY kulcsú és "size" meretű szegmens. */
    shmflg = 0;
    if ((shmflg=shmget(key, size, shmflg)) < 0) {
        printf("Nem létezik meg a szegmens! Kérem készítsuk el!\n");
        shmflg = 00666 | IPC_CREAT;

        if ((shmflg=shmget(key, size, shmflg)) < 0) {
            perror("Az shmget() system-call sikertelen!\n");
            exit(-1);
        }
        else printf("Van már ilyen shm szegmens!\n");

        printf("\n Az shm szegmens azonosítója %d: \n", shmid);
        shmflg = 00666 | SHM_RND; // Az SHM_RND-t az igazításhoz használom
        segm = (struct vmi *)shmat(shmid, NULL, shmflg); // OS -re bízom, hogy milyen címtartományt használjon (NULL)
        if (segm == (struct vmi *)-1) {
            perror("Sikertelen attach!\n");
            exit(-1);
        }

        int id; // id a szemafor azonosítója lesz
        if ((id = semget(KEY, 1, 0)) < 0) { // ekkor meg nem létezik
            if ((id = semget(KEY, 1, PERM | IPC_CREAT)) < 0) {
                perror("A szemafor nem nyitható meg.");
                exit(-1);
            }
        }
        else {
            perror("Már létezik ez a szemafor.");
            exit(0);
        }

        if (semctl(id, 0, SETVAL, 1) < 0) {
            perror("Nem sikerült inicializálni.");
        }
        else {
            puts("A szemafor kész és inicializált is.");
        }

        struct sembuf up[1] = {0, 1, SEM_UNDO};
        struct sembuf down[1] = {0, -1, SEM_UNDO};
        puts("Itt fut a nem kritikus szakasz...");

        semop(id, down, 1); /* Ez lesz a belepési szakasz */
        puts("Itt fut a kritikus szakasz...");
        strcpy(segm->szoveg, "Bodnar Mate laszlo Neptunkod:GOVLM0");
        sleep(3);
        semop(id, up, 1); /* Itt pedig a kilepési szakasz */
        puts("Itt megint nem kritikus szakasz fut...");
        if (semctl(id, 0, IPC_RMID, 0) < 0) {
            perror("Nem sikerült törölni.");
            exit(-1);
        }

        puts("A szemaforot megszüntettük. \n");
        shmdt(segm);
        return 0;
    }
}
```

Get HelpExitWrite OutRead FileWhere IsReplaceCut TextPaste TextJustifyTo SpellCur PosGo To LineUndoRedoMark TextCopy TextTo BracketWhere WasPreviousNextBackForwardAktiválja a Windows rendszert

OS [Fut] - Oracle VM VirtualBox

FájlGépNézetBevitelEszközökSúgó

mint@mint: ~/beadando

FileEditViewSearchTerminalHelp

GNU nano 4.813.c

Get HelpExitWrite OutRead FileWhere IsReplaceCut TextPaste TextJustifyTo SpellCur PosGo To LineUndoRedoMark TextCopy TextTo BracketWhere WasPreviousNextBackForwardAktiválja a Windows rendszert

```
OS [Fut] - Oracle VM VirtualBox
Fájl Gép Nézet Bevitel Eszközök Súlyó

File Edit View Search Terminal Help
GNU nano 4.8 13-1.c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <string.h>

#define SHMKEY 69L

int main()
{
    int shmid; /* osztott mem azonosito */
    key_t key; /* kulcs a shmem-hez */
    int size=512; /* osztott szegmens meret byte-ban */
    int shmflg; /* flag a jellemzokhoz */
    char *vni; /* szegmens */
    int hossz;
    char szoveg[512-1]; /* az egesz merete 512, ezert a hossz változot levonjuk a tombol */
    } *segm; /* Ezt a strukturat kepezzuk a szegmensre */

    key = SHMKEY;

    shmflg = 0; /* Nincs IPC_CREAT, feltetelozzuk, az shmcreate.c
    keszített osztott memoria szegmenst */

    if ((shmid=shmget(key, size, shmflg)) < 0) {
        perror("Az shmget system-call sikertelen!\n");
        exit(-1);
    }

    /* Attach - rakcsolodunk */

    shmflg = 0666 | SHM_RWP; /* RWP az igratashoz */
    segm = (char *)shmatt(shmid, NULL, shmflg);
    if (segm == NULL) /* Itt a NULL azt jelenti, hogy az OS-re bizon, milyen cintartomanyt hasznaljon. */
    {
        perror("Sikertelen attach!\n");
        exit(-1);
    }

    // Sikeres attach utan a "segm" cimre ott az osztott memoria.

    if (strlen(segm->szoveg) > 0)
        printf("A Regi szoveg: %s (%d hosszon)", segm->szoveg, segm->hossz);
    int rtn = shmctl(shmid, IPC_RMID, 0);
    printf("Szegmens torolve. Hibakod: %d\n", rtn);
    return 0;
}

Aktiválja a Windows rendszert a Gépház
```

## OS [Fut] - Oracle VM VirtualBox

Fájl Gép Nézet Bevitel Eszközök Súlyó

```
File Edit View Search Terminal Help
mint@mint:~/beadando$ gcc 13.c -o proba
mint@mint:~/beadando$ ./proba
Nem letezik meg a szegmens! Kerem keszitsuk el

Az shm szegmens azonositoja 98325:
A szemafor kesz es inicializalt is.
Itt fut a nem kritikus szakasz...
Itt fut a kritikus szakasz...
Itt megint nem kritikus szakasz fut...
A szemafort megszuntettuk.

mint@mint:~/beadando$ gcc 13-1.c -o proba2
mint@mint:~/beadando$ ./proba2

Regi szoveg: Bodnar Mate laszlo Neptunkod:GOVLWD (0 hosszon)Szegmens torolve. Hibakod: 0
mint@mint:~/beadando$
```