JEGYZŐKÖNYV

Operációs rendszerek BSc

2022. tavasz féléves feladat

Készítette: **Sziráczki Soma**

Neptunkód: **BK6QE8**

A feladat leírása:

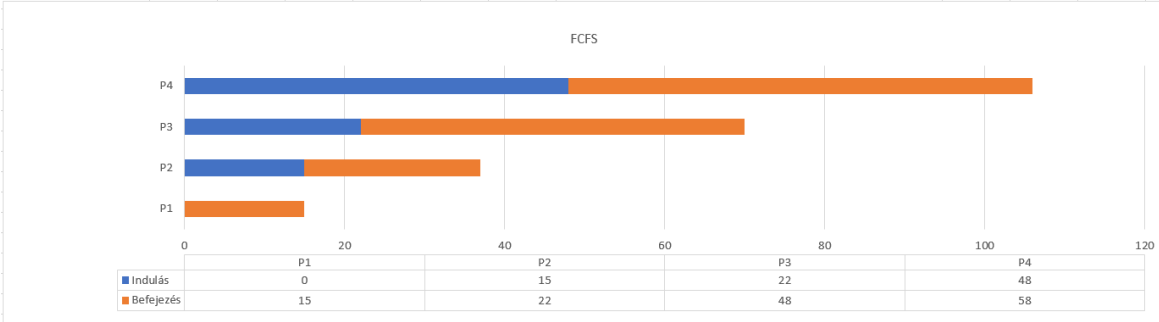
2. Adott az alábbi terhelés esetén a rendszer. Határozza meg az *indulás, befejezés, várakozás/átlagos várakozás és körülfordulás/átlagos körülfordulás, válasz/átlagos válaszidő* és a *CPU kihasználtság* értékeket az FCFS ütemezési algoritmusok mellett! (cs: 0,1ms; sch: 0,1ms)

	P1	P2	P3	P4
Érkezés	0	8	12	20
CPU idő	15	7	26	10
Indulás				
Befejezés				
Várakozás				

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

A futtatás eredménye:

FCFS	P1	P2	P3	P4			CPU kih.%=((58+0.4)-0.4)/58.4=0.9931--->99.31%
Érkezés	0	8	12	20			
CPU idő	15	7	26	10			Átlag vár=(sum)vár/(sum)p=45/4=11.25
Indulás	0	15	22	48			
Befejezés	15	22	48	**58**			Átlag kör.=(sum)kör./(sum)p=101/4=25.25
Várakozás	0	7	10	28			
Körülfordulási idők	15	14	36	38			Átlag vál.=(sum)vál/(sum)p=45/4=11.25
Válasz	0	7	10	28			



	P1	P2	P3	P4
Indulás	0	15	22	48
Befejezés	15	22	48	58

A feladat leírása:

Írjon egy C programot, amely
létrehoz egy osztott memória szegmenst és majd rácsatlakozik.Továbbá egy másik program olvasson be 3 számot egy file-ból(amik a háromszög oldalainak hosszát jelentik) az osztott memóriába és döntse el,hogy szerkeszthető-e belőlük háromszög.
A döntési eredmény a file kimeneten, ha készíthető háromszög van kerülete illetve területe, ha nincs akkor ezekre -1 et ad vissza.
Az adatokat és az eredményt egy fájl kimeneten adjuk vissza.

A futtatás eredménye:

```
sziraczki@sziraczki-VirtualBox:~$ cd Asztal
sziraczki@sziraczki-VirtualBox:~/Asztal$ cd beadando
sziraczki@sziraczki-VirtualBox:~/Asztal/beadando$ cd program
sziraczki@sziraczki-VirtualBox:~/Asztal/beadando/program$ chmod +x elso masodik
sziraczki@sziraczki-VirtualBox:~/Asztal/beadando/program$ ./elso
Van mar ilyen shm szegmens!

Az shm szegmens azonositoja 360501:

Add meg a parancs szamat:
0. IPC_STAT (status)
1. IPC_RMID (torles)
> 0

Az shm szegmens azonositoja 360501:
Szegmens merete: 512
Utolso shmop()-os processz pid-je: 13627
sziraczki@sziraczki-VirtualBox:~/Asztal/beadando/program$ ./masodik
Szerkeztheto 3szog
sziraczki@sziraczki-VirtualBox:~/Asztal/beadando/program$
```

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

#define SHMKEY 60001L

int main()
{
    int shmid;
    key_t key;
    int size=512;
    int shmflg;
    int rtn;
    key = SHMKEY;
    struct shmid_ds shmid_ds, *buf;
    buf = &shmid_ds;
    struct vmi {
        int oldal1, oldal2, oldal3;
        } *segm;
    shmflg = 0;
    if ((shmid=shmget(key, size, shmflg)) < 0) {
       printf("Nincs meg szegmens! Keszitsuk el!\n");
       shmflg = 00666 | IPC_CREAT;
       if ((shmid=shmget(key, size, shmflg)) < 0) {
          perror("Az shmget() system-call sikertelen!\n");
          exit(-1);
       }
    } else printf("Van mar ilyen shm szegmens!\n");

    printf("\nAz shm szegmens azonositoja %d: \n", shmid);

    shmflg = 00666 | SHM_RND;
    segm = (struct vmi *)shmat(shmid, NULL, shmflg);
    if (segm == (void *)-1) {
        perror("Sikertelen attach!\n");
        exit(-1);
    }
    int cmd;
    do
    {
```

```
    int cmd;
    do
    {
        printf("\nAdd meg a parancs szamat:\n");
        printf("0. IPC_STAT (status)\n");
        printf("1. IPC_RMID (torles)\n> ");
        scanf("%d",&cmd);
    } while (cmd < 0 || cmd > 1);

    switch (cmd)
    {
        case 0:
        rtn = shmctl(shmid, IPC_STAT, buf);
        printf("\nAz shm szegmens azonositoja %d: \n", shmid);
        printf("Szegmens merete: %d\n",buf->shm_segsz);
        printf("Utolso shmop()-os processz pid-je: %d\n",buf->shm_lpid);
        break;

        case 1:
        shmdt(segm);
        rtn = shmctl(shmid, IPC_RMID, NULL);
        printf("Szegmens torolve. Hibakod: %d\n", rtn);
    }

    exit(0);
}
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <math.h>

#define SHMKEY 60001L

int main()
{
    int shmid;
    key_t key;
    int size=512;
    int shmflg;
    FILE *in_file;
    FILE *out_file;
    struct vmi {
        int oldal1, oldal2, oldal3;
    } *segm;

    key = SHMKEY;

    shmflg = 0;

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

    shmflg = 00666 | SHM_RND;
    segm = (struct vmi *)shmat(shmid, NULL, shmflg);
    if (segm == (void *)-1) {
        perror("Sikertelen attach!\n");
        exit(-1);
    }

    in_file = fopen("oldalak", "r");
    out_file = fopen("haromszog","w");
    int kerulet = 0;
    double terulet = 0;
```

```c
  if (in_file == NULL)
  {
      printf("Can't open file for reading.\n");
  }

  else
  {
      fscanf(in_file, "%d", &segm->oldal1);
      fscanf(in_file, "%d", &segm->oldal2);
      fscanf(in_file, "%d", &segm->oldal3);

  if(segm->oldal1 < (segm->oldal2+segm->oldal3) && segm->oldal2 < (segm->oldal1+segm->oldal3)  && segm->oldal3 < (segm->oldal1+segm->oldal2) )
  {
      printf("Szerkeztheto 3szog.\n");
      kerulet = segm->oldal1+ segm->oldal2 + segm->oldal3;
      int s = kerulet / 2;
      int teruletResz = s*(s-segm->oldal1)*(s-segm->oldal2)*(s-segm->oldal3);
      terulet = sqrt(teruletResz);

      if (out_file == NULL)
      {
          printf("Can't open file for writing.\n");
      }else
      {
          fprintf(out_file,"A kapott adatokbol: %d %d %d lehet haromszoget alkotni! \n Kerulete: %d\n Terulete: %d\n",segm->oldal1, segm->oldal2, segm->oldal3,kerulet,terulet);
          fclose(out_file);
      }
  }

  else{
      if (out_file == NULL)
      {
          printf("Can't open file for writing.\n");
      }else
      {
          fprintf(out_file,"-1");
          fclose(out_file);
          }
      }
      fclose(in_file);
  }
```