

Operációs rendszerek BSc

9. Gyak.

2022. 04. 04.

Készítette:

Nyíri Levente
Mérnökinformatikus
F023QC

Miskolc, 2022

1. feladat

```
int main()
{
    int fileHandle = open(FILE, O_RDWR);
    if(fileHandle == -1)
    {
        perror("Nem sikerult megnyitni a fajt!");
        return 1;
    }
    else
    {
        printf("Sikeres volt a fajt megnyitasa!\n");
    }
    char tartalom[120];

    int olvasott = read(fileHandle, tartalom, sizeof(tartalom));
    printf("Beolvasott tartalom: \"%s\" osszesen: \"%i\" byte.\n", tartalom, olvasott);

    lseek(fileHandle, 0, SEEK_SET);

    char text[] = "teszt";
    int irt = write(fileHandle, text, sizeof(text));
    printf("A fajtba irtuk a(z) \"%s\" szoveget. Osszesen: \"%i\" byte.\n", text, irt);

    close(fileHandle);

    return 0;
}
```

2. feladat

```
#include <stdio.h>
#include <unistd.h>
#include <signal.h>

void handleSignals(int signum);

int main()
{
    void (*sigHandlerInterrupt)(int);
    void (*sigHandlerQuit)(int);
    void (*sigHandlerReturn)(int);
    sigHandlerInterrupt = sigHandlerQuit = handleSignals;
    sigHandlerReturn = signal(SIGINT, sigHandlerInterrupt);

    if(sigHandlerReturn == SIG_ERR)
    {
        perror("Signal error");
        return 1;
    }

    sigHandlerReturn = signal(SIGQUIT, sigHandlerQuit);

    if(sigHandlerReturn == SIG_ERR)
    {
        perror("Signal error");
        return 1;
    }

    for(;;)
    {
        printf("A program leallitasahoz a kovetkezo ket vegesse el: \n");
        printf("1. Nyisson meg egy masik terminalt.\n");
        printf("2. Adj a parancsot: kill: wd \n", getpid());
        sleep(10);
    }

    return 0;
}

void handleSignals(int signum)
{
    switch(signum)
    {
        case SIGINT:
            printf("\n CTRL+C-t eszelt\n");
            signal(SIGINT, SIG_DFL);
            break;
        case SIGQUIT:
            printf("SIGQUIT aktivlodott\n");
            break;
        default:
            printf("\n Fogadott jel szama: %d\n", signum);
            break;
    }

    return ;
}
```

```
A program leállításához a következőket végezze el:
1. Nyisson meg egy másik terminált.
2. Adja ki a parancsot: kill: 1128
```

3. feladat

	P1	P2	P3	P4
Érkezés	0	0	2	5
CPU idő	24	3	6	3
Indulás	0	24	27	33
Befejezés	24	27	33	36
Várakozás	0	24	25	28
Körülfordulási idő	24	27	31	31

Algoritmus neve:	FCFS
CPU kihasználtság	156
Körülfordulási idő átlaga	28,25
Várakozási idő átlaga	19,25
Válaszidő átlaga	91

	P1	P2	P3	P4
Erkezés	0	0	2	5
CPU idő	24	3	6	3
Indulás	0	24	30	27
Befejezés	24	27	36	30
Várakozás	0	24	28	22
Körülfordulási idő	24	27	34	25
Algoritmus neve: SJF			153	
CPU kihasználtság			27,5	
Körülfordulási idő átlaga			18,5	
Várakozási idő átlaga			88	
Válaszidő átlaga				

4 ms	P1	P2	P3	P4
Érkezés	0, 4, 15	0	2, 11	5
CPU idő	24, 20, 16	3	6, 2	3
Indulás	0, 11, 20	4	7, 18	15
Befejezés	4, 15, 36	7	11, 20	18
Várakozás	0, 7, 5	4	5, 7	10
	4, 11, 21	7	9, 9	13

Algoritmus neve: RR

CPU kihasználtság	156,4
Körülfordulási idők átlaga	10
Várakozási idők átlaga	7
Válaszidők átlaga	95,6

```

graph LR
    subgraph Timeline [Time]
        direction LR
        T[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36]
    end

    P1[P1] --- P1_1[0-4]
    P1[P1] --- P1_2[11-20]
    
    P2[P2] --- P2_1[0-3]
    
    P3[P3] --- P3_1[0-4]
    P3[P3] --- P3_2[11-15]
    P3[P3] --- P3_3[20-24]
    
    P4[P4] --- P4_1[0-5]
    P4[P4] --- P4_2[7-9]
  
```

Gyakorló feladat:

```

void handleSigalarm();

int main()
{
    printf("A program pidje: %d\n",getpid());
    signal(SIGALRM, handleSigalarm);
    pause();
    printf("Kibillent\n");
    exit(0);
    return 0;
}

void handleSigalarm()
{
    printf("F023QC\n");
}

```

```

1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <signal.h>
4  #include <unistd.h>
5
6  void kezelo(int i)
7  {
8      printf("Signal kezelese: %d\n",i);
9      return;
10 }
11
12
13 int main()
14 {
15     printf("PID: %d\n",getpid());
16     printf("Signal kezelo atvetele: %d\n",signal(SIGTERM,&kezelo));
17     while(1)
18     {
19         printf("lepes\n");
20         sleep(3);
21     }
22 }
23

```