

# **Operációs rendszerek BSc**

**5. Gyak.**

2022. 03. 07.

**Készítette:**

Nyíri Levente Bsc

Mérnökinformatikus

F023QC

**Miskolc, 2022**

**1. feladat**

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/types.h>
4 #include <sys/wait.h>
5
6 int main()
7 {
8     int status;
9
10    if((status = system("date")) < 0) //éppen aktuális nap dátum
11        perror("system() error"); //hiba esetén lefut a perror() függvény
12
13    if(WIFEXITED(status))
14        printf("Normalis befejezodes, visszaadott ertekek = %d\n", WIFEXITED(status));
15
16 }
17

```

## 2. feladat

```

#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>

int main()
{
    char input[100];
    printf("Adjon meg egy parancsot: ");
    scanf("%s", input);
    system(input);

    return 0;
}

```

```

Adjon meg egy parancsot: uname
Linux

Process returned 0 (0x0)   execution time : 6.141 s
Press ENTER to continue.

```

## 3. feladat

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/wait.h>
4 #include <sys/types.h>
5
6 int main()
7 {
8     pid_t pid;
9
10    if((pid = fork()) < 0)
11    {
12        perror("fork error");
13    }
14    else if(pid == 0)
15    {
16        if(execl("./child", "child", (char *) NULL) < 0)
17            perror("execl error");
18    }
19    if(waitpid(pid, NULL, 0) < 0)
20    {
21        perror("wait error");
22    }
23
24    return 0;
25 }
26

```

XY\_parent.c

```

int main()
{
    for(int i =0; i<10; i++){
        printf("Nyiri Levente, F023QC\n");
    }
    return 0;
}

```

XY\_child.c

#### 4. feladat

```

int main()
{
    pid_t pid;
    if((pid=fork())<0){
        perror("fork error")
    }
    else if(pid==0){

        if(exclp("ls","-l","/home/levente/04_GYAK",NULL)<0){
            perror("execl error")
        }

        if(waitpid(pid,NULL,0)<0){
            perror("wait error")
        }
        return 0;
    }
}

```

#### 5. feladat

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/wait.h>
4 #include <sys/types.h>
5 #include <unistd.h>
6
7 int main()
8 {
9     pid_t pid, status;
10
11     if((pid = fork()) < 0)
12     {
13         perror("Hiba a forkban!");
14         exit(7);
15     }
16     else if(pid == 0)
17     {
18         abort();
19         if(wait(&status) != pid)
20         {
21             perror("Hiba a wait-el!");
22         }
23     }
24     if(WIFEXITED(status))
25     {
26         printf("Sikeres! :");
27     }
28     return 0;
29 }

```

```

Sikeres! :)
Process returned 0 (0x0)   execution time : 0.006 s
Press ENTER to continue.

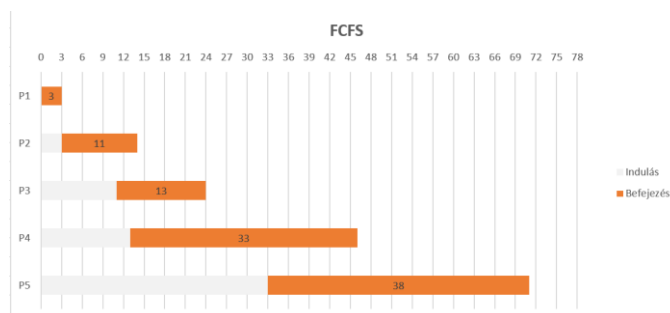
```

## 6. feladat

### FCFS megoldás:

FCFS	Érkezés	CPU idő	Indulás	Befejezés	Várakozás
P1	0	3	0	3	0
P2	1	8	3	11	2
P3	3	2	11	13	8
P4	9	20	13	33	4
P5	12	5	33	38	21

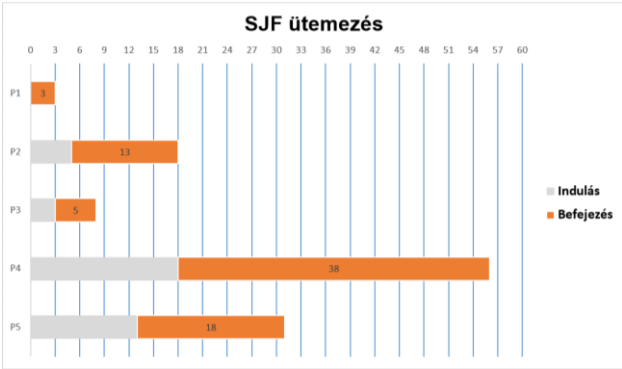
### Diagram:



### SJF megoldás:

SJF	Érkezés	CPU idő	Indulás	Befejezés	Várakozás
P1	0	3	0	3	0
P2	1	8	5	13	4
P3	3	2	3	5	0
P4	9	20	18	38	9
P5	12	5	13	18	1

Diagram:



Round Robin:

RR: 5ms	Érkezés	CPU idő	Indulás	Befejezés	Várakozás	Várakozó processz
P1	0	3	0	3	0	P2
P2	1	8	3	8	2	P2, P3
P3	3	2	8	10	5	P2, P4
P4	9	20	13	18	4	P4, P5
P5	12	5	18	23	6	P4

Diagram:

