Operációs rendszerek

Készítette: Novák László Lajos
Neptun: T8F0F0

1. Készítsen egy prant.c és egy child.c programot. a parent.c elindít egy gyermek processzt, ami különbözik a szülőtől. A szülő megvárja a gyermek lefutását. A gyermek szöveget ír a szabványos kimenetre (5x hallgató neve és neptun kódja)

parent.c:

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

int pid, ppid;

void main()
{
ppid=getpid();
if ((pid = fork()) == 0)
        {
        execl("./child", "ls", "-1", (char*)NULL);
        }
}
```

child.c:

```c
#include<stdio.h>

int i=0;
void main()
{
        for(i=0;i<5; i++)
        {
                printf("\n Novak Laszlo T8F0F0");
        }
        printf("\n");
}
```

Futtatás:

```
nlaszlo94@nlaszlo94-VirtualBox:~/Documents/beadando$ gcc child.c -o child
nlaszlo94@nlaszlo94-VirtualBox:~/Documents/beadando$ gcc parent.c -o parent
nlaszlo94@nlaszlo94-VirtualBox:~/Documents/beadando$ ./parent
nlaszlo94@nlaszlo94-VirtualBox:~/Documents/beadando$
 Novak Laszlo T8F0F0
 Novak Laszlo T8F0F0
 Novak Laszlo T8F0F0
 Novak Laszlo T8F0F0
 Novak Laszlo T8F0F0

nlaszlo94@nlaszlo94-VirtualBox:~/Documents/beadando$
```

2. Adott a következő terhelés esetén egy rendszer:

	P1	P2	P3	P4
Érkezés	0	8	12	20
CPU idő	15	7	26	10
Indulás	0	15	22	48
Befejezés				
Várakozás				

A tanult ütemezési algoritmus (FCFS, SJF, RR: 10ms) felhasználásával haározza meg

a, Várakozási/átlagos várakozási időt, befejezési időt

First Come First Served

FCFS	P1	P2	P3	P4	
Érkezés	0	8	12	20	
CPU idő	15	7	26	10	
Indulás	0	15	22	48	
Befejezés	15	22	48	58	Átlag várakozás
Várakozás	0	7	10	28	11,25
Sorrend	1	2	3	4	

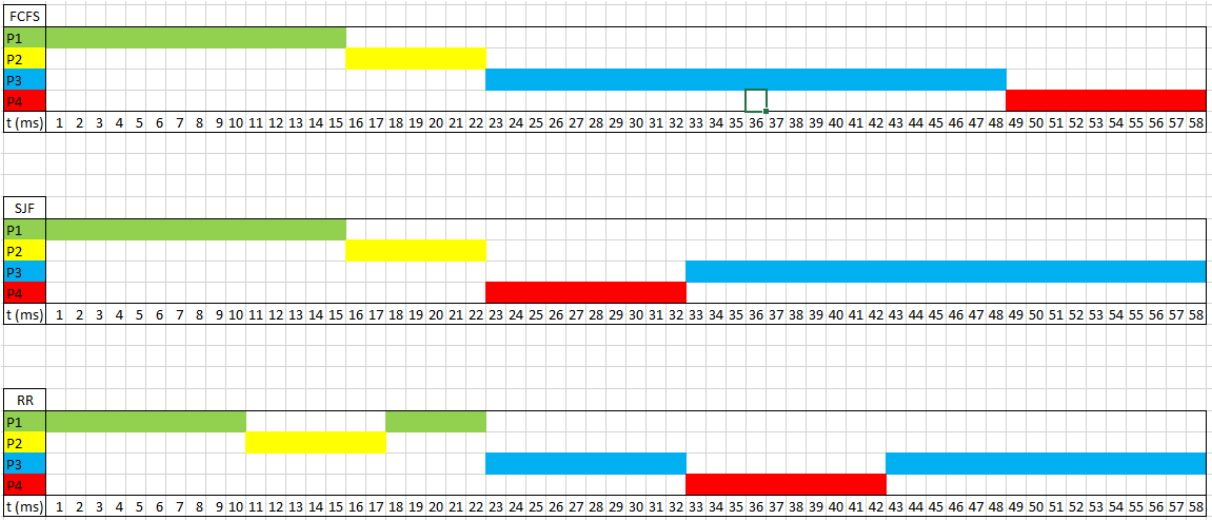
Shortest Job First

SJF	P1	P2	P3	P4	
Érkezés	0	8	12	20	
CPU idő	15	7	26	10	
Indulás	0	15	32	22	
Befejezés	15	22	58	32	Átlag várakozás
Várakozás	0	7	20	2	7,25
Sorrend	1	2	4	3	

Round Robin

RR	P1	P2	P3	P4	
Érkezés	0	8	12	20	
CPU idő	15	7	26	10	
Indulás	0	10	22	32	
Befejezés	22	17	58	42	Átlag várakozás
Várakozás	0+7	2	10+10	12	10,25
Sorrend	1; 3	2	4; 6	5	

b, Ábrázolja Gantt diagrammal az aktív/várakozó folyamatok futásának sorrendjét



4. a, Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy csővezetéket, a gyerek processz beleír egy szöveget a csővezetékbe (A kiírt szöveg: XY neptunkod), a szülő processz ezt kiolvassa, és kiírja a standard kimenetre.
Mentés: neptunkod_unnamed.c

T8F0F0_unnamed_parent:

```c
#include <stdio.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h>

#define BUFFER 1024

int main()
{
    int pid;
    int fd[2];
    char buf[BUFFER];

    /* pipe elkészítése (névtelen csővezeték) */
    pipe(fd);

    /*gyerek indítása*/
    if((pid=fork()) == 0)
    {
        close(fd[0]);
        write(fd[1], "Novak Laszlo", sizeof("Novak Laszlo"));
        close(fd[1]);
    } else {
        close(fd[1]);
        read(fd[0], buf, BUFFER);
        close(fd[0]);
    }

    /* üzenet fogadasa */
    wait(NULL);
    printf("%s\n",buf);

    return 0;
}
```

futtatás:

```
nlaszlo94@nlaszlo94-VirtualBox:~/Documents/beadando/nevtelen$ gcc T8F0F0_unnamed_parent.c -o parent
nlaszlo94@nlaszlo94-VirtualBox:~/Documents/beadando/nevtelen$ ./parent
??^?
Novak Laszlo
nlaszlo94@nlaszlo94-VirtualBox:~/Documents/beadando/nevtelen$
```

b. , Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy nevesített csővezetéket (neve: neptunkod), a gyerek processz beleír egy szöveget a csővezetékbe (A hallgató neve: pl. Keserű Ottó), a szülő processz ezt kiolvassa, és kiírja a standard kimenetre.
Mentés: neptunkod_named.c

T8F0F0_named_parent:

```c
#include <stdio.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

#define BUFFER 1024

int main()
{
    int ff, pid;
    char * myfifo = "/tmp/myfifo";
    char buf[BUFFER];

    /* FIFO elkészítése (névvel ellátott csővezeték) */
    mkfifo(myfifo, 0666);

    /*gyerek létrehozása*/
    if((pid=fork()) == 0)
        {
            execl("./child",  "ls", "-l", (char*)NULL);
        }

    /* üzenet fogadasa */
    ff = open(myfifo, O_RDONLY);
    read(ff, buf, BUFFER);
    printf("%s\n",buf);
    close(ff);

    /* FIFO eltávolítása */
    unlink(myfifo);

    return 0;
}
```

T8F0F0_named_child:

```c
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <unistd.h>

#define BUFFER 1024

int main()
{
        int ff;
        char * myfifo = "/tmp/myfifo";
        char buf[BUFFER];

        /* Adott szöveg a fifoba */
        ff = open(myfifo, O_WRONLY);
        write(ff, "Novak Laszlo", sizeof("Novak Laszlo"));
        close(ff);

        return 0;
}
```

futtatál:

```
nlaszlo94@nlaszlo94-VirtualBox:~/Documents/beadando/nevesitett$ gcc T8F0F0_named_parent.c -o parent
nlaszlo94@nlaszlo94-VirtualBox:~/Documents/beadando/nevesitett$ gcc T8F0F0_named_child.c -o child
nlaszlo94@nlaszlo94-VirtualBox:~/Documents/beadando/nevesitett$ ./parent
Novak Laszlo
nlaszlo94@nlaszlo94-VirtualBox:~/Documents/beadando/nevesitett$
```