

# Operációs rendszerek BSc

5.Gyak

2022.03.08.

**Készítette:**

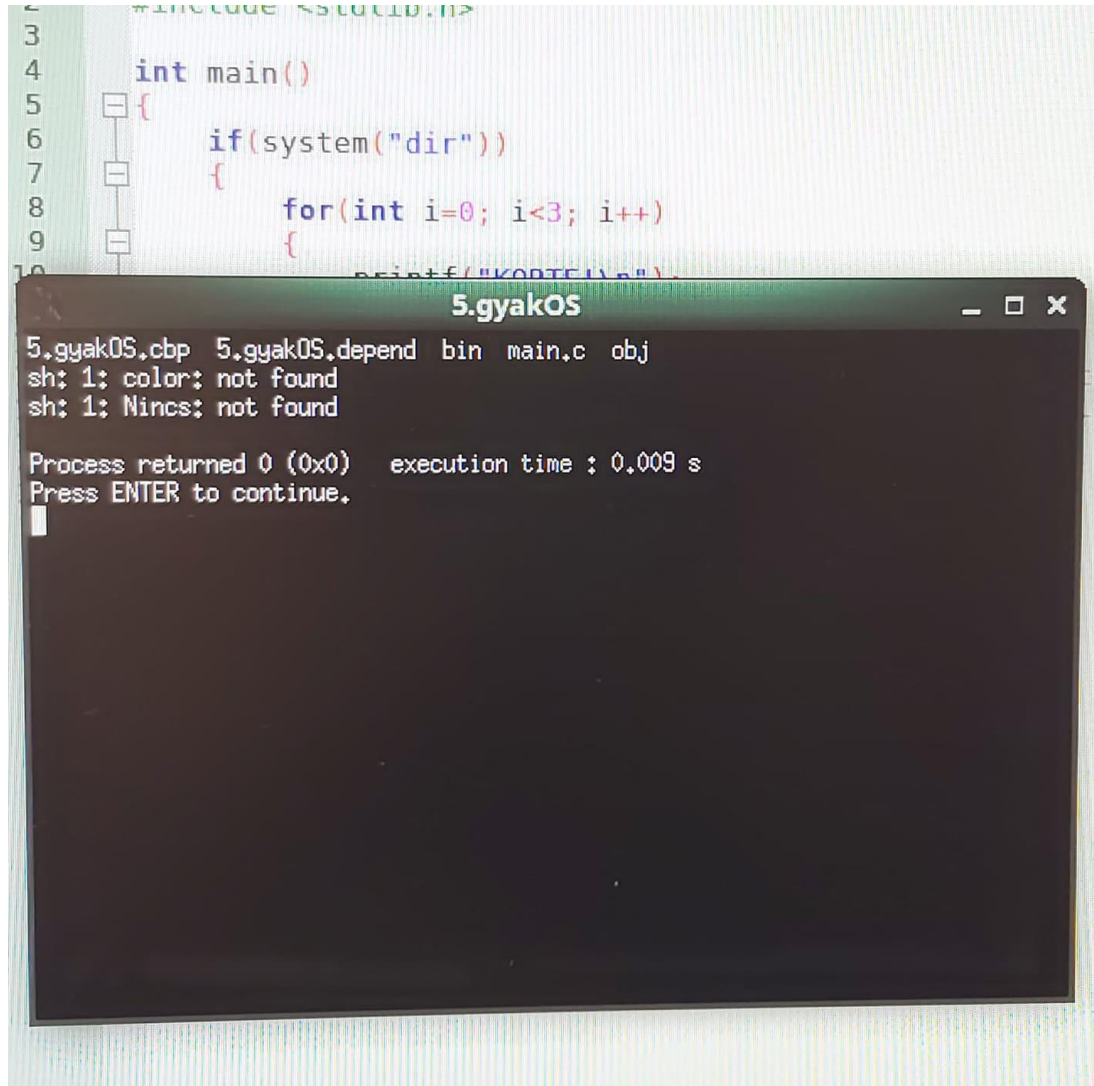
Tucsa Eszter Boglárka  
Mérnökinformatikus BSc  
G2QWPO

## 1.feladat:

„1. A `system()` rendszerhívással hajtson végre létező és nem létező parancsot, és vizsgálja a visszatérési értéket, magyarázza egy-egy mondattal

A fordítás/futtatás után készítsen egy képernyőképet (minden parancs esetén) és illessze be a dokumentumba.

Mentés: *neptunkodIfel.c*



```
1 //include <stdio.h>
2
3
4 int main()
5 {
6     if(system("dir"))
7     {
8         for(int i=0; i<3; i++)
9         {
10             printf("#%03d\n", i);
11         }
12     }
13 }
```

5.gyakOS

5.gyakOS.cbp 5.gyakOS.depend bin main.c obj

sh: 1: color: not found

sh: 1: Nincs: not found

Process returned 0 (0x0) execution time : 0.009 s

Press ENTER to continue.

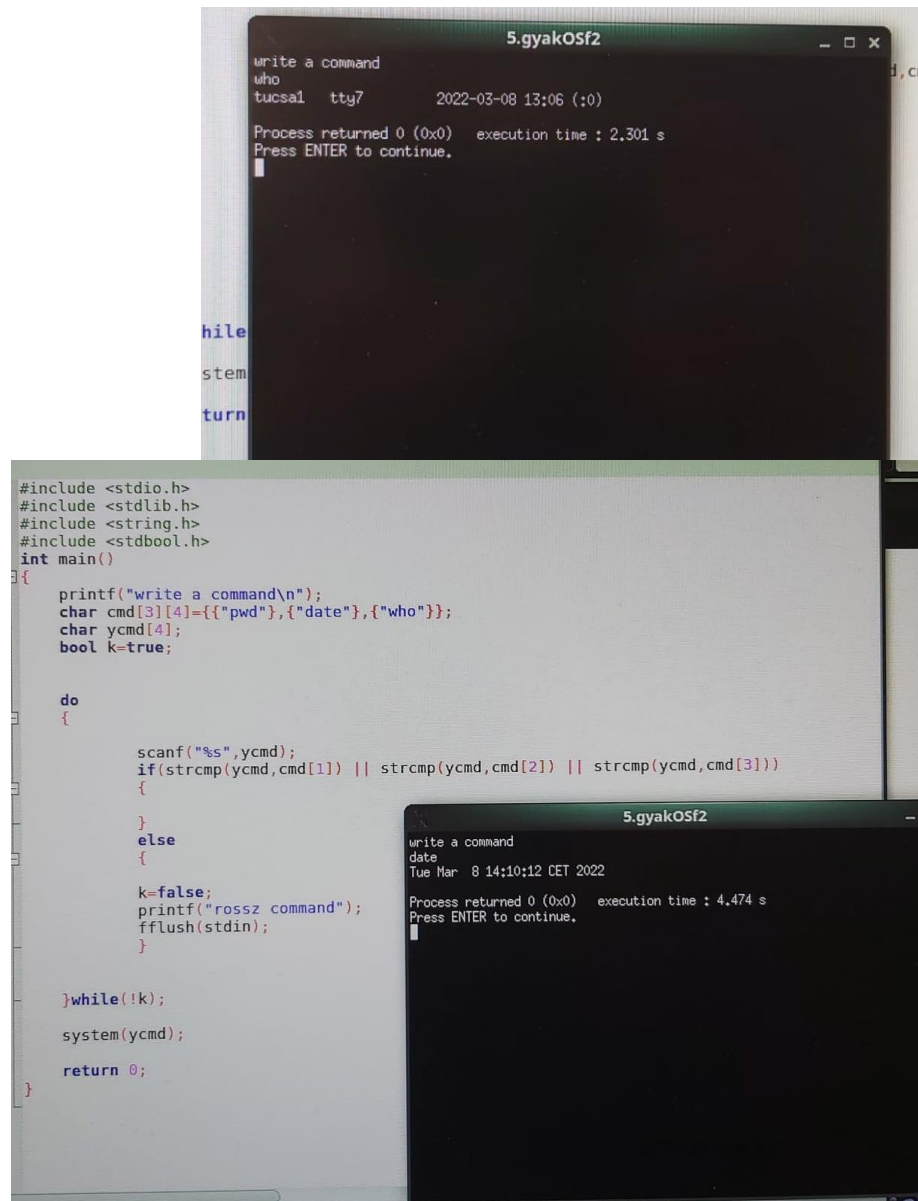
//a kód a kódmappában van github-on, képernyőkép helyett lefényképezve a gyakorlat idejében, mert én és a jelszavakra emlékezés két külön fogalom, így viszont biztosan megvan minden.

Itt nem lehetett kipróbálni a `color` commandot, de terminálban működött, zölddé tette a szövegeket.

**2.feladat:** 2. Írjon programot, amely billentyűzetről bekér Unix parancsokat és végrehajtja őket, majd kiírja a szabványos kimenetre. (pl.: amit bekér: date, pwd, who etc.; kilépés: CTRL-\\) - magyarázza egy-egy mondat

A fordítás/futtatás után készítsen egy képernyőképet (minden parancs esetén) és illessze be a dokumentumba.

Mentés: *neptunkod2fel.c*



```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
int main()
{
    printf("write a command\n");
    char cmd[3][4]={"pwd"},{"date"},{"who"};
    char ycmd[4];
    bool k=true;

    do
    {
        scanf("%s",ycmd);
        if(strcmp(ycmd,cmd[1]) || strcmp(ycmd,cmd[2]) || strcmp(ycmd,cmd[3]))
        {
        }
        else
        {
            k=false;
            printf("rossz command");
            fflush(stdin);
        }

    }while(!k);

    system(ycmd);

    return 0;
}
```

```
5.gyakOSf2
write a command
who
tucsal  tty7      2022-03-08 13:06 (:0)

Process returned 0 (0x0)  execution time : 2.301 s
Press ENTER to continue.
```

```
5.gyakOSf2
write a command
date
Tue Mar  8 14:10:12 CET 2022

Process returned 0 (0x0)  execution time : 4.474 s
Press ENTER to continue.
```

Minden parancs működött, pwd is, bár azt nem fényképeztem le.

### 3-5.feladat:

3. Készítsen egy `parent.c` és a `child.c` programokat. A `parent.c` elindít egy gyermek processzt, ami különbözik a szülőtől. A szülő megvárja a gyermek lefutását. A gyermek szöveget ír a szabványos kimenetre (10-ször) (pl. a hallgató neve és a neptunkód)! - magyarázza egy-egy mondattal. A fordítás/futtatás után készítsen egy képernyőképet (minden parancs esetén) és illessze be a dokumentumba.

Mentés: `parent.c`, ill. `child.c`

4. A `fork()` rendszerhívással hozzon létre egy gyerek processzt és abban hívjon meg egy `exec` családbeli rendszerhívást (pl. `execip`). A szülő várja meg a gyerek futását! - magyarázza egy-egy mondattal.

A fordítás/futtatás után készítsen egy képernyőképet (minden parancs esetén) és illessze be a dokumentumba.

Mentés: `neptunkod4fel.c`

5. A `fork()` rendszerhívással hozzon létre gyerekeket, várja meg és vizsgálja a befejeződési állapotokat (gyerekekben: `exit`, `abort`, nullával való osztás)! - magyarázza egy-egy mondattal!

Ez egy olyan feladat volt, aminek futtatásával a console ablakban nem sok minden jött elő. Konkrétan csak a „Process returned 0”, mint az előző képeknek a végén is. Itthon nem működött Windows-on. A kódok a github mappában.

### 6.feladat:

6.

Adott a következő ütemezési feladat, amit a FCFS, SJF és Round Robin (RR) ütemezési algoritmus használatával készítsen el (külön-külön táblázatba):

**I. Határozza meg FCFS és SJF esetén**

a.) A befejezési időt?

b.) A várakozási/átlagos várakozási időt?

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

Megj.: a Gantt diagram ábrázolása szerkesztő program segítségével vagy Excel programmal.

Mentés: `neptunkod6fel.pdf`

FCFS

| FCFS | Érkezés | CPU idő |
|------|---------|---------|
| P1   | 0       | 3       |
| P2   | 1       | 8       |
| P3   | 3       | 2       |
| P4   | 9       | 20      |
| P5   | 12      | 5       |

FCFS megoldás

| FCFS | Érkezés | CPU idő | Indulás | Befejezés | Várakozás |
|------|---------|---------|---------|-----------|-----------|
|      |         |         |         |           |           |
|      |         |         |         |           |           |
|      |         |         |         |           |           |
|      |         |         |         |           |           |

SJF

| SJF | Érkezés | CPU idő |
|-----|---------|---------|
| P1  | 0       | 3       |
| P2  | 1       | 5       |
| P3  | 3       | 2       |
| P4  | 9       | 5       |
| P5  | 12      | 5       |

SJF megoldás

| SJF | Érkezés | CPU idő | Indulás | Befejezés | Várakozás | Legrövidebb |
|-----|---------|---------|---------|-----------|-----------|-------------|
|     |         |         |         |           |           |             |
|     |         |         |         |           |           |             |
|     |         |         |         |           |           |             |
|     |         |         |         |           |           |             |

## II. Round Robin (RR) esetén

- a.) Ütemezze az adott időszletet (5ms) alapján az egyes processzek (befejezési és várakozási/átlagos várakozási idő) paramétereit (ms)!
- b.) A rendszerben lévő processzek végrehajtásának sorrendjét?
- c.) Ábrázolja Gantt diagram segítségével az *aktív/várakozó* processzek futásának menetét!

Megj.: a Gantt diagram ábrázolása szerkesztő program segítségével vagy Excel programmal.

Mentés: neptunkodtjel.pdf

| RR: 5ms | Érkezés | CPU idő |
|---------|---------|---------|
| P1      | 0       | 3       |
| P2      | 1       | 8       |
| P3      | 3       | 2       |
| P4      | 9       | 20      |
| P5      | 12      | 5       |

RR megoldás

| RR: 5ms | Érkezés | CPU idő | Indulás | Befejezés | Várakozás | Várakozó processz |
|---------|---------|---------|---------|-----------|-----------|-------------------|
|         |         |         |         |           |           |                   |
|         |         |         |         |           |           |                   |
|         |         |         |         |           |           |                   |
|         |         |         |         |           |           |                   |

| FCFS | Érkezés | CPU idő |
|------|---------|---------|
| P1   | 0       | 3       |
| P2   | 1       | 8       |
| P3   | 3       | 2       |
| P4   | 9       | 20      |
| P5   | 12      | 5       |

| FCFS | Érkezés | CPU idő | Indulás | Bef. | Vár. |
|------|---------|---------|---------|------|------|
| 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   |

Végrehajtási sorrend: P1-P2-P3-P4-P5

$$\text{Átlagos várakozási idő: } \frac{\sum_{i=1}^5 P_i \cdot \text{vár.}}{5} = \frac{35}{5} = 7$$

| SJF | Érkezés | CPU idő |
|-----|---------|---------|
| P1  | 0       | 3       |
| P2  | 1       | 5       |
| P3  | 3       | 2       |
| P4  | 9       | 5       |
| P5  | 12      | 5       |

| SJF | Érkezés | CPU idő | Indulás | Befejezés | Várakozási |
|-----|---------|---------|---------|-----------|------------|
| P3  | 3       | 2       | 0       | 2         | 0          |
| P1  | 0       | 3       | 2       | 5         | 2          |
| P2  | 1       | 5       | 5       | 10        | 4          |
| P4  | 9       | 5       | 10      | 15        | 1          |
| P5  | 12      | 5       | 15      | 20        | 3          |

Végrehajtási sorrend: ~~P1-P2-P3-P4-P5~~

P3-P1-P2-P4-P5 (mivel a legrövidebbel kezdünk)

$$\text{Átlagos várakozási idő: } \frac{10}{5} = 2$$

## II. Round-Robin

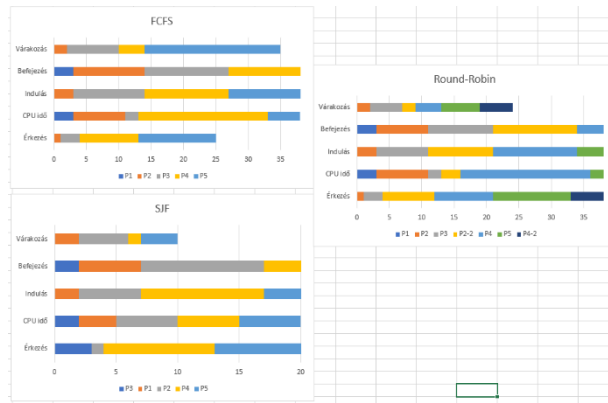
| RR: 5ms | Érkezés | CPU idő |
|---------|---------|---------|
| P1      | 0       | 3       |
| P2      | 1       | 8       |
| P3      | 3       | 2       |
| P4      | 9       | 20      |
| P5      | 12      | 5       |

| RR: 5ms | Érkezés | CPU idő | Indulás | Befejezés | Várakozás | Várakozó proc. |
|---------|---------|---------|---------|-----------|-----------|----------------|
| P1      | 0       | 3       | 0       | 3         | 0         | P2             |
| P2      | 1; 8    | 8; 3    | 3; 10   | 8; 13     | 2; 2      | P3; P4         |
| P3      | 3       | 2       | 8       | 10        | 5         | P2             |
| P4      | 9; 18   | 20; 15  | 13; 23  | 18; 38    | 4; 5      | P5             |
| P5      | 12      | 5       | 18      | 23        | 6         | P4             |

Végrehajtási sorrend: P1-P2-P3-P2-P4-P5-P4

//, helyett; hogy ne keverjem össze a

$$\text{Átlagos várakozási idő: } \frac{24}{7} = 3,43 \text{ tizedes számmal}$$



A minta példa és a képletek alapján kiszámoltam mindent kézileg, az adatokat beírtam excel-be és Grantt-diagramot készítettem belőle (legalábbis valami hasonlót).