

# Operációs rendszerek BSc

5. Gyak.

2022. 03. 08.

Készítette:

Boján Miron Noel Bsc

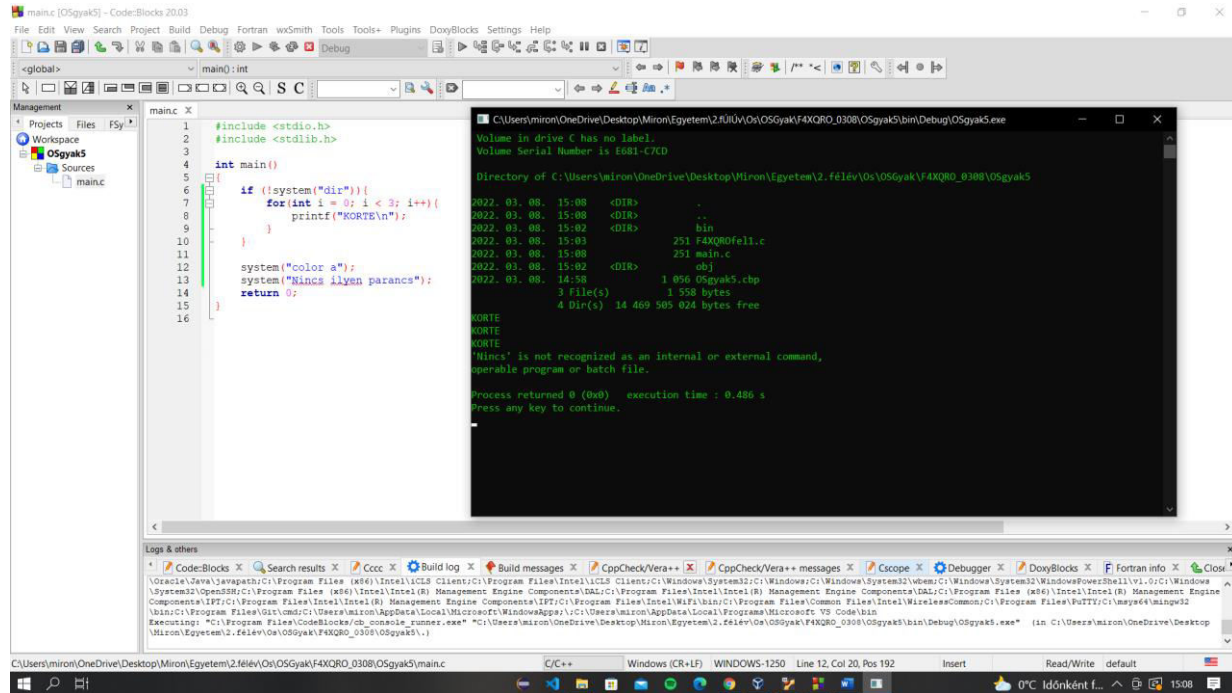
Programtervező informatikus

F4XQRO

Miskolc, 2022

## 1. feladat

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.



The screenshot shows the Code::Blocks IDE with a C++ project named 'OSgyak5'. The source file 'main.c' contains the following code:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     if (!system("dir")){
7         for(int i = 0; i < 3; i++){
8             printf("KORTE\n");
9         }
10    }
11
12    system("color a");
13    system("Nincs ilyen parancs");
14    return 0;
15 }
```

The execution output window shows the following results:

```
Volume in drive C has no label.
Volume Serial Number is E681-C7CD

Directory of C:\Users\miron\OneDrive\Desktop\Miron\Egyetem\2.fel\Os\OSgyak\F4XQRO_0308\OSgyak5

2022. 03. 08. 15:08 <DIR>          .
2022. 03. 08. 15:08 <DIR>          ..
2022. 03. 08. 15:02 <DIR>          bin
2022. 03. 08. 15:03                251 F4XQROfell.c
2022. 03. 08. 15:08                251 main.c
2022. 03. 08. 15:02 <DIR>          obj
2022. 03. 08. 14:58                1 056 OSgyak5.chp
                3 file(s)              1 536 bytes
                4 dir(s) 14 469 505 024 bytes free

KORTE
KORTE
KORTE

'Nincs' is not recognized as an internal or external command,
operable program or batch file.

Process returned 0 (0x0)   execution time : 0.486 s
Press any key to continue.
```

The status bar at the bottom indicates the file path: `C:\Users\miron\OneDrive\Desktop\Miron\Egyetem\2.fel\Os\OSgyak\F4XQRO_0308\OSgyak5\main.c`.

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 mondattal

The screenshot shows a C++ IDE with a project named 'F4XQROfe12'. The source file 'main.c' contains the following code:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6
7     char command[50];
8     scanf("%s", &command);
9
10    system(command);
11
12    return 0;
13 }
14

```

The console window shows the program's execution:

```

date
The current date is: 2022. 03. 08.
Enter the new date: (yy-mm-dd)

```

The status bar at the bottom indicates the file is 'main.c' at line 6, column 1, position 58.

The screenshot shows the same C++ IDE with the same source file 'main.c'. The console window shows the program's execution after running a 'dir' command:

```

dir
Volume in drive C has no label.
Volume Serial Number is E681-C7CD

Directory of C:\Users\miron\OneDrive\Desktop\Miron\Egyetem\2.fel\OssGyak\F4XQRO_0308\F4XQROfe12

2022. 03. 08. 15:34 <DIR> .
2022. 03. 08. 15:34 <DIR> ..
2022. 03. 08. 15:15 <DIR> bin
2022. 03. 08. 15:14 1 065 F4XQROfe12.chp
2022. 03. 08. 15:34 172 F4XQROfe12.depend
2022. 03. 08. 15:34 155 main.c
2022. 03. 08. 15:15 <DIR> obj
3 File(s) 1 392 bytes
4 Dir(s) 14 468 128 768 bytes free

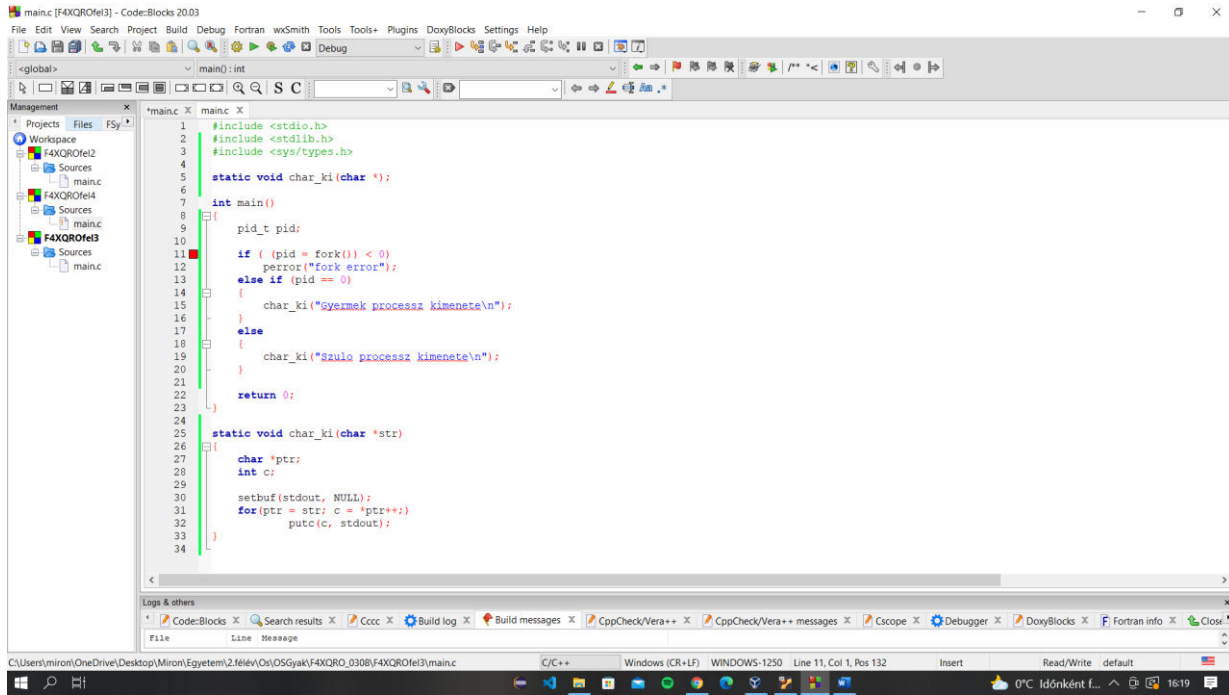
Process returned 0 (0x0) execution time : 1.188 s
Press any key to continue.

```

The status bar at the bottom indicates the file is 'main.c' at line 6, column 1, position 58.

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

4.) A fork() rendszerhívással hozzon létre egy gyerek processzt-t és abban hívjon meg egy exec családbeli rendszerhívást (pl. execlp). 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.



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/types.h>
4
5 static void char_ki(char *);
6
7 int main()
8 {
9     pid_t pid;
10
11     if ( (pid = fork()) < 0 )
12         perror("fork error");
13     else if (pid == 0)
14     {
15         char_ki("Gyerek processz kimenete\n");
16     }
17     else
18     {
19         char_ki("Szulo processz kimenete\n");
20     }
21
22     return 0;
23 }
24
25 static void char_ki(char *str)
26 {
27     char *ptr;
28     int c;
29
30     setbuf(stdout, NULL);
31     for(ptr = str; c = *ptr++;)
32         putchar(c, stdout);
33 }
34
```

5.) A fork() rendszerhívással hozzon létre gyerekeket, várja meg és vizsgálja a befejeződési állapotokat (gyerekekben: exit, abort, nullával való osztás)! - magyarázza egy-egy mondattal! A fordítás/futtatás után készítsen egy képernyőképet (minden parancs esetén) és illessze be a dokumentumba.