

# **Operációs rendszerek BSc**

**5. Gyak.**

2022. 03. 08.

**Készítette:**

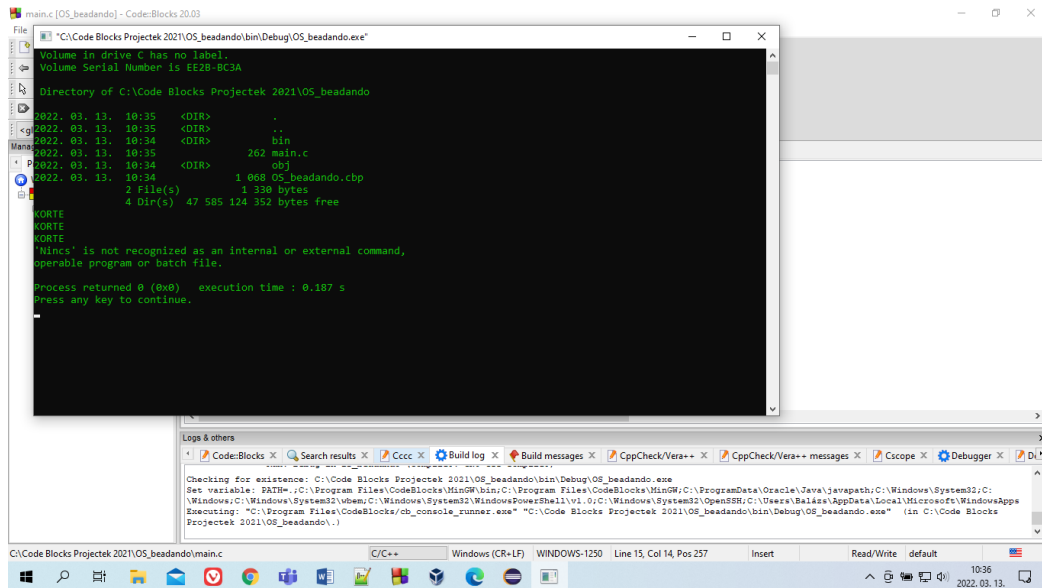
Petró Balázs Bsc  
Mérnökinformatikus  
FO71M2

**Miskolc, 2022**

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: neptunkod1fel.c

A `system(„dir”) rendszerhívással létező parancs végre hajtása.`



```
main.c [OS_beadando] - Code::Blocks 20.03
"C:\Code Blocks Projektek 2021\OS_beadando\bin\Debug\OS_beadando.exe"
Volume in drive C has no label:
Volume Serial Number is EE28-BC3A

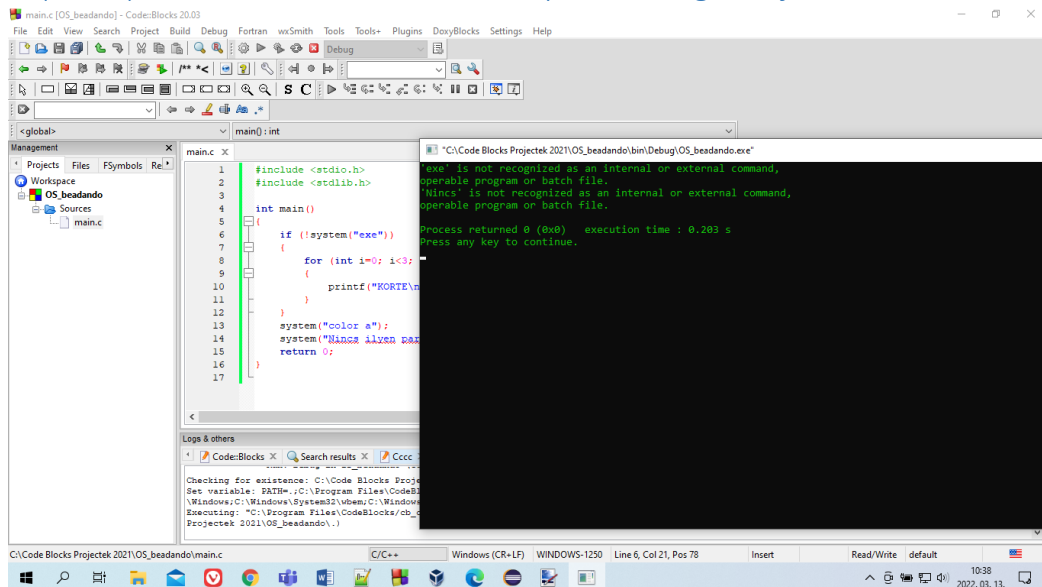
Directory of C:\Code Blocks Projektek 2021\OS_beadando

2022. 03. 13. 10:35 <DIR>      .
2022. 03. 13. 10:35 <DIR>      ..
2022. 03. 13. 10:34 <DIR>      bin
2022. 03. 13. 10:35 <DIR>      262 main.c
2022. 03. 13. 10:34 <DIR>      obj
2022. 03. 13. 10:34          1 068 OS_beadando.cbp
2 File(s)              1 330 bytes
4 Dir(s)  47 585 124 352 bytes free

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

Process returned 0 (0x0)   execution time : 0.187 s
Press any key to continue.
```

A `system(„exe”) rendszerhívással nem létező parancs végre hajtása.`



```
main.c [OS_beadando] - Code::Blocks 20.03
"C:\Code Blocks Projektek 2021\OS_beadando\bin\Debug\OS_beadando.exe"

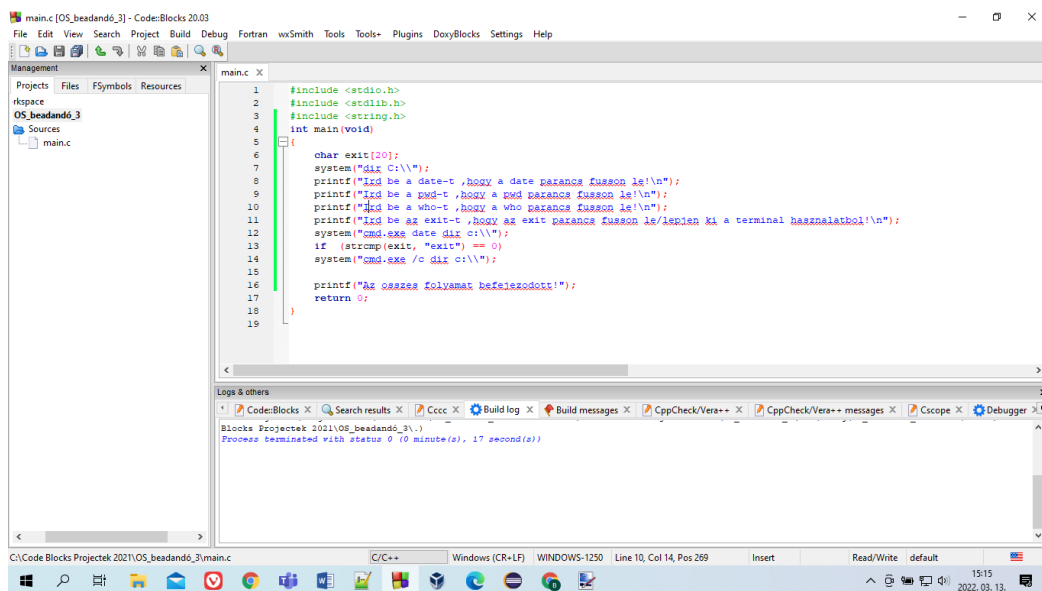
'exe' is not recognized as an internal or external command,
operable program or batch file.
'Wincs' is not recognized as an internal or external command,
operable program or batch file.

Process returned 1 (0x1)   execution time : 0.203 s
Press any key to continue.
```

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

Mentés: neptunkod2fel.c

A terminál megnyitása a feladat elindításával. Akármelyik parancsot begépelve, amit alpból el lehet, azon az operációs rendszeren végezni azt elvégzi. Ugyanakkor az „exit” szó beírására a terminál ablakon a „Press any key to continue”-ra jelenik meg, ami után akármelyik gomb lenyomása után kilép a terminálból.



The screenshot shows the Code::Blocks IDE interface. The main window displays the source code for 'main.c', which is a C program designed to execute system commands and handle the exit key (CTRL-\). The code includes headers for stdio, stdlib, and string, and defines a main function that uses printf to prompt the user for commands, system() to execute them, and a loop to handle the exit key. The output window at the bottom shows the program's execution, indicating it terminated successfully after 17 seconds.

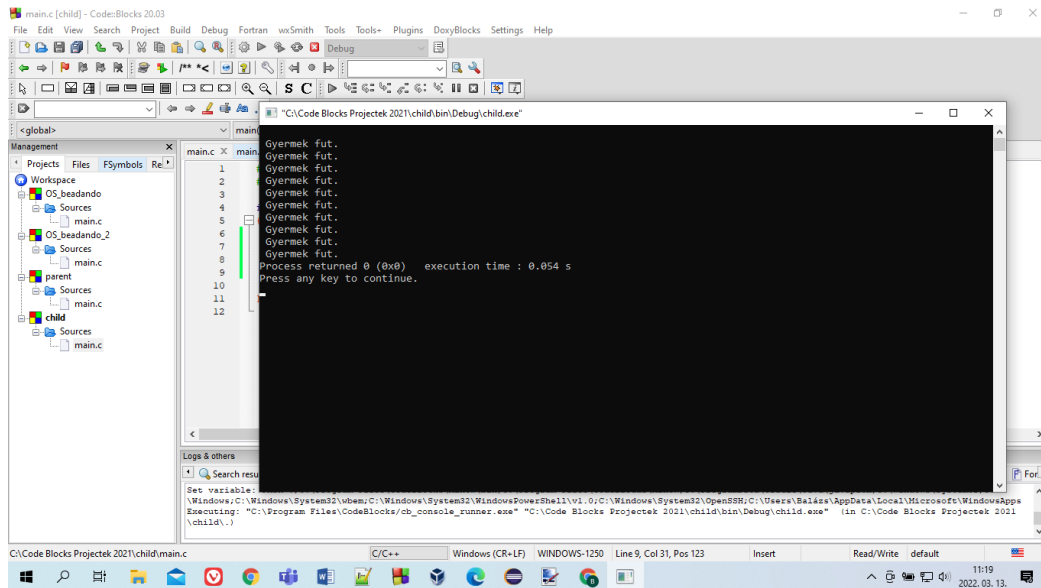
```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 int main(void)
5 {
6     char exit[20];
7     system("dir c:\\");
8     printf("Írj be a date-t ,hogy a date parancs fusson le!\n");
9     printf("Írj be a pwd-t ,hogy a pwd parancs fusson le!\n");
10    printf("Írj be a who-t ,hogy a who parancs fusson le!\n");
11    printf("Írj be az exit-t ,hogy az exit parancs fusson le/lepien ki a terminal hasznalatból!\n");
12    system("cmd.exe date dir c:\\");
13    if (strcmp(exit, "exit") == 0)
14        system("cmd.exe /c dir c:\\");
15
16    printf("Az összes folyamat befejeződött!");
17    return 0;
18 }
19
```

Output window logs:

```
Blocks Projectek 2021\08_beadandó_3\...
Process terminated with status 0 (0 minute(s), 17 second(s))
```



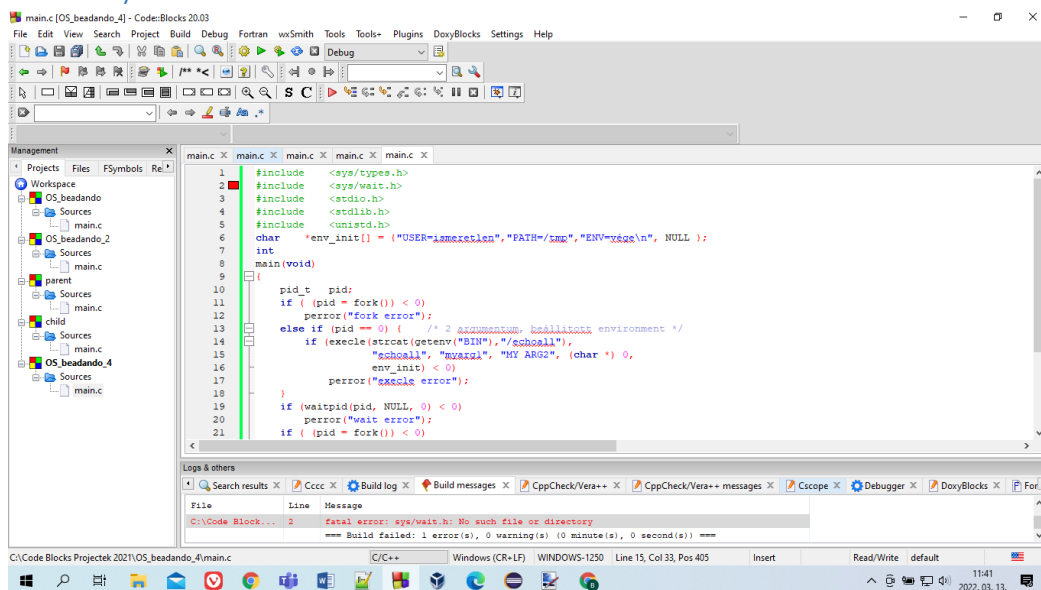
## A child.c futtatása utáni eredmény:



- 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.**

**Mentés: neptunkod4fel.c**

A program még fordítás előtt fordítási hibába ütközik a sys/wait.h header állományt nem ismeri fel.



```

10 pid_t pid;
11 if ( (pid = fork()) < 0)
12     perror("fork error");
13 else if (pid == 0) { /* 2 argument, echoall environment */
14     if (execlp("echoall", "echoall", "MY ARG2", (char *) 0,
15               env_init) < 0)
16         perror("execlp error");
17     }
18 if (waitpid(pid, NULL, 0) < 0)
19     perror("wait error");
20 if ( (pid = fork()) < 0)
21     perror("fork error");
22 else if (pid == 0) { /* 1 arg, echoall environment */
23     if (execlp("echoall", "echoall", "only 1 arg", (char *) 0) < 0)
24         perror("execlp error");
25     }
26     exit(0);
27 }
28 }
29 }
30 }

```

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

**Mentés: neptunkod5fel.c**

A fork() rendszerhívás nem működik ennél a feladatnál/ fordítási hibát kapok.

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/types.h>
4 #include <fcntl.h>
5 #include <unistd.h>
6 pid_t pid;
7 //pid_t fork();
8
9 int main()
10 {
11     int ppid, pid, qpid;
12     ppid = getpid();
13     if ( (pid = fork()) == 0)
14     {
15         qpid = getpid();
16         exit(0);
17     }
18     exit(1);
19     return 0;
20 }

```

error: 1d returned 1 exit status  
Build failed: 2 error(s), 3 warning(s) (0 minute(s), 0 second(s))