

# **Operációs rendszerek BSc**

**1. Gyak.**

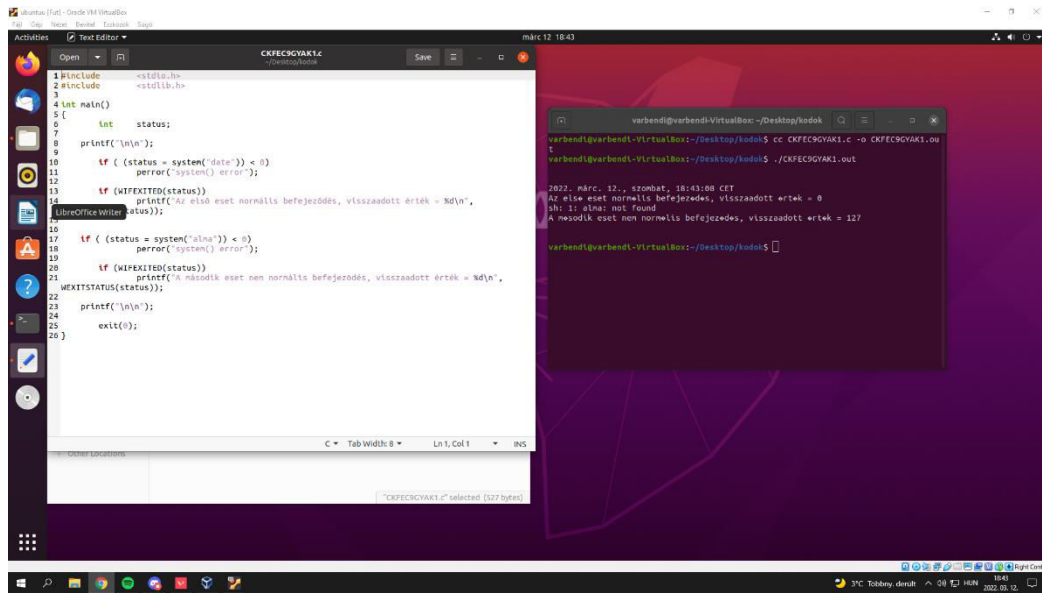
**2022. 02. 08.**

**Készítette:**

Varga Bence  
Mérnökinformatika  
CKFEC9

**Miskolc, 2022**

## 1. feladat –

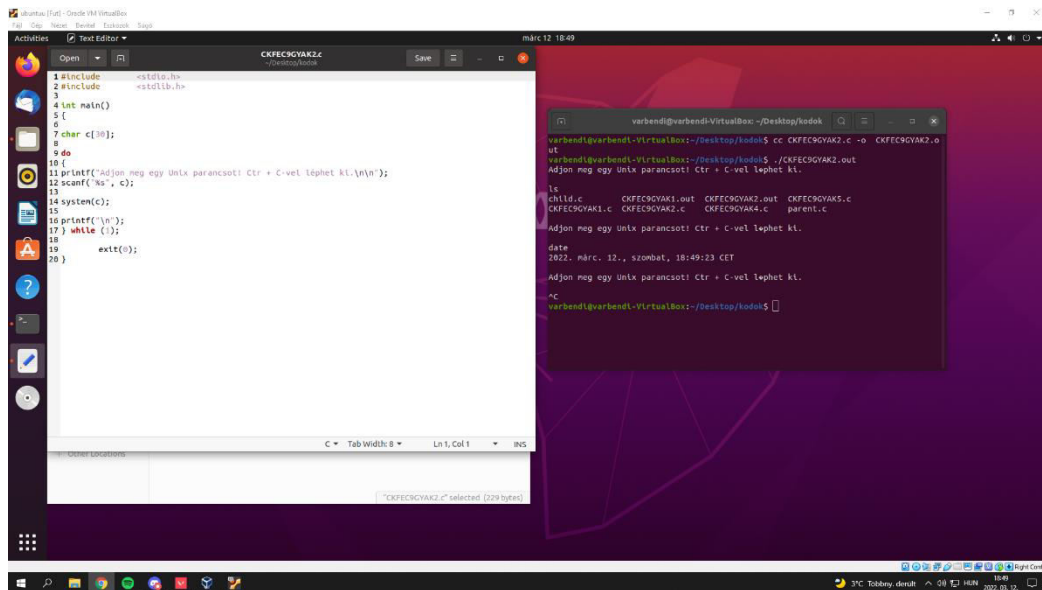


The screenshot shows a Linux desktop with a terminal window and a code editor. The code editor displays a C program named `CKFECSGYAK1.c` that uses `system()` to execute `date` and `ls` commands. The terminal window shows the compilation and execution of this program, resulting in the current date and the contents of the `parent.c` file being printed.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int status;
    printf("\n\n");
    if ( (status = system("date")) < 0 )
        perror("system() error");
    if (WIFEXITED(status))
        printf("Az első eset normális befejeződés, visszaadott érték = %d\n",
            WEXITSTATUS(status));
    if ( (status = system("ls")) < 0 )
        perror("system() error");
    if (WIFEXITED(status))
        printf("A második eset nem normális befejeződés, visszaadott érték = %d\n",
            WEXITSTATUS(status));
    printf("\n\n");
    exit(0);
}
```

```
varbendi@varbendi-VirtualBox: ~/Desktop/kodok
varbendi@varbendi-VirtualBox:~/Desktop/kodok$ cc CKFECSGYAK1.c -o CKFECSGYAK1.out
varbendi@varbendi-VirtualBox:~/Desktop/kodok$ ./CKFECSGYAK1.out
2022. márc. 12., szombat, 18:43:08 CET
ls: else eset normális befejeződés, visszaadott érték = 0
ls: ls: alina: not found
A második eset nem normális befejeződés, visszaadott érték = 127
varbendi@varbendi-VirtualBox:~/Desktop/kodok$
```

## 2. feladat –



The screenshot shows a Linux desktop with a terminal window and a code editor. The code editor displays a C program named `CKFECSGYAK2.c` that reads a command from standard input and executes it using `system()`. The terminal window shows the compilation and execution of this program, where the user enters `ls` and `date`, and the program outputs the results of these commands.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    char c[10];
    do
    {
        printf("Adjon meg egy Unix parancsot! Ctr + C-vel lehet ki.\n\n");
        scanf("%s", c);
        system(c);
        printf("\n");
    } while (!);
    exit(0);
}
```

```
varbendi@varbendi-VirtualBox: ~/Desktop/kodok
varbendi@varbendi-VirtualBox:~/Desktop/kodok$ cc CKFECSGYAK2.c -o CKFECSGYAK2.o
varbendi@varbendi-VirtualBox:~/Desktop/kodok$ ./CKFECSGYAK2.out
Adjon meg egy Unix parancsot! Ctr + C-vel lehet ki.
ls
child.c  CKFECSGYAK1.out  CKFECSGYAK2.out  CKFECSGYAK5.c
CKFECSGYAK1.c  CKFECSGYAK2.c  CKFECSGYAK6.c  parent.c
Adjon meg egy Unix parancsot! Ctr + C-vel lehet ki.
date
2022. márc. 12., szombat, 18:49:23 CET
Adjon meg egy Unix parancsot! Ctr + C-vel lehet ki.
^C
varbendi@varbendi-VirtualBox:~/Desktop/kodok$
```

The image shows a Linux desktop environment with a terminal window open. The terminal is running a C program that demonstrates process creation using `fork()` and `waitpid()`.

**Source Code (C):**

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

7 int main()
8 {
9     pid_t pid;
10    if (pid = fork() == -1)
11        perror("hiba");
12
13    else if (pid == 0) {
14        exec("/bin/ls", "ls", "-la", NULL);
15        printf("csoplok hiba\n");
16    }
17
18    else {
19        waitpid(pid, 0);
20    }
21
22    printf("\n gyermek process befejezte a futasat!\n");
23
24    return 0;
25 }
```

**Compilation:**

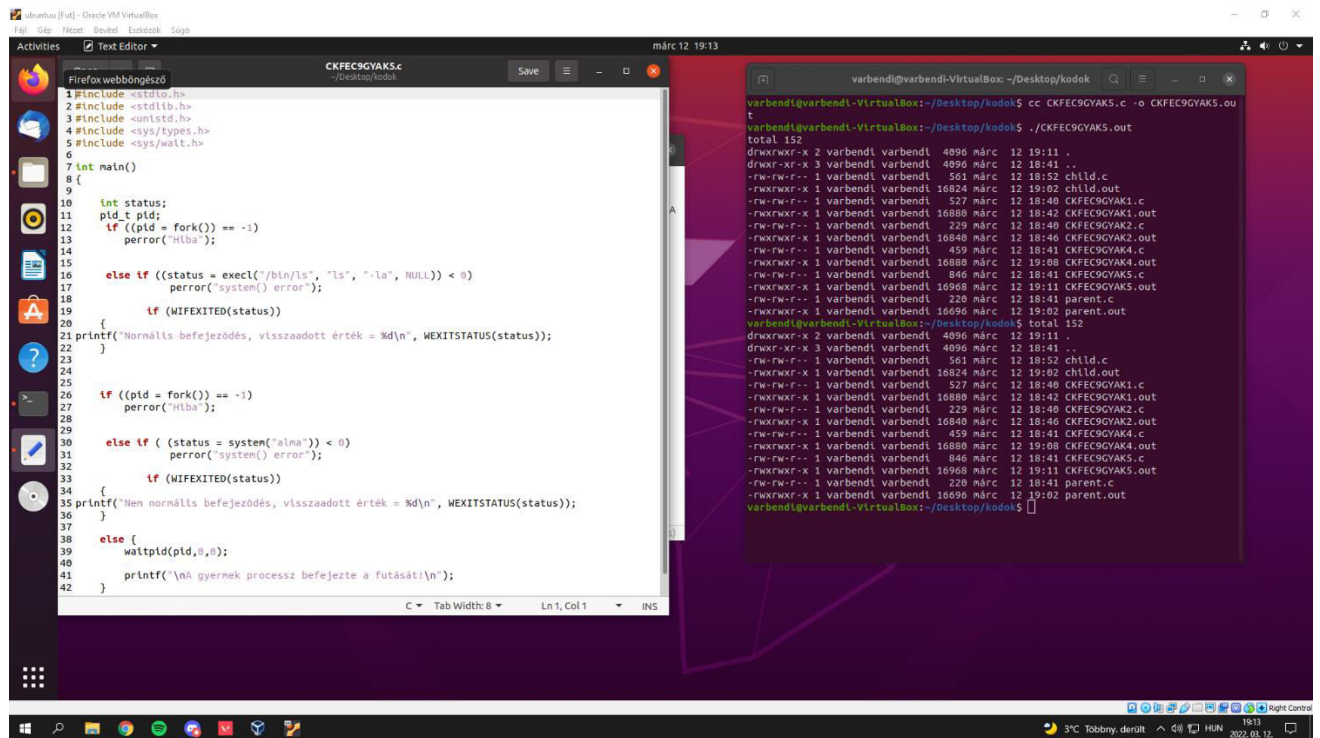
```
cc -o CFKFCSGYAK4.o CFKFCSGYAK4.c
```

**Execution Output:**

```
varbendi@varbendi-VirtualBox: ~/Desktop/kodok
varbendi@varbendi-VirtualBox:~/Desktop/kodok$ CC CFKFCSGYAK4.c -o CFKFCSGYAK4.o
total 132
drwxr-xr-x 2 varbendi varbendi 4096 marc 12 19:08
drwxr-xr-x 3 varbendi varbendi 4096 marc 12 18:41 ..
-rw-r--r-- 1 varbendi varbendi 568 marc 12 18:42 child.c
-rw-r--r-x 1 varbendi varbendi 16824 marc 12 19:02 child.out
-rw-r--r-- 1 varbendi varbendi 527 marc 12 18:40 CFKFCSGYAK1.c
-rw-r--r-x 1 varbendi varbendi 16880 marc 12 18:42 CFKFCSGYAK1.out
-rw-r--r-- 1 varbendi varbendi 229 marc 12 18:40 CFKFCSGYAK2.c
-rw-r--r-x 1 varbendi varbendi 16880 marc 12 18:40 CFKFCSGYAK2.out
-rw-r--r-- 1 varbendi varbendi 459 marc 12 18:41 CFKFCSGYAK3.c
-rw-r--r-x 1 varbendi varbendi 16880 marc 12 19:08 CFKFCSGYAK4.out
-rw-r--r-- 1 varbendi varbendi 846 marc 12 18:41 CFKFCSGYAK5.c
-rw-r--r-- 1 varbendi varbendi 220 marc 12 18:41 parent.c
-rw-r--r-x 1 varbendi varbendi 16696 marc 12 19:02 parent.out

A gyermek process befejezte a futasat!
varbendi@varbendi-VirtualBox:~/Desktop/kodok$
```

## 5. feladat –



The screenshot shows a Linux desktop environment with a terminal window and a text editor. The text editor displays a C program named `CKFEC9GYAK5.c` located at `~/Desktop/kodok`. The program uses `fork()` to create child processes and `system()` to execute system commands. The terminal window shows the output of the program, including the execution of `cc CKFEC9GYAK5.c -o CKFEC9GYAK5.out` and the execution of the program itself, which prints the total number of processes (152) and the execution of system commands.

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

int main()
{
    int status;
    pid_t pid;
    if ((pid = fork()) == -1)
        perror("Hiba");
    else if ((status = execl("/bin/ls", "ls", "-la", NULL)) < 0)
        perror("system() error");
    if (WIFEXITED(status))
    {
        printf("Normális befejeződés, visszaadott érték = %d\n", WEXITSTATUS(status));
    }
    if ((pid = fork()) == -1)
        perror("Hiba");
    else if ((status = system("alma")) < 0)
        perror("system() error");
    if (WIFEXITED(status))
    {
        printf("Nem normális befejeződés, visszaadott érték = %d\n", WEXITSTATUS(status));
    }
    else {
        waitpid(pid, 0, 0);
        printf("\nA gyermek processz befejezte a futását!\n");
    }
}
```

```
varbendi@varbendi-VirtualBox: ~/Desktop/kodok$ cc CKFEC9GYAK5.c -o CKFEC9GYAK5.out
varbendi@varbendi-VirtualBox: ~/Desktop/kodok$ ./CKFEC9GYAK5.out
total 152
drwxr-xr-x 2 varbendi varbendi 4096 márc 12 19:11 .
drwxr-xr-x 3 varbendi varbendi 4096 márc 12 18:41 ..
-rw-rw-r-- 1 varbendi varbendi 561 márc 12 18:52 child.c
-rwxr-xr-x 1 varbendi varbendi 16824 márc 12 19:02 child.out
-rw-rw-r-- 1 varbendi varbendi 527 márc 12 18:40 CKFEC9GYAK1.c
-rwxr-xr-x 1 varbendi varbendi 16880 márc 12 18:42 CKFEC9GYAK1.out
-rw-rw-r-- 1 varbendi varbendi 229 márc 12 18:40 CKFEC9GYAK2.c
-rwxr-xr-x 1 varbendi varbendi 16840 márc 12 18:40 CKFEC9GYAK2.out
-rw-rw-r-- 1 varbendi varbendi 16880 márc 12 19:08 CKFEC9GYAK4.c
-rwxr-xr-x 1 varbendi varbendi 16968 márc 12 19:08 CKFEC9GYAK4.out
-rw-rw-r-- 1 varbendi varbendi 846 márc 12 18:41 CKFEC9GYAK5.c
-rwxr-xr-x 1 varbendi varbendi 16968 márc 12 19:11 CKFEC9GYAK5.out
-rw-rw-r-- 1 varbendi varbendi 220 márc 12 18:41 parent.c
-rwxr-xr-x 1 varbendi varbendi 16690 márc 12 19:02 parent.out
varbendi@varbendi-VirtualBox: ~/Desktop/kodok$ total 152
drwxr-xr-x 2 varbendi varbendi 4096 márc 12 19:11 .
drwxr-xr-x 3 varbendi varbendi 4096 márc 12 18:41 ..
-rw-rw-r-- 1 varbendi varbendi 561 márc 12 18:52 child.c
-rwxr-xr-x 1 varbendi varbendi 16824 márc 12 19:02 child.out
-rw-rw-r-- 1 varbendi varbendi 527 márc 12 18:40 CKFEC9GYAK1.c
-rwxr-xr-x 1 varbendi varbendi 16880 márc 12 18:42 CKFEC9GYAK1.out
-rw-rw-r-- 1 varbendi varbendi 229 márc 12 18:40 CKFEC9GYAK2.c
-rwxr-xr-x 1 varbendi varbendi 16840 márc 12 18:40 CKFEC9GYAK2.out
-rw-rw-r-- 1 varbendi varbendi 16880 márc 12 19:08 CKFEC9GYAK4.c
-rwxr-xr-x 1 varbendi varbendi 16968 márc 12 19:08 CKFEC9GYAK4.out
-rw-rw-r-- 1 varbendi varbendi 846 márc 12 18:41 CKFEC9GYAK5.c
-rwxr-xr-x 1 varbendi varbendi 16968 márc 12 19:11 CKFEC9GYAK5.out
-rw-rw-r-- 1 varbendi varbendi 220 márc 12 18:41 parent.c
-rwxr-xr-x 1 varbendi varbendi 16690 márc 12 19:02 parent.out
varbendi@varbendi-VirtualBox: ~/Desktop/kodok$
```