

# Operációs rendszerek BSc

5. Gyak.

2022. 03. 09.

**Készítette:**

Sikora Dávid Ádám Bsc

Mérnökinformatika

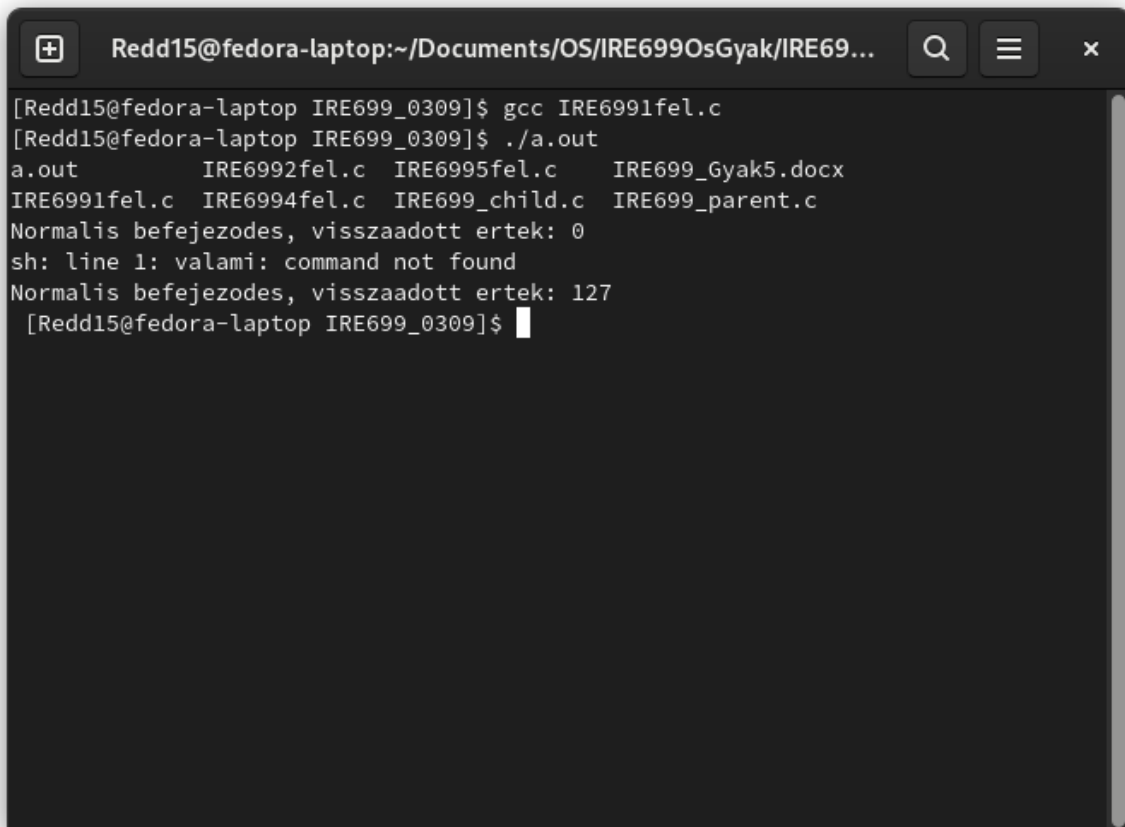
IRE699

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.



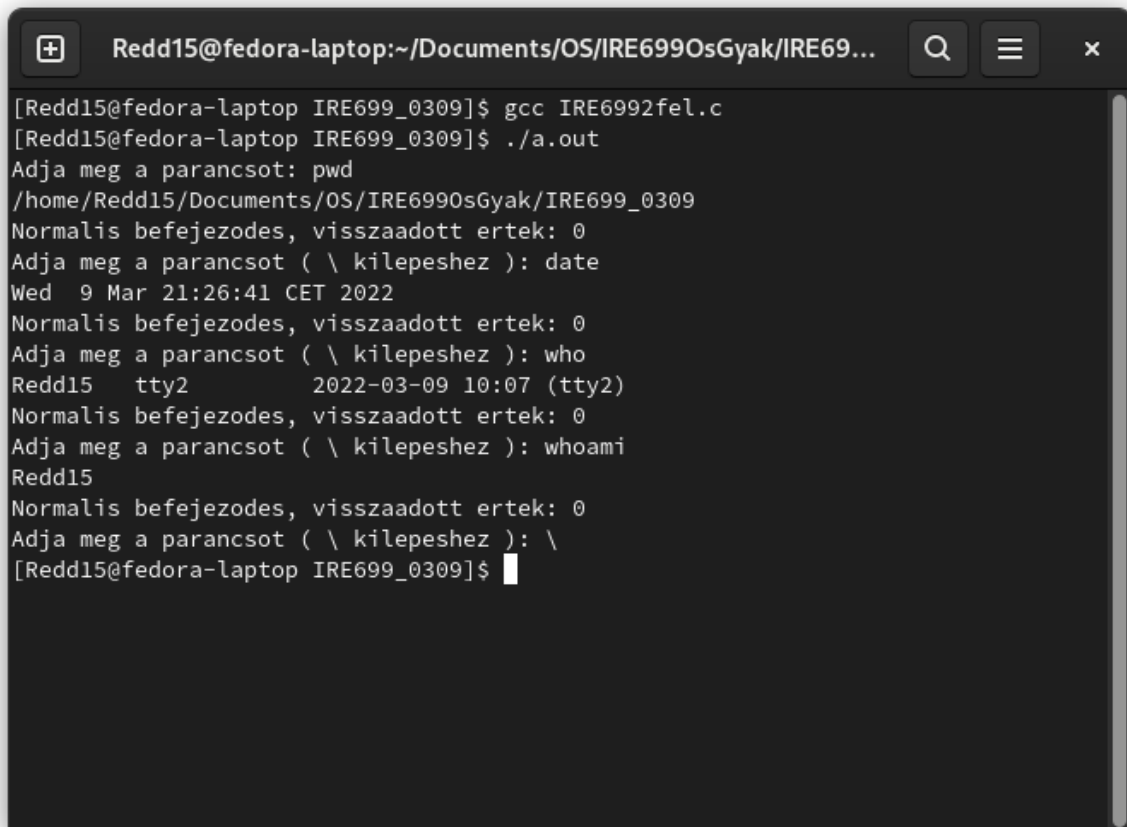
```
Redd15@fedora-laptop:~/Documents/OS/IRE699OsGyak/IRE69...  
[Redd15@fedora-laptop IRE699_0309]$ gcc IRE6991fel.c  
[Redd15@fedora-laptop IRE699_0309]$ ./a.out  
a.out          IRE6992fel.c  IRE6995fel.c    IRE699_Gyak5.docx  
IRE6991fel.c  IRE6994fel.c  IRE699_child.c  IRE699_parent.c  
Normalis befejezodes, visszaadott ertek: 0  
sh: line 1: valami: command not found  
Normalis befejezodes, visszaadott ertek: 127  
[Redd15@fedora-laptop IRE699_0309]$
```

A lefordított kód meghívása után látjuk hogy a valós parancs után (`ls`) a visszatérési értékünk 0 volt, míg a nem valós parancs után 127-es visszatérési értéket látunk.

## 2.Feladat

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.

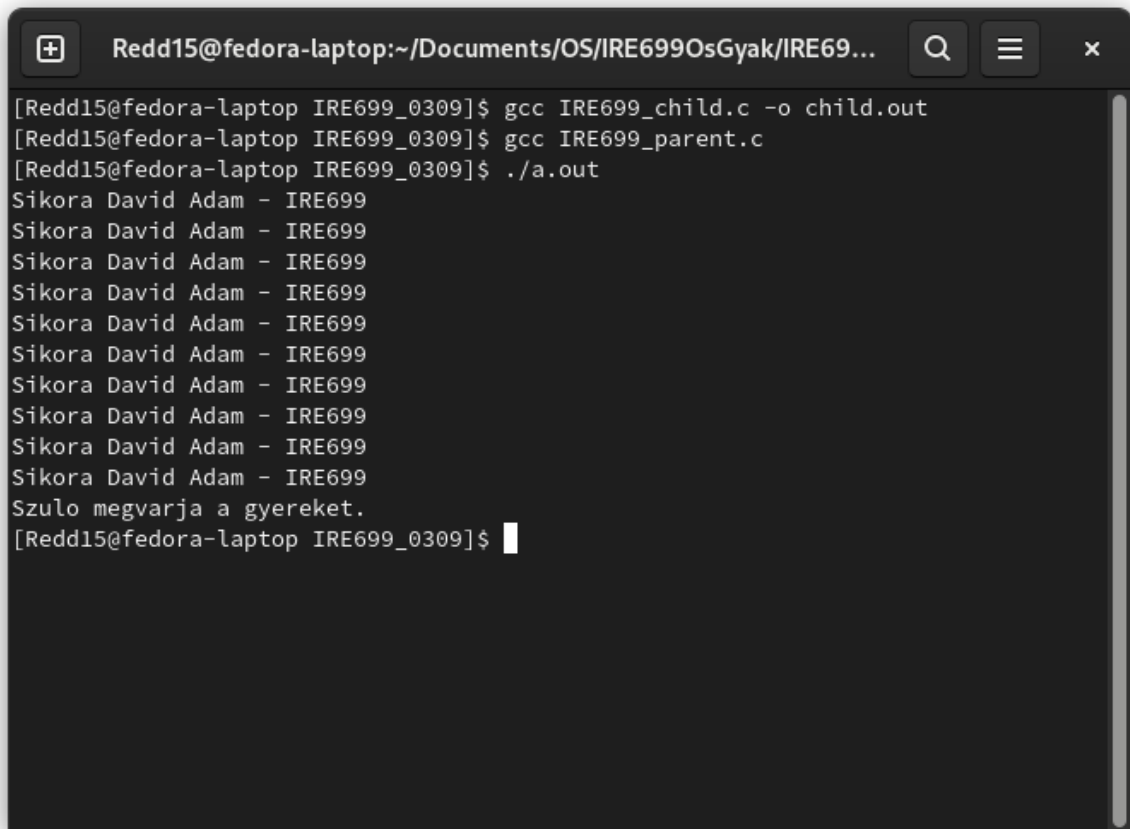


```
Redd15@fedora-laptop:~/Documents/OS/IRE699OsGyak/IRE69...
[Redd15@fedora-laptop IRE699_0309]$ gcc IRE6992fel.c
[Redd15@fedora-laptop IRE699_0309]$ ./a.out
Adja meg a parancsot: pwd
/home/Redd15/Documents/OS/IRE699OsGyak/IRE699_0309
Normalis befejezodes, visszaadott ertek: 0
Adja meg a parancsot ( \ kilepeshez ): date
Wed 9 Mar 21:26:41 CET 2022
Normalis befejezodes, visszaadott ertek: 0
Adja meg a parancsot ( \ kilepeshez ): who
Redd15  tty2          2022-03-09 10:07 (tty2)
Normalis befejezodes, visszaadott ertek: 0
Adja meg a parancsot ( \ kilepeshez ): whoami
Redd15
Normalis befejezodes, visszaadott ertek: 0
Adja meg a parancsot ( \ kilepeshez ): \
[Redd15@fedora-laptop IRE699_0309]$
```

A system() rendszerhívással bármilyen kódot le lehet futtatni amit a bemenetébe string formátumba adunk meg, így egy string változóba elmentett bármilyen bevitt kódot a programunk le tud futtatni.

### 3.Feladat

Készítsen egy XY\_parent.c és a XY\_child.c programokat. A XY\_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 egyegy mondattal  
A fordítás/futtatás után készítsen egy képernyőképet (minden parancs esetén) és illessze be a dokumentumba.



```
Redd15@fedora-laptop:~/Documents/OS/IRE699OsGyak/IRE69...  
[Redd15@fedora-laptop IRE699_0309]$ gcc IRE699_child.c -o child.out  
[Redd15@fedora-laptop IRE699_0309]$ gcc IRE699_parent.c  
[Redd15@fedora-laptop IRE699_0309]$ ./a.out  
Sikora David Adam - IRE699  
Sikora David Adam - IRE699  
Sikora David Adam - IRE699  
Sikora David Adam - IRE699  
Sikora David Adam - IRE699  
Sikora David Adam - IRE699  
Sikora David Adam - IRE699  
Sikora David Adam - IRE699  
Sikora David Adam - IRE699  
Sikora David Adam - IRE699  
Sikora David Adam - IRE699  
Szulo megvarja a gyereket.  
[Redd15@fedora-laptop IRE699_0309]$
```

A gyermek processbe egy for-ciklust helyeztünk el. Ezt a processt a szülő processben hívtuk meg a fork() paranccsal. Ezek után a szülő process megvárta a gyermek processt amit a waitpid() paranccsal tudtunk elérni.

## 4.Feladat

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.

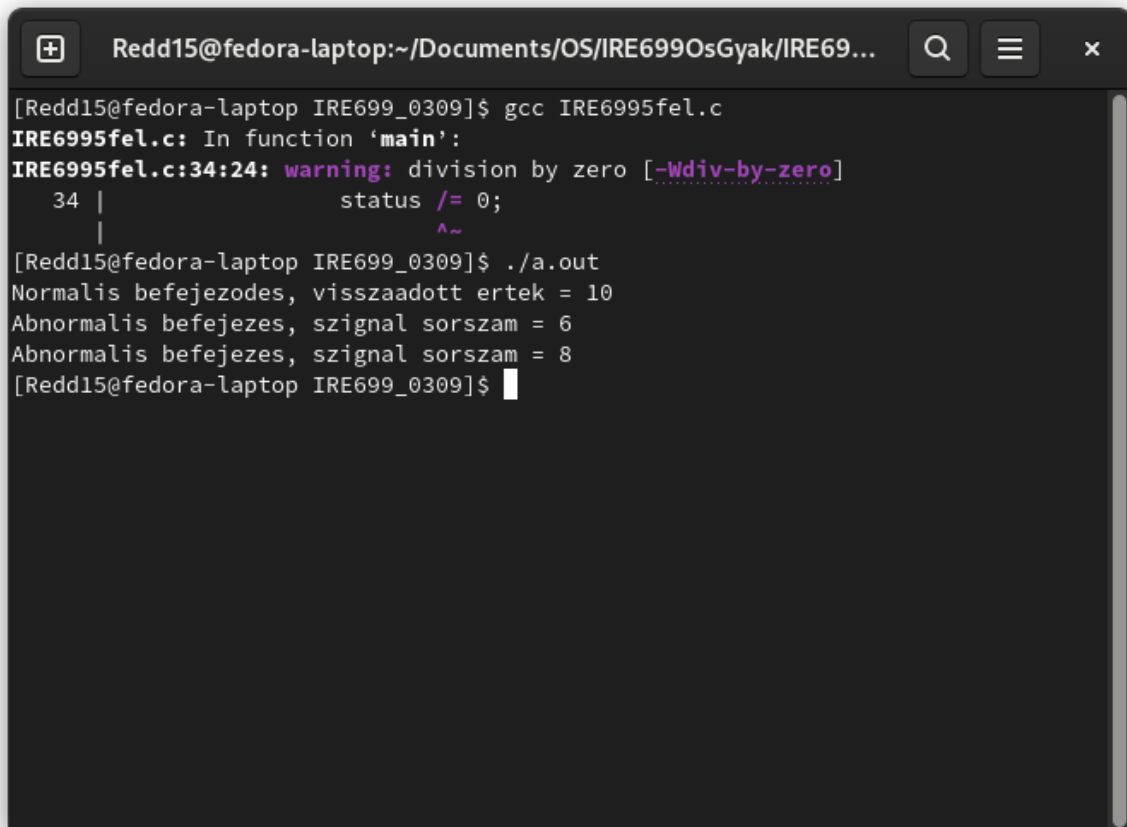
```
Redd15@fedora-laptop:~/Documents/OS/IRE699OsGyak/IRE69...  
[Redd15@fedora-laptop IRE699_0309]$ gcc IRE6994fel.c  
[Redd15@fedora-laptop IRE699_0309]$ ./a.out  
Wed 9 Mar 21:29:30 CET 2022  
A szulo processz megvarta a gyermeket.  
[Redd15@fedora-laptop IRE699_0309]$
```

Ebben a feladatban szintén a `fork()` paranccsal indítottunk el egy gyermek processt, ami most nem egy másik C program volt, hanem az `execlp()` paranccsal meghívott `date` parancs.

## 5.Feladat

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.



```
[Redd15@fedora-laptop IRE699_0309]$ gcc IRE6995fel.c
IRE6995fel.c: In function 'main':
IRE6995fel.c:34:24: warning: division by zero [-Wdiv-by-zero]
   34 |         status /= 0;
      |                ^~
[Redd15@fedora-laptop IRE699_0309]$ ./a.out
Normalis befejezodes, visszaadott ertekek = 10
Abnormalis befejezes, szignal sorszam = 6
Abnormalis befejezes, szignal sorszam = 8
[Redd15@fedora-laptop IRE699_0309]$
```

Itt először azt látjuk, hogy a 0-val való osztás hibát adott vissza fordításkor, de a program ettől függetlenül lefutott.

Futáskor látjuk hogy az első esetben a processt az `exit(10)` paranccsal állítottuk le, így egy normális befejezési értéket látunk, mely 10-zel tért vissza.

A következő két process már hibával tért vissza, az első process a 6-os kóddal, ami `abort`-ot jelent, a második pedig 8-assal, ami pedig aritmetikai hibát jelent.