

Operációs Rendszerek Bsc

3.gyak.

2021.03.03.

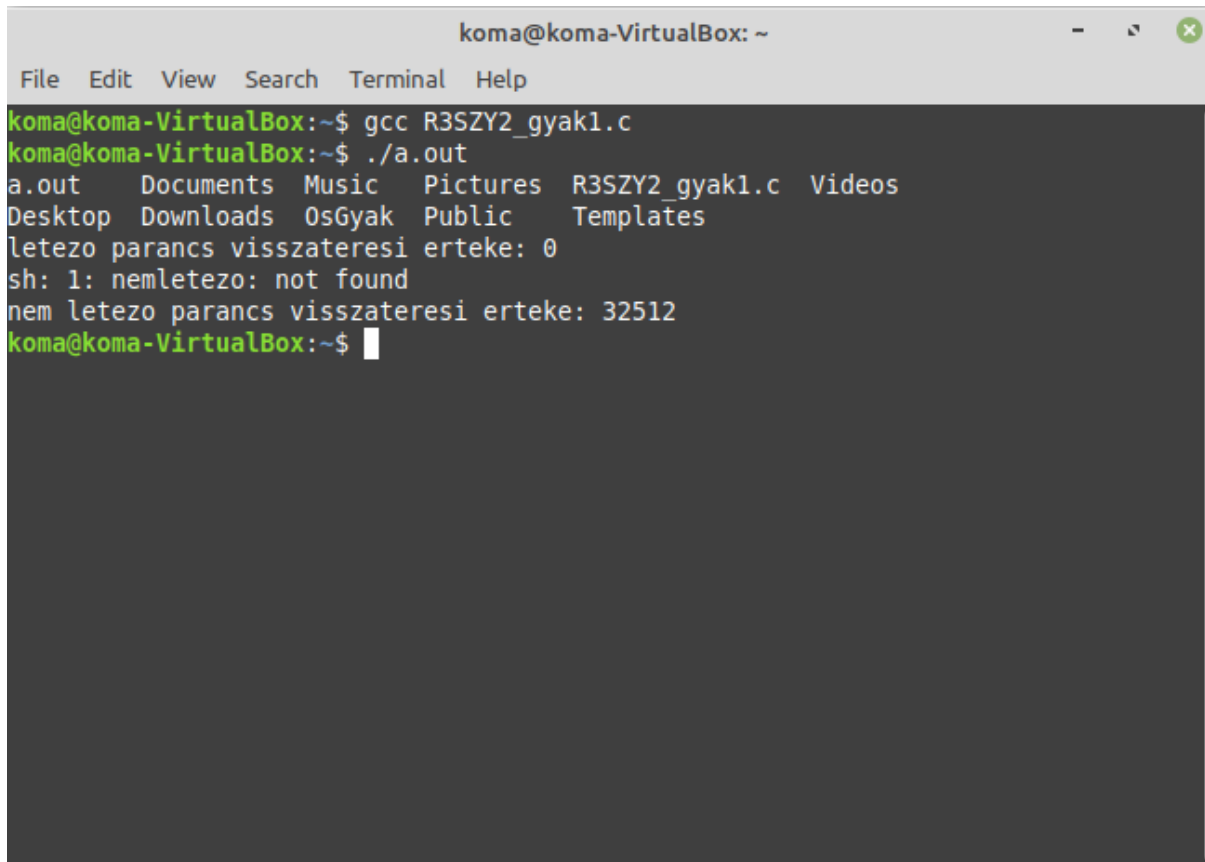
Készítette:

Horváth Ákos Bsc

Programtervező informatikus
szak

R3SZY2

1.feladat- A `system()` rendszerhívással hajtson végre létező és nem létező parancsot, és vizsgálja a visszatérési értéket! Mentés: `neptunkodgyak1.c`



```
koma@koma-VirtualBox: ~  
File Edit View Search Terminal Help  
koma@koma-VirtualBox:~$ gcc R3SZY2_gyak1.c  
koma@koma-VirtualBox:~$ ./a.out  
a.out Documents Music Pictures R3SZY2_gyak1.c Videos  
Desktop Downloads OsGyak Public Templates  
letezo parancs visszateresi erteke: 0  
sh: 1: nemletezo: not found  
nem letezo parancs visszateresi erteke: 32512  
koma@koma-VirtualBox:~$
```

Mint sok C függvénynél, a `system()` függvénynél is a 0 jelenti a normális visszatérést, és bármely nem 0 érték nem megfelelő lefutást jelent.

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-\)

3.feladat-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 (5-ször) (pl. a hallgató neve és a neptunkód)!

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!

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

Ezekről a feladatokról nem csináltam képernyőképet, de a forráskód megtalálható a github repositorymban.