

# Operációs rendszerek BSc

## 5.Gyak.

2022.03.08.

**Készítette:**

Stremler László Bsc

Programtervező Informatikus

AQYO8L

**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.**

```
AQYO8L1fel
AQYO8L1fel.cbp bin main.c obj
KORTE!
KORTE!
KORTE!
sh: 1: color: not found
sh: 1: Nincs: not found

Process returned 0 (0x0)   execution time : 0,005 s
Press ENTER to continue.
```

Az első sorban az adott mappa tartalma íródik ki a parancssorba, majd háromszor kiírja a “KORTE!” szót. Linuxon a color parancsot mivel nem ismeri a rendszer, így arra hibát dob, továbbá a “Nincs ilyen parancs!” kiírásra is hibát dob.

**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**

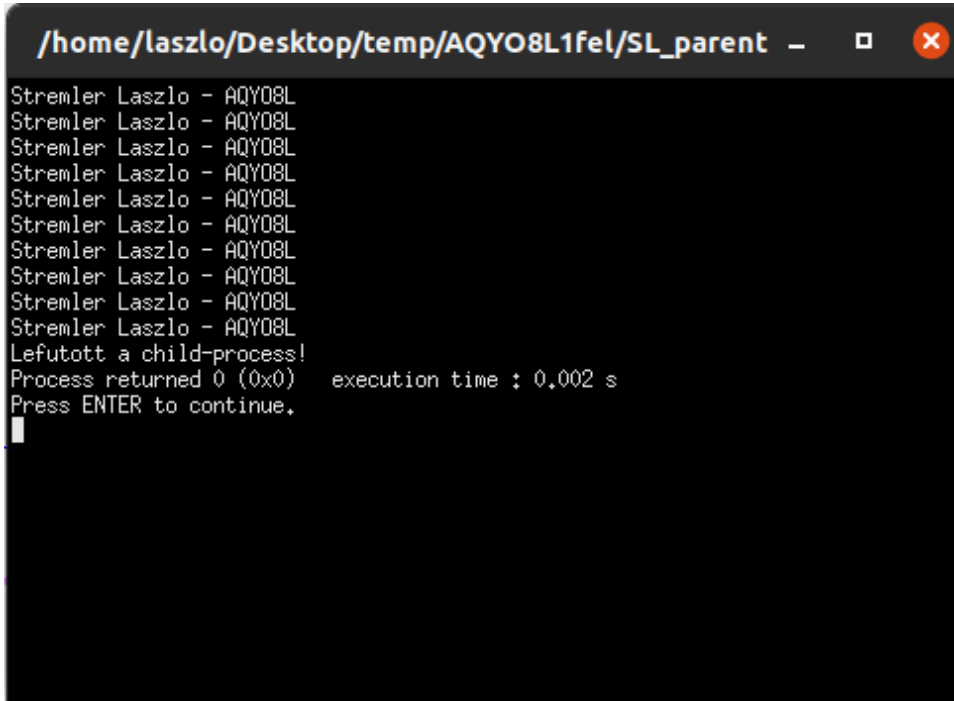
**A fordítás/futtatás után készítsen egy képernyőképet (minden parancs esetén) és illessze be a dokumentumba.**

```
/home/laszlo/Desktop/temp/AQYO8L1fel/AQYO8L2...
Kerek egy parancsot: date
2022. márc. 8., kedd, 13:42:08 CET
Kerek egy parancsot: pwd
/home/laszlo/Desktop/temp/AQYO8L1fel
Kerek egy parancsot: who
laszlo  :0      2022-03-08 14:18 (:0)
Kerek egy parancsot: etc
sh: 1: etc: not found
Kerek egy parancsot: ^C
Process returned -1 (0xFFFFFFFF)   execution time : 34,689 s
Press ENTER to continue.
```

Bekértem a felsorolt parancsokat: A date az aktuális dátumot adta vissza, a pwd az aktuális mappát, a who pedig az aktuális felhasználót. CTRL+C-vel lehet kilépni a programból.

**3. 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.**



```
/home/laszlo/Desktop/temp/AQYO8L1fel/SL_parent - □ ×
Stremler Laszlo - AQYO8L
Stremler Laszlo - AQYO8L
Stremler Laszlo - AQYO8L
Stremler Laszlo - AQYO8L
Stremler Laszlo - AQYO8L
Stremler Laszlo - AQYO8L
Stremler Laszlo - AQYO8L
Stremler Laszlo - AQYO8L
Stremler Laszlo - AQYO8L
Stremler Laszlo - AQYO8L
Lefutott a child-process!
Process returned 0 (0x0)   execution time : 0.002 s
Press ENTER to continue.
█
```

Elindítottam az SL\_Parent nevű programot, ami a fork-kal meghívta gyerek processzként az SL\_Child nevű programot. A kimeneten először a child process fut le, majd folytatja futását a fő processz.

**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.**

```
/home/laszlo/Desktop/temp/AQYO8L1fel/AQYO8L4... - □ ×
./parent
./AQYO8L2fel.c
./SL_parent.o
./AQYO8L2fel.o
./parent.o
./obj
./obj/Debug
./obj/Debug/main.o
./AQYO8L4fel.c
./AQYO8L1fel.layout
./bin
./bin/Debug
./bin/Debug/AQYO8L1fel
./SL_parent
./XY_parent.o
./AQYO8L4fel
./parent.c
./AQYO8L1fel.c
./XY_child
Lefutott a gyerek processz!

Process returned 0 (0x0)   execution time : 0,003 s
Press ENTER to continue.
```

A find parancsot futtatja le a child-process-ben. Ezután a program kiírja, hogy lefutott a gyerekfolyamat.

**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.**

```
/home/laszlo/Desktop/temp/AQYO8L1fel/AQYO8L5... - □ ×
exit status was 53
Floating point exception (core dumped)
exit status was 136

Process returned 0 (0x0)   execution time : 0.279 s
Press ENTER to continue.
█
```

Kettő gyerek process-t indít el a főprogram: Az egyik az 53-as return kóddal tér vissza, míg a másik megpróbál nullával osztani egy egész számot. A normál visszatérési értéket a fő program adja, miután megvárja míg minden gyerek processz lefutott.