

# **Operációs rendszerek BSc**

**5. Gyak.**

**2022. 03. 09.**

**Készítette:**

Palencsár Enikő Bsc

Mérnökinformatikus

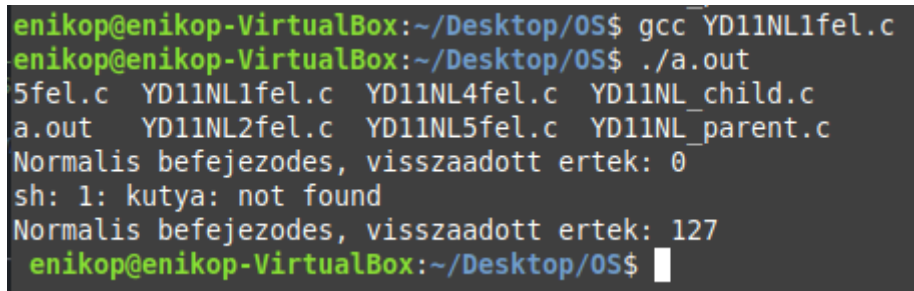
YD11NL

**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.



```
enikop@enikop-VirtualBox:~/Desktop/OS$ gcc YD11NL1fel.c
enikop@enikop-VirtualBox:~/Desktop/OS$ ./a.out
5fel.c YD11NL1fel.c YD11NL4fel.c YD11NL_child.c
a.out YD11NL2fel.c YD11NL5fel.c YD11NL_parent.c
Normalis befejezodes, visszaadott ertek: 0
sh: 1: kutya: not found
Normalis befejezodes, visszaadott ertek: 127
enikop@enikop-VirtualBox:~/Desktop/OS$
```

A `system()` rendszerhívással az `ls` parancs lefut, a szabványos kimenetre kiíródik a jegyzék tartalma, majd a `system()` visszatér 1-gyel. Ha nem létező parancsot írok be, a `system()` hívás ugyanúgy normális befejeződésű lesz, a visszaadott érték azonban 127, azaz „command not found”.

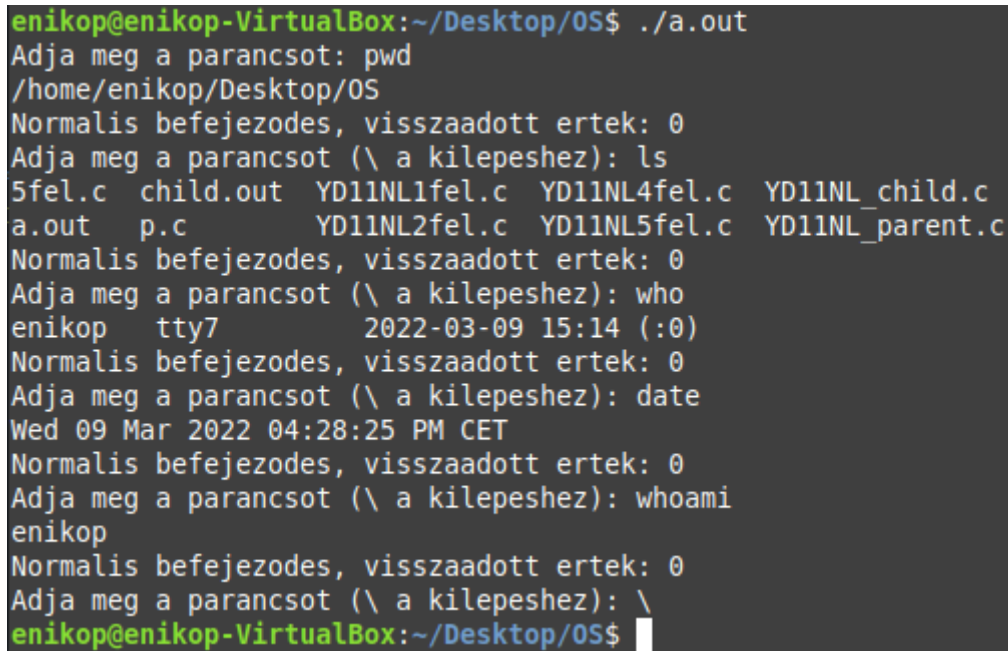
A státuszt a `system()` visszatérési értékeként kapjuk meg, ezen kell aztán a `WIFEXITED()` makrófüggvénnyel ellenőrizni, hogy normális volt-e a terminálódás. Ha igen, akkor a `WEXITSTATUS()`-szal megkaphatjuk magát a visszaadott értéket.

Mentés: neptunkod1fel.c

## 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.



```
enikop@enikop-VirtualBox:~/Desktop/OS$ ./a.out
Adja meg a parancsot: pwd
/home/enikop/Desktop/OS
Normalis befejezodes, visszaadott ertek: 0
Adja meg a parancsot (\ a kilepeshez): ls
5fel.c  child.out  YD11NL1fel.c  YD11NL4fel.c  YD11NL_child.c
a.out   p.c        YD11NL2fel.c  YD11NL5fel.c  YD11NL_parent.c
Normalis befejezodes, visszaadott ertek: 0
Adja meg a parancsot (\ a kilepeshez): who
enikop  tty7      2022-03-09 15:14 (:0)
Normalis befejezodes, visszaadott ertek: 0
Adja meg a parancsot (\ a kilepeshez): date
Wed 09 Mar 2022 04:28:25 PM CET
Normalis befejezodes, visszaadott ertek: 0
Adja meg a parancsot (\ a kilepeshez): whoami
enikop
Normalis befejezodes, visszaadott ertek: 0
Adja meg a parancsot (\ a kilepeshez): \
enikop@enikop-VirtualBox:~/Desktop/OS$
```

A system() rendszerhívást használva a megadott utasítások lefutnak, eredményük kiíródik a szabványos kimenetre. Ezután a \ billentyű lenyomásával lehet kilépni a programból.

Mentés: neptunkod2fel.c

### 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

```
enikop@enikop-VirtualBox:~/Desktop/OS$ gcc YD11NL_child.c -o child.out
enikop@enikop-VirtualBox:~/Desktop/OS$ gcc YD11NL_parent.c
enikop@enikop-VirtualBox:~/Desktop/OS$ ./a.out
Palencsar Eniko - YD11NL
Palencsar Eniko - YD11NL
Palencsar Eniko - YD11NL
Palencsar Eniko - YD11NL
Palencsar Eniko - YD11NL
Palencsar Eniko - YD11NL
Palencsar Eniko - YD11NL
Palencsar Eniko - YD11NL
Palencsar Eniko - YD11NL
Palencsar Eniko - YD11NL
Szulo megvarta a gyereket.
```

A gyermek processz egy egyszerű for ciklust tartalmaz, míg a szülő processzben fork() rendszerhívással gyermeket hozunk létre, ebben a gyermekben execls() hívással elindítjuk a child.c lefordított, .out kiterjesztésű kódját, amely kiírja a szöveget tízszer. Azt, hogy a szülő megvárja a gyerek-processzt, a waitpid() hívással érjük el (ennek sikerességét a waitpid() hívás után egy kiíratással ellenőriztem).

Mentés: XY\_parent.c, ill. XY\_child.

## 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.

```
enikop@enikop-VirtualBox:~/Desktop/05$ gcc YD11NL4fel.c
enikop@enikop-VirtualBox:~/Desktop/05$ ./a.out
Wed 09 Mar 2022 03:35:46 PM CET
A szulo processz megvarta a gyermeket.
enikop@enikop-VirtualBox:~/Desktop/05$
```

A feladatot a 3. feladattal analóg módon oldottam meg, ezúttal viszont az `execlp()`-vel egyszerű parancsot (`date`) futtattam, nem általam létrehozott kódot.

Mentés: `neptunkod4fel.c`

## 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.

```
enikop@enikop-VirtualBox:~/Desktop/OS$ gcc YD11NL5fel.c
YD11NL5fel.c: In function 'main':
YD11NL5fel.c:34:10: warning: division by zero [-Wdiv-by-zero]
   34 |     status /= 0;
      |           ^~
enikop@enikop-VirtualBox:~/Desktop/OS$ ./a.out
Normalis befejezodes, visszaadott ertek = 10
Abnormalis befejezes, szignal sorszam = 6
Abnormalis befejezes, szignal sorszam = 8
enikop@enikop-VirtualBox:~/Desktop/OS$
```

A nullával osztás a fordításkor warningot generált ugyan, de a program lefutott.

Az `exit(0)` normális befejeződést jelent, a visszatérési érték a hívás utáni zárójelben megadott szám, jelen esetben ez 10 volt. Az `abort()` és a nullával való osztás abnormalis befejezést eredményeztek, a gyermek processzek 6-ot és 8-at szignáloztak a szülőnek. (6=SIGABRT; 8=SIGFPE(aritmetikai hiba))

A szignálok megjelenítése a `WIFSIGNALED()` (volt-e szignál) és a `WTERMSIG()` (mi volt a szignál) makrók segítségével történt.

Mentés: neptunkod5fel.c