

Operációs rendszerek BSc

5. gyak.

2022. 03. 09.

Készítette:

Takács Bálint Bsc

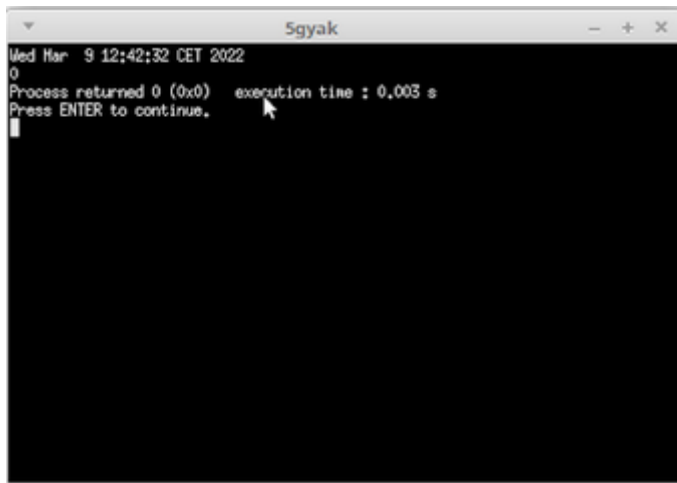
Programtervező informatikus

P2GNFT

Miskolc, 2022

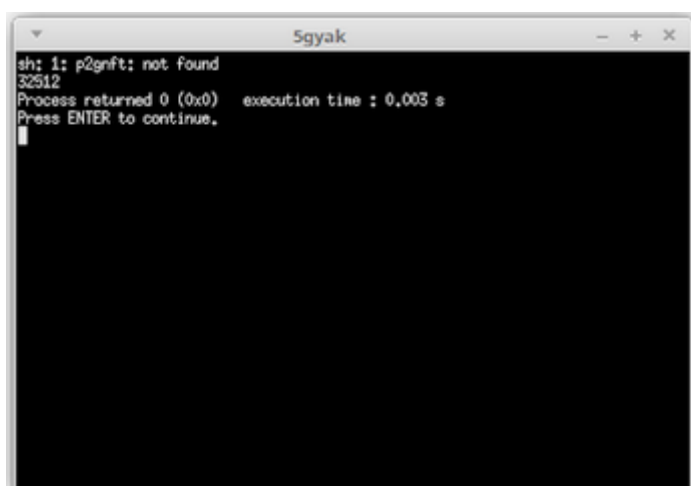
1. A `system()` rendszerhívással hajtson végre létező és nem létező parancsot, és vizsgálja a visszatérési értéket, magyarázza egy-egy mondattal

Az alábbi képen egy „`system()`” rendszerhívás látható létező paranccsal. Abban az esetben, ha a „`system()`” számára egy létező parancsot adunk meg paraméterként, akkor a megadott parancs végrehajtódik. Jelen esetben a „`date`” parancsot adtam meg paraméternek.



```
Wed Mar 9 12:42:32 CET 2022
0
Process returned 0 (0x0)   execution time : 0.003 s
Press ENTER to continue.
```

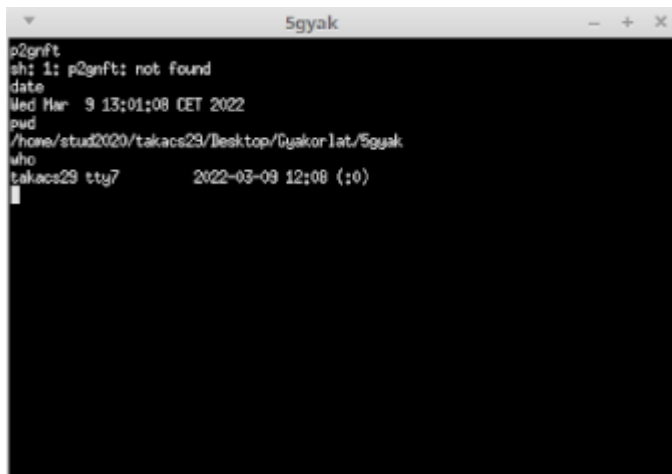
Az alábbi képen egy „`system()`” rendszerhívás látható egy nem létező paranccsal. Abban az esetben, ha a „`system()`” számára egy nem létező parancsot adunk meg paraméterként, akkor a konzolra hibaüzenetet ír ki, hogy a paraméterben megadott parancs nem ismert, ezért a rendszer nem is tudja értelmezni. Jelen esetben a neptunkódomat (`p2gnft`) adtam meg paraméternek.



```
sh: 1: p2gnft: not found
32512
Process returned 0 (0x0)   execution time : 0.003 s
Press ENTER to continue.
```

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

Ezt a feladatot szintén egy „system()” rendszerhívással valósítottam meg, a konzolon a billentyűzetről bekérek egy szót amely bekerül majd a „system()” parancs paraméterébe. Létező parancs esetén értelmezi a parancsot és végrehajtja azt. Nem létező parancs esetén, pedig hiba üzenetet dob a program.



```
p2gnft
sh: 1: p2gnft: not found
date
Wed Mar 9 13:01:08 CET 2022
pwd
/home/stud2020/takacs29/Desktop/Gyakorlat/Sgyak
who
takacs29 tty7 2022-03-09 12:08 (:0)
```

3. 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 (10-ször) (pl. a hallgató neve és a neptunkód)! - magyarázza egy-egy mondattal A fordítás/futtatás után készítsen egy képernyőképet (minden parancs esetén) és illessze be a dokumentumba.

A gyermek program tartalmazza azt a parancsot amely 10-szer kiírja az adott szöveget. A szülő programban ennek a gyermek programnak a meghívása van és ennek a meghívásnak az eredménye lesz, hogy végrehajtnak a gyermek programban levő utasítások.

```
child
Takacs Balint P2GNFT
Takacs Balint P2GNFT
Takacs Balint P2GNFT
Takacs Balint P2GNFT
Takacs Balint P2GNFT
Takacs Balint P2GNFT
Takacs Balint P2GNFT
Takacs Balint P2GNFT
Takacs Balint P2GNFT
Takacs Balint P2GNFT
Process returned 0 (0x0)   execution time : 0.002 s
Press ENTER to continue.
```

4. A `fork()` rendszerhívással hozzon létre egy gyerek processzt-t és abban hívjon meg egy `exec` családbeli rendszerhívást (pl. `execvp`). A szülő várja meg a gyerek futását! - magyarázza egy-egy mondattal.

Az alábbi programban az `execpl` rendszerhívást alkalmaztam amelynek a segítségével a felhasználót kérdeztem le, úgy hogy a rendszerhívás paramétereként a „who” parancsot adtam meg.

```

P2GNFTgyak4
takacsb tty7 2022-03-18 20:05 (:0)
Process returned 0 (0x0) execution time : 0.026 s
Press ENTER to continue.

```

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

```
P2GNFTgyak5
exit
abort
osztas 0-val
0
136
134
Process returned 0 (0x0)   execution time : 0.464 s
Press ENTER to continue.
```