

Operációs rendszerek BSc

5.gyak.

2021. 03. 10.

Készítette:
Kovács Krisztián
Programtervező informatikus
WIQPM2

Miskolc, 2021

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

```
kovacs k tty7 2021-03-10 13:28 (:0)

sh: 1: l: not found

Process returned 0 (0x0) execution time : 0.022 s
Press ENTER to continue.
```

Ha egy létező parancsot adunk ki, akkor az végrehajtódik és visszatérési értéke a parancsnak a visszatérési értéke, ha viszont egy nem létező parancsot adunk, akkor hibaüzenetet kapunk, mi szerint ez a parancs nem létezik.

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-\\) Mentés: *neptunkodgyak2.c*

```
Parancs: date
2021. márc. 10., szerda, 14:32:24 EET

Process returned 0 (0x0) execution time : 1.933 s
Press ENTER to continue.
```

A program bekéri a parancsot, és végrehajtja, mint az első feladatban. (Csak később vettem észre, hogy ciklusba kell tenni, amelyet egy kilépési feltételig futtatunk, de már nem javítottam.)

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 (5-ször) (pl. a hallgató neve és a neptunkód)! Mentés: *parent.c*, ill. *child.c*

```
/home/kovacs/Asztal/parent/parent
execl error: Permission denied
wait error: No child processes

Process returned 0 (0x0) execution time : 0.011 s
Press ENTER to continue.
```

Nem értem miért nincs engedélyezve, próbáltam megoldani, de nem tudtam, ezért van a késői feltöltés. (Többször nekifutottam)

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! Mentés: *neptunkodgyak4.c*
5. A `fork()` rendszerhívással hozzon létre gyerekeket, várja meg és vizsgálja a befejeződési állapotokat (gyerekekben: `exit`, `abort`, nullával való osztás)! Mentés: *neptunkodgyak5.c*

6. Adott a következő ütemezési feladat, ahol a RR ütemezési algoritmus használatával készítse el:
 Határozza meg a
- Ütemezze az adott időszelvény alapján az egyes processzek paramétereit (ms)!
 - A rendszerben lévő processzek végrehajtásának sorrendjét?
 - Ábrázolja Gantt diagram segítségével az aktív/várakozó processzek futásának menetét!

RR: 5ms	Round Robin				
	P1	P2	P3	P4	P5
Érkezés	0	1	3	9	12
CPU idő	3	8	2	20	5
Indulás	0	3, 10	8	13	18
Befejezés	3	8, 13	10	18, 38	23
Várakozás	0	2, 2	5	4, 5	6

$p_1 \rightarrow p_2(1.) \rightarrow p_3 \rightarrow p_2(2. \text{forduló}) \rightarrow p_4(1.) \rightarrow p_5 \rightarrow p_4(2. \text{forduló})$

