

Operációs rendszerek Bsc

5. Gyak.

2022. 03. 08.

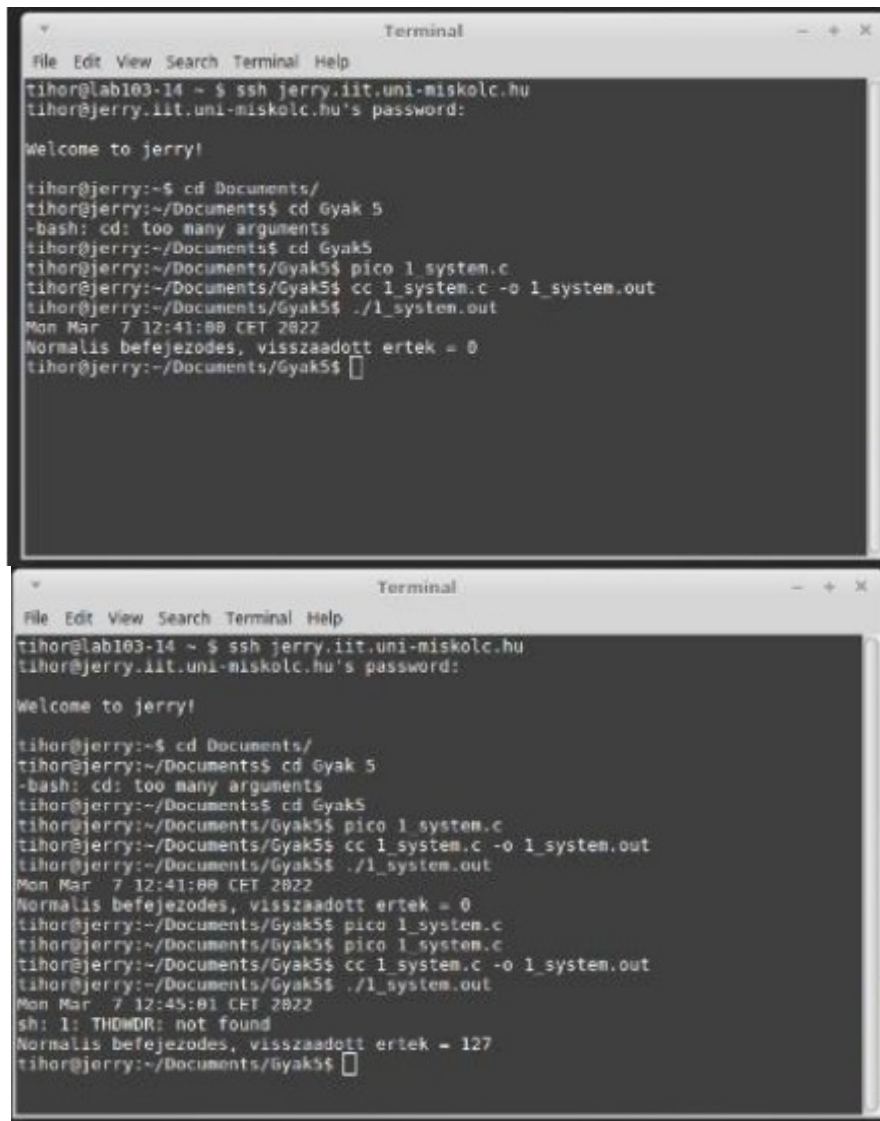
Készítette:

Flaskó Lilian Laura
Programtervező informatikus
GCNS8S

Miskolc, 2022.05.

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.

Megvalósítás: Két system() hívás van, ahol az elsőben a date parancsot hajtjuk végre, másodikban egy nem létező parancsot hajtunk végre. Az elsőnél normális volt a befejeződés, a másikon viszont nem, mivel nem létező parancs volt.



```
tihor@lab103-14 ~ $ ssh jerry.iit.uni-miskolc.hu
tihor@jerry.iit.uni-miskolc.hu's password:
Welcome to jerry!

tihor@jerry:~$ cd Documents/
tihor@jerry:~/Documents$ cd Gyak 5
-bash: cd: too many arguments
tihor@jerry:~/Documents$ cd Gyak5
tihor@jerry:~/Documents/Gyak5$ pico 1_system.c
tihor@jerry:~/Documents/Gyak5$ cc 1_system.c -o 1_system.out
tihor@jerry:~/Documents/Gyak5$ ./1_system.out
Mon Mar 7 12:41:00 CET 2022
Normalis befejezodes, visszaadott ertek = 0
tihor@jerry:~/Documents/Gyak5$

tihor@jerry:~$ cd Documents/
tihor@jerry:~/Documents$ cd Gyak 5
-bash: cd: too many arguments
tihor@jerry:~/Documents$ cd Gyak5
tihor@jerry:~/Documents/Gyak5$ pico 1_system.c
tihor@jerry:~/Documents/Gyak5$ cc 1_system.c -o 1_system.out
tihor@jerry:~/Documents/Gyak5$ ./1_system.out
Mon Mar 7 12:41:00 CET 2022
Normalis befejezodes, visszaadott ertek = 0
tihor@jerry:~/Documents/Gyak5$ pico 1_system.c
tihor@jerry:~/Documents/Gyak5$ pico 1_system.c
tihor@jerry:~/Documents/Gyak5$ cc 1_system.c -o 1_system.out
tihor@jerry:~/Documents/Gyak5$ ./1_system.out
Mon Mar 7 12:45:01 CET 2022
sh: 1: THQWDR: not found
Normalis befejezodes, visszaadott ertek = 127
tihor@jerry:~/Documents/Gyak5$
```

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.

Megvalósítás:

```
jerry.iit.uni-miskolc.hu - PuTTY
tihor@jerry:~/Documents/Gyak5$ ./2_parancs.out
Irjon be egy parancsot!
valami
valamitihor@jerry:~/Documents/Gyak5$ ./2_parancs.out
Irjon be egy parancsot!
Q
Qtihor@jerry:~/Documents/Gyak5$ nano 2_parancs.c
tihor@jerry:~/Documents/Gyak5$ ./2_parancs.out
Irjon be egy parancsot!
ls
lstihor@jerry:~/Documents/Gyak5$ nano 2_parancs.c
tihor@jerry:~/Documents/Gyak5$ s
-bash: s: command not found
tihor@jerry:~/Documents/Gyak5$ ls
1_system.out 2_parancs.out 3_fork.out THDWDRlifel.c THDWDRgyakl
2_parancs.c 3_fork.c 5_fork2.c THDWDRlifel.out
tihor@jerry:~/Documents/Gyak5$ nano 2_parancs.c
tihor@jerry:~/Documents/Gyak5$ ls
1_system.out 2_parancs.out 3_fork.out THDWDRlifel.c THDWDRgyakl
2_parancs.c 3_fork.c 5_fork2.c THDWDRlifel.out
tihor@jerry:~/Documents/Gyak5$ ./2_parancs.out
Irjon be egy parancsot!
ls
1_system.out 2_parancs.out 3_fork.out THDWDRlifel.c THDWDRgyakl
2_parancs.c 3_fork.c 5_fork2.c THDWDRlifel.out
tihor@jerry:~/Documents/Gyak5$
tihor@jerry:~/Documents/Gyak5$
tihor@jerry:~/Documents/Gyak5$
tihor@jerry:~/Documents/Gyak5$
```

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 (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.

Megvalósítás:

```
jerry.iit.uni-miskolc.hu - PuTTY
[declaration]
exit(0);
^~~~
3.c:16:2: warning: incompatible implicit declaration of built-in function 'exit'
3.c:16:2: note: include '<stdlib.h>' or provide a declaration of 'exit'
tihor@jerry:~/Documents/Gyak5$ nano 3.c
tihor@jerry:~/Documents/Gyak5$ cc 3.c -o 3.out
3.c: In function 'main':
3.c:11:13: warning: implicit declaration of function 'fork' [-Wimplicit-function
-declaration]
    if( (pid = fork())<0 )
               ^~~~
tihor@jerry:~/Documents/Gyak5$
tihor@jerry:~/Documents/Gyak5$ nano 3.c
tihor@jerry:~/Documents/Gyak5$ cc 3.c -o 3.out
tihor@jerry:~/Documents/Gyak5$ ./3.out
Szulo processz kGimeyenermteek
processz kimenete
tihor@jerry:~/Documents/Gyak5$ nano 3.c
tihor@jerry:~/Documents/Gyak5$ ./3.out
Szulo processz kimGenyeterere
mek processz kimenete
tihor@jerry:~/Documents/Gyak5$
```

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.

Megvalósítás

```
jerry.iit.uni-miskolc.hu - PuTTY
tihor@jerry:~/Documents/Gyak5$ nano THDWDR4fel.c
tihor@jerry:~/Documents/Gyak5$ ./THDWDR4fel.out
execlp error: No such file or directory
tihor@jerry:~/Documents/Gyak5$ nano THDWDR4fel.c
tihor@jerry:~/Documents/Gyak5$ ./THDWDR4fel.out
execlp error: No such file or directory
tihor@jerry:~/Documents/Gyak5$ nano THDWDR4fel.c
tihor@jerry:~/Documents/Gyak5$ ./THDWDR4fel.out
execlp error: No such file or directory
tihor@jerry:~/Documents/Gyak5$ ./THDWDR4fel.out
execlp error: No such file or directory
tihor@jerry:~/Documents/Gyak5$
```

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.

Megvalósítás

```
jerry.iit.uni-miskolc.hu - PuTTY
tihor@jerry:~/Documents/Gyak5$ ./THDWDR5fel.out
Normalis befejezodes,visszaadott ertek = 7
Abnormalis befejezes, a szignal sorszam = 6
Abnormalis befejezodes, a szignal sorszam = 8
tihor@jerry:~/Documents/Gyak5$
tihor@jerry:~/Documents/Gyak5$
tihor@jerry:~/Documents/Gyak5$
tihor@jerry:~/Documents/Gyak5$
```

6. feladat - Adott a következő ütemezési feladat, amit a FCFS, SJF és Round Robin (RR) ütemezési algoritmus használatával készítsen el (külön-külön táblázatba):

I. Határozza meg FCFS és SJF esetén

a.) A befejezési időt?

b.) A várakozási/átlagos várakozási időt?

c.) Ábrázolja Gantt diagram segítségével az aktív/várakozó processzek futásának menetét.

FCFS

FCFS	Érkezés	CPU idő
P1	0	3
P2	1	8
P3	3	2
P4	9	20
P5	12	5

FCFS	Érkezés	CPU idő	Indulás	Befejezés	Várakozás
P1	0	3	0	3	0
P2	1	8	3	11	2
P3	3	2	11	13	8
P4	9	20	13	33	4
P5	12	5	33	38	21

SJF

SJF	Érkezés	CPU idő
-----	---------	---------

SJF megoldás

SJF	Érkezés	CPU idő	Indulás	Befejezés	Várakozás	Legrövidebb
P1	0	3	5	8	5	2
P2	1	5	8	13	7	3
P3	3	2	3	5	0	1
P4	9	5	13	18	4	4
P5	12	5	18	23	6	5

II. Round Robin (RR) esetén a.)

Ütemezze az adott időszelket (5ms) alapján az egyes processzek (befejezési és várakozási/átlagos várakozási idő) paramétereit (ms)!

b.) A rendszerben lévő processzek végrehajtásának sorrendjét?

c.) Ábrázolja Gantt diagram segítségével az aktív/várakozó processzek futásának menetét!”

RR: 5ms	Érkezés	CPU idő
P1	0	3
P2	1	8
P3	3	2
P4	9	20
P5	12	5

RR megoldás

RR: 5ms	Érkezés	CPU idő	Indulás	Befejezés	Várakozás	Várakozó processz
P1	0	3	0	3	0	P2,P3
P2	1	8	3	8	2	P3
P3	3	2	8	10	5	P2,P4
P2	1	8	10	13	2	P4,P5
P4	9	20	13	18	4	P5
P5	12	5	18	23	6	P4
P4	9	20	23	28	5	-
P4	9	20	28	33	0	-
P4	9	20	33	38	0	-