

Operációs rendszerek BSc

2. Gyak.

2022. 03.18

Készítette:

Juhász Tibor

Mérnökinformatikus BSc. levelező

X7KWVG

1, Feladat leírás:

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.

Megoldás:

Program megírása:

```
X7KWVG_1fel.c x
#include <stdlib.h>
#include <stdio.h>

int main(int argc, char *argv[])
{
    int ret;

    printf("System call\n");

    ret = system("ls");

    printf("ret= %d\n", ret);

    return 0;
}
```

Fordítás és futtatás:

```
tibi@tibi-VirtualBox:~$ gcc -o X7KWVG_1fel X7KWVG_1fel.c
tibi@tibi-VirtualBox:~$ ./X7KWVG_1fel
System call
ls
ret= 0
tibi@tibi-VirtualBox:~$
```

A program lefutott, a megadott parancsot végrehajtotta. A visszatérési érték 0, mivel sikeresen lefutott a parancs.

Amennyiben hibás parancsot adunk meg:

```
*X7KWVG_1fel.c x
#include <stdlib.h>
#include <stdio.h>

int main(int argc, char *argv[])
{
    int ret;

    printf("System call\n");

    ret = system("Ez egy hibás parancs");

    printf("ret= %d\n", ret);

    return 0;
}
```

Fordítás és futtatás:

```
tibi@tibi-VirtualBox:~$ gcc -o X7KWVG_1fel X7KWVG_1fel.c
tibi@tibi-VirtualBox:~$ ./X7KWVG_1fel
System call
sh: 1: Ez: not found
ret= 32512
tibi@tibi-VirtualBox:~$
```

A visszatérési érték nem 0, mivel a parancs futtatása nem sikerült.

2, Feladat leírás:

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.

Megoldás:

Program megírása:

Tartalmaz egy végtelen ciklust, hogy a program fusson, ameddig nem akarunk kilépni belőle.



```
X7KWVG_2fel.c x X7KWVG_1fel.c x
#include <stdlib.h>
#include <stdio.h>

int main(int argc, char *argv[])
{
    char str[50];

    for(int i=0;;i++){
        printf("Kérem adja meg a %d. parancsot\n",i+1);
        scanf("%s",str);
        system(str);
    }

    printf("hello");
    return 0;
}
```

A kiírás után várja a következő parancsot, majd végrehajtja és újabb parancsot vár.

Fordítás és futtatás:

```
tibi@tibi-VirtualBox:~$ gcc -o X7KWVG_2fel X7KWVG_2fel.c
tibi@tibi-VirtualBox:~$ ./X7KWVG_2fel
Kérem adja meg a 1. parancsot
```

Parancsok megadása:

```
tibi@tibi-VirtualBox:~$ gcc -o X7KWVG_2fel X7KWVG_2fel.c
tibi@tibi-VirtualBox:~$ ./X7KWVG_2fel
Kérem adja meg a 1. parancsot
ps
  PID TTY          TIME CMD
 1859 pts/0    00:00:00 bash
 7345 pts/0    00:00:00 X7KWVG_2fel
 7347 pts/0    00:00:00 sh
 7348 pts/0    00:00:00 ps
Kérem adja meg a 2. parancsot
w
 18:59:02 up  4:30,  1 user,  load average: 0,14, 0,13, 0,12
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
tibi      tty7     :0               k20     46:50m 50.65s  0.46s cinnamon-session --session cinnamon2d
Kérem adja meg a 3. parancsot
```

Kilépés:

```
tibi@tibi-VirtualBox:~$ ./X7KWVG_2fel
Kérem adja meg a 1. parancsot
ps
  PID TTY          TIME CMD
 1859 pts/0    00:00:00 bash
 7345 pts/0    00:00:00 X7KWVG_2fel
 7347 pts/0    00:00:00 sh
 7348 pts/0    00:00:00 ps
Kérem adja meg a 2. parancsot
w
 18:59:02 up  4:30,  1 user,  load average: 0,14, 0,13, 0,12
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
tibi      tty7     :0               k20     46:50m 50.65s  0.46s cinnamon-session --session cinnamon2d
Kérem adja meg a 3. parancsot
^C
tibi@tibi-VirtualBox:~$ ^C
```

3, Feladat leírás:

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

Megoldás:

Programok megírása:

parent.c:

```
/*  
Forras:  
Dr. Vadasz Denes: Operacios rendszerek  
*/  
#include <stdio.h>  
#include <unistd.h>  
#include <stdlib.h>  
#include <sys/types.h>  
#include <sys/wait.h>  
  
int main(int argc, char const *argv[])  
{  
    int pid = fork();  
    char *args[] = {NULL};  
  
    if(pid == 0) {  
        execv("./child",args);  
    } else {  
        wait(NULL);  
        printf("A gyermek processz lefutott.\n");  
    }  
  
    return 0;  
}
```

child.c:

```
/*  
Forras:  
Dr. Vadasz Denes: Operacios rendszerek  
*/  
  
#include <stdio.h>  
#include <unistd.h>  
#include <stdlib.h>  
  
int main(int argc, char const *argv[])  
{  
    int i;  
    for(i=0; i<10; i++){  
        printf("Juhász Tibor X7KWVG\n");  
    }  
    return 0;  
}
```

Fordítás és futtatás:

```
tibi@tibi-VirtualBox:~$ gcc -o parent parent.c
tibi@tibi-VirtualBox:~$ gcc -o child child.c
tibi@tibi-VirtualBox:~$ ./parent
Juhász Tibor X7KWVG
Juhász Tibor X7KWVG
Juhász Tibor X7KWVG
Juhász Tibor X7KWVG
Juhász Tibor X7KWVG
Juhász Tibor X7KWVG
Juhász Tibor X7KWVG
Juhász Tibor X7KWVG
Juhász Tibor X7KWVG
Juhász Tibor X7KWVG
A gyermek processz lefutott.
tibi@tibi-VirtualBox:~$
```

A parent.c program elindítja a child.c programot, és megvárja amíg lefut, utána pedig kiírja, hogy lefutott a gyermek processz.

A child.c program 10-szer kiírja a nevemet és neptun kódomat, utána pedig leáll.

3, Feladat leírás:

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.

Megoldás:

Program megírása.

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>

int main(int argc, char const *argv[])
{
    int pid = fork();
    char *args[] = {NULL};

    if(pid == 0) {
        system("exec ls -l");
    } else {
        wait(NULL);
        printf("A gyermek processz lefutott.\n");
    }

    return 0;
}
```

Fordítás és futtatás:

```
tibi@tibi-VirtualBox:~$ gcc -o X7KwVG_4fel X7KwVG_4fel.c
tibi@tibi-VirtualBox:~$ ./X7KwVG_4fel
Asztal
child
child.c
Dokumentumok
gyumolcs
Képek
Letöltések
'Névtelen mappa'
'Névtelen mappa 2'
Nyilvános
parent
parent.c
Sablonok
Videók
X7KwVG
X7KwVG_1fel
X7KwVG_1fel.c
X7KwVG_2fel
X7KwVG_2fel.c
X7KwVG_4fel
X7KwVG_4fel.c
Zenék
A gyermek processz lefutott.
tibi@tibi-VirtualBox:~$
```

A program létrehoz egy gyermek processzt, amely végrehajtja az `exec ls` parancsot. A szülő processz megvárja, amíg a gyermek processz fut, csak utána áll le.