

Operációs rendszerek

Ötödik gyakorlat

Készítette: Énekes Zoltán László

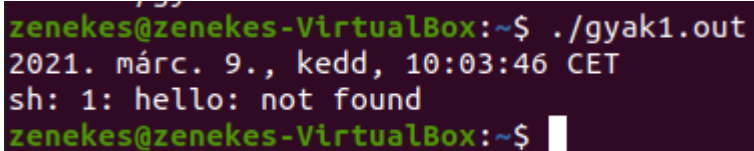
Linux OS – Rendszerhívások

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

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

```
int main(void) {
    system("date");
    // nem létező
    system("hello");

    return 0;
}
```



```
zenekes@zenekes-VirtualBox:~$ ./gyak1.out
2021. márc. 9., kedd, 10:03:46 CET
sh: 1: hello: not found
zenekes@zenekes-VirtualBox:~$
```

2. Írjon programot, amely billentyűzetről bekér Unix parancsokat és végrehajtja őket, majd kiírja a szabványos kimenetre.

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

```
#define BUFFER_SIZE 255
```

```
int main(void) {
    char buffer[BUFFER_SIZE];
    do
    {
        printf(">");
        fgets(buffer, BUFFER_SIZE, stdin);
        system(buffer);
    }
    while (1);

    return 0;
}
```

```

>date
2021. márc. 9., kedd, 10:16:11 CET
>ls
CIJA2K          CIJA2Kgyak22.c  Documents  gyak2.out  Public  Videos
CIJA2Kgyak1     CIJA2Kgyak2.c  Downloads  Music      snap
CIJA2Kgyak1.c   Desktop        gyak1.out  Pictures   Templates
>tree
.
├── CIJA2K
│   ├── bokor
│   │   ├── fa
│   │   ├── malna
│   │   │   └── description.txt
│   │   │       └── description.txt
│   │   └── mogyoro
│   └── eper
│       └── fa
│           ├── alma
│           ├── eper
│           ├── fa
│           │   ├── malna
│           │   ├── mogyoro
│           │   └── szilva
│           ├── gesztenye
│           ├── listing.txt
│           └── listing.txt,
├── CIJA2Kgyak1
├── CIJA2Kgyak1.c
├── CIJA2Kgyak22.c
└── CIJA2Kgyak2.c

```

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 írja szabványos kimenetre!

child.c

```
#include <stdio.h>
```

```
#include <unistd.h>
```

```

int main(void) {
    for (int i = 0; i < 5; i++)
    {
        printf("Énekes Zoltán CIJA2K\n");
        sleep(1);
    }

    return 0;
}

```

parent.c

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

```
int main(void) {
    int pid;

    if ((pid = fork()) < 0)
        perror("fork error");
    else if (pid == 0)
    {
        if (execl("./child", "child", (char *)NULL) < 0)
            perror("execl error");
    }
    if (waitpid(pid, NULL, 0) < 0)
        perror("wait error");

    return 0;
}
```

```
zenekes@zenekes-VirtualBox:~/gyakot$ cc child.c -o child.out
zenekes@zenekes-VirtualBox:~/gyakot$ ./child.out
Énekes Zoltán CIJA2K
Énekes Zoltán CIJA2K
Énekes Zoltán CIJA2K
Énekes Zoltán CIJA2K
Énekes Zoltán CIJA2K
```

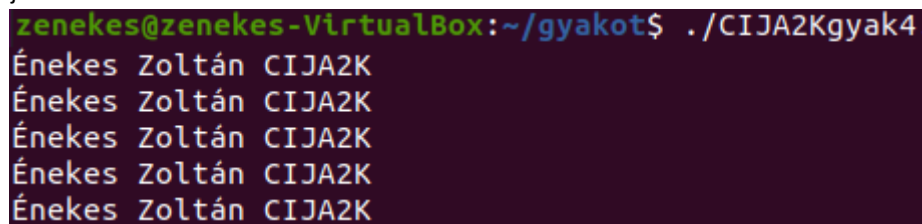
4. A `fork()` rendszerhívással hozzon létre egy gyerek processzt-t és abban hívjon meg egy `exec` családbeli rendszerhívást. A szülő várja meg a gyerek futását!

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

int main(void)
{
    int pid;
    int status;

    if ((pid = fork()) < 0)
        perror("fork hiba");
    else if (pid == 0) /* gyerek */
        if (execl("./child", "child", (char *)NULL) < 0)
            perror("execl error");
    if (wait(&status) != pid)
        perror("wait hiba"); /*szülő */

    return 0;
}
```



```
zenekes@zenekes-VirtualBox:~/gyakot$ ./CIJA2Kgyak4
Énekes Zoltán CIJA2K
Énekes Zoltán CIJA2K
Énekes Zoltán CIJA2K
Énekes Zoltán CIJA2K
Énekes Zoltán CIJA2K
```

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

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

```
int main(void)
{
    int pid;
    int status;

    if ((pid = fork()) < 0)
        perror("fork hiba");
    else if (pid == 0) /* gyerek */
        exit(7); /* befejeződik */
    if (wait(&status) != pid)
        perror("wait hiba"); /*szülő */
    if (WIFEXITED(status))
        printf("Normális befejeződés, visszaadott érték =%d\n", WEXITSTATUS(status));

    return 0;
}
```

```
zenekes@zenekes-VirtualBox:~/gyakot$ ./child.out
Normális befejeződés, visszaadott érték =7
```