

# JEGYZŐKÖNYV

Operációs rendszerek BSc

2021 tavasz féléves feladat

Készítette: **Czobor Patrik**

Neptunkód: **KLYSO1**

## Feladat leírása(3):

Írjon egy olyan C nyelvű programot ami létrehoz két gyermek processzt, ezek a processzek létrehoznak 3-3 további „unoka” processzt. Az unokák várnak 3 másodpercet, utána megszűnnek. A szülők megvárják a gyerek processzek lefutását és utána ők is megszűnnek.

## Feladat elkészítésének lépései:

Header állományok importálása:

*stdio* és *stdlib* a hagyományos  
függvények miatt kell

*unistd* a `fork()` függvény miatt

*sys/wait* a `wait()` függvény miatt

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

Használandó adattagok deklarálása/inicializálása:

```
pid_t gyerek1, gyerek2;    //A két unokákat készítő processz
int unokaProcessz = 3;     //Unoka processzek darabszáma
int sec = 3;               //Idő másodpercben amennyit várakozni fogunk
```

Elkészítjük az első gyermekprocesszt (*gyermek1*) és megnézzük, hogy a processz, ami eljutott ehhez a részhez gyermek vagy szülő processz-e:

## Gyermek:

Ha gyermek (tehát *gyerek1* == 0), akkor elkészítjük a három unokát egy for ciklusban, aztán várunk 3 másodpercet és kilépünk az unoka processzből.

```
gyerek1 = fork();           //Elkészítjük az első processzünket

if (gyerek1 == 0)
{
    //Gyerek1 programkód
    printf("-----Gyerek1-----\n");
    for(int i=0;i<unokaProcessz;i++)    //gyerek1 3db unokájának a létrehozása és várakozása
    {
        if(fork() == 0)
        {
            printf("[%d] pid %d [gyerek1]-tol pid %d\n",i+1,getpid(),getppid());
            sleep(sec); //Vár sec percet
            exit(0);
        }
    }
}
```

Ezután megívjuk a második gyermekprocesszt (*gyermek2*) aminek a gyerek programkódjának a része ugyanaz, mint *gyermek1* gyerek programkódja.

```
gyerek2 = fork();
if (gyerek2 == 0)
{
    //Gyerek2 programkód
    printf("-----Gyerek2-----\n");
    for(int i=0;i<unokaProcessz;i++)    //gyerek2 3db unokájának a létrehozása és várakozása
    {
        if(fork() == 0)
        {
            printf("[%d] pid %d [gyerek2]-tol pid %d\n",i+1,getpid(),getppid());
            sleep(sec);
            exit(0);
        }
    }
}
```

## Szülő:

*gyerek2* szülő kódrésztében megvárjuk a *gyerek2* által létrehozott unokák kilépését (*wait(NULL)*) és utána kilépünk a *gyerek2*-ből is.

```
else
{
    //Gyerek2 parent programkód
    for(int i=0;i<unokaProcessz;i++)
    {
        wait(NULL);    //Ez annyiszor fut le amennyi unokat kreáltunk a gyerek2-ben, ezzel vár a befejezésükre
        printf("\tVarakozas unokakra gyerek2 %dx \n",i+1);
    }
    printf("--Gyerek2 exit--\n");
    exit(0);    //Gyerek2 bezárása
}
```

*gyerek1* szülő kódrésztében ugyanúgy megvárjuk a *gyerek1* által létrehozott unokák kilépését, mint a *gyerek2*-ben, utána megvárjuk a *gyerek2* kilépését és aztán a *gyerek1*-et is termináljuk.

```
else
{
    //Gyerek1 parent programkód
    for(int i=0;i<unokaProcessz;i++)
    {
        wait(NULL);    //Ez annyiszor fut le amennyi unokat kreáltunk a gyerek2-ben, ezzel vár a befejezésükre
        printf("\tVarakozas unokakra gyerek1 %dx \n",i+1);
    }
    printf("Varakozas gyerek2-re\n");
    wait(NULL);    //megvárja a gyerek2 kilépését
    printf("--Gyerek1 exit--\n");
    exit(0);    //Gyerek1 kilépése
}
```

Ezután az eredeti szülőbe jutunk, ahol megvárjuk a *gyerek1* kilépését, és utána kilépünk a programból.

```
//Parent
wait(NULL);    //Megvárja a gyerek1 kilépését
printf("\t-----Return-----\n");    //Ezzel jelzi hogy vége minden folyamatnak és kilép
return 0;
```

## A futtatás eredménye:

```
jerry.iit.uni-miskolc.hu - PuTTY
czobor@jerry:~/Documents/OS$ ./a.out
-----Gyerek1-----
[1] pid 3009 [gyerek1]-tol pid 3007
[2] pid 3010 [gyerek1]-tol pid 3007
[3] pid 3011 [gyerek1]-tol pid 3007
-----Gyerek2-----
[1] pid 3013 [gyerek2]-tol pid 3012
[2] pid 3014 [gyerek2]-tol pid 3012
[3] pid 3016 [gyerek2]-tol pid 3012
    Varakozas unokakra gyerek2 1x
    Varakozas unokakra gyerek2 2x
    Varakozas unokakra gyerek2 3x
--Gyerek2 exit--
    Varakozas unokakra gyerek1 1x
    Varakozas unokakra gyerek1 2x
    Varakozas unokakra gyerek1 3x
Varakozas gyerek2-re
--Gyerek1 exit--
    -----Return-----
czobor@jerry:~/Documents/OS$
```