

# JEGYZŐKÖNYV

Operációs rendszerek BSc

2022. tavasz féléves feladat

Készítette: **Kiss Bence**

Neptunkód: **BYO2P7**

## A feladat leírása:

3. Írjon C nyelvű programot, ami:  
létrehoz két gyermekprocesszt  
ezek a gyermekprocesszek létrehoznak 3-3 további gyereket  
ezek az unokák várakoznak néhány másodpercet és szűnjenek meg  
a szülők várják meg a gyerekek befejeződését és csak utána szűnjenek meg.

## A feladat elkészítésének lépései:

1. Szülő és két gyermekprocessz létrehozása
2. Három unokaprocessz létrehozása a gyermekprocesszeken belül
3. Unokák várakozása megírása `sleep()`-el
4. A szülők gyermekeinek megvárásának megírása `wait()`-el

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

int main(int argc, char* argv[]) {
    int parent, child1, child2, grandchild1, grandchild2, grandchild3,
    grandchild4;

    child1 = fork(); //gyermek1 létrejön

    if (child1 == 0) { //ha gyermek1
        //child1 kódja
        grandchild1 = fork(); //gyermek1 = unoka1 | unoka2 létrejön

        if (grandchild1 == 0) { //ha unoka2
```

```

        sleep(2); //várakozik 2 másodpercet
    }

    else { //ha unoka1
        //unoka1 kódja

        grandchild2 = fork(); //gyermek1 = unoka2 és unoka3 létrejön
        if (grandchild2 == 0) { //ha unoka3
            //unoka3 kódja
            sleep(2); //várakozik 2 másodpercet
        }

        else { //ha unoka2
            //unoka2 kódja
            sleep(2); //várakozik 2 másodpercet
        }

        sleep(2); //várakozik 2 másodpercet
    }

    if (grandchild1 != 0) {
        wait(NULL);
    }
}

else { //ha szülő
    child2 = fork(); //gyermek2 létrejön

    if (child2 == 0) { //ha gyermek2
        //child2 kódja

```

```
    grandchild3 = fork();

    if (grandchild3 == 0) {
        sleep(2);
    }

    else {
        grandchild4 = fork();
        if (grandchild4 == 0) {
            sleep(2);
        }

        else {
            sleep(2);
        }
        sleep(2);
    }

    if (grandchild3 != 0) {
        wait(NULL);
    }
}

else {
    if (child1 != 0) {
        wait(NULL);
    }
}
}
```

}

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
[1] + Done
benice@linuxmint:~$ "/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm} @c*/tmp/Microsoft.MIEngine.In-uloarjbe.ekv" 1>/tmp/Microsoft.MIEngine.Out-rvhy0rbf.rn2"
```

8. Adott négy processz (A, B, C, D) a rendszerbe, induláskor a  $p\_cpu$  értéke  $A=0$ ,  $B=6$ ,  $C=0$ ,  $D=0$ . A rendszerben a  $P\_USER = 60$

Induláskor a  $p\_usrpri$   $A=60$ ,  $B=65$ ,  $C=60$  és  $D=60$ .

Ha egy processz megkapja a CPU-t a quantum-ában végig használja (azaz 1 quantum-ban), a  $p\_cpu$  növekmény értéke 70. Mind a négy processznél a  $p\_nice$  érték 0.

Határozza meg öt quantum-ban hogyan változnak a prioritások és a  $p\_cpu$ , melyik processz, milyen sorrendben kap CPU-t.

Igazolja az ütemező algoritmus leírásával, képlettel és számítással az eredményeket.

1. Megadott értékek leírása
2. Képletek leírása
3. Értékek kiszámolása a képletekkel és futó processzek sorrendjének meghatározása

Quantum	Process A		Process B		Process C		Process D		<p>P_USER: 60</p> <p>p_cpu növekmény: 70</p> <p>p_nice: 0</p>
	p-uszrl	p-cpu	p-uszrl	p-cpu	p-uszrl	p-cpu	p-uszrl	p-cpu	
	60	0	65	6	60	0	60	0	
1.	70	...							<p>const1 = 2</p> <p>const2 = 2</p> <p>const3 = 2</p>
	77	35	60	3	60	0	60	0	
2.			...	73					<p>p_cpu = p_cpu / const1</p> <p>p-uszrl = P_USER + p_cpu / const2 + const3 * p_nice</p>
	68	17	78	36	60	0	60	0	
3.									
	64	8	69	18	77	35	60	0	
4.									<p>...</p> <p>70</p>
	62	4	64	9	68	17	77	35	
5.									
		...							
		74							

  

Számítások: (egész értékek)	Process A		Process B		Process C		Process D	
	p-uszrl	p-cpu	p-uszrl	p-cpu	p-uszrl	p-cpu	p-uszrl	p-cpu
1. Quantum	60	0	65	6	60	0	60	0
2. Quantum	$60 + 35 / 2 + 2 * 0 = 77$	$70 / 2 = 35$	$60 + 1 / 2 + 2 * 0 = 60$	$6 / 2 = 3$	$60 + 0 / 2 + 2 * 0 = 60$	$0 / 2 = 0$	$60 + 0 / 2 + 2 * 0 = 60$	$0 / 2 = 0$
3. Quantum	$60 + 17 / 2 + 2 * 0 = 68$	$35 / 2 = 17$	$60 + 36 / 2 + 2 * 0 = 78$	$73 / 2 = 36$	$60 + 0 / 2 + 2 * 0 = 60$	$0 / 2 = 0$	$60 + 0 / 2 + 2 * 0 = 60$	$0 / 2 = 0$
4. Quantum	$60 + 8 / 2 + 2 * 0 = 64$	$17 / 2 = 8$	$60 + 18 / 2 + 2 * 0 = 69$	$36 / 2 = 18$	$60 + 35 / 2 + 2 * 0 = 77$	$70 / 2 = 35$	$60 + 0 / 2 + 2 * 0 = 60$	$0 / 2 = 0$
5. Quantum	$60 + 4 / 2 + 2 * 0 = 62$	$8 / 2 = 4$	$60 + 9 / 2 + 2 * 0 = 64$	$18 / 2 = 9$	$60 + 17 / 2 + 2 * 0 = 68$	$35 / 2 = 17$	$60 + 35 / 2 + 2 * 0 = 77$	$70 / 2 = 35$