

# **Operációs rendszerek BSc**

**9. Gyak.**

2022. 04. 06.

**Készítette:**

Pázmán András Bsc

Mérnökinformatikus

H2Z4X3

**Miskolc, 2022**



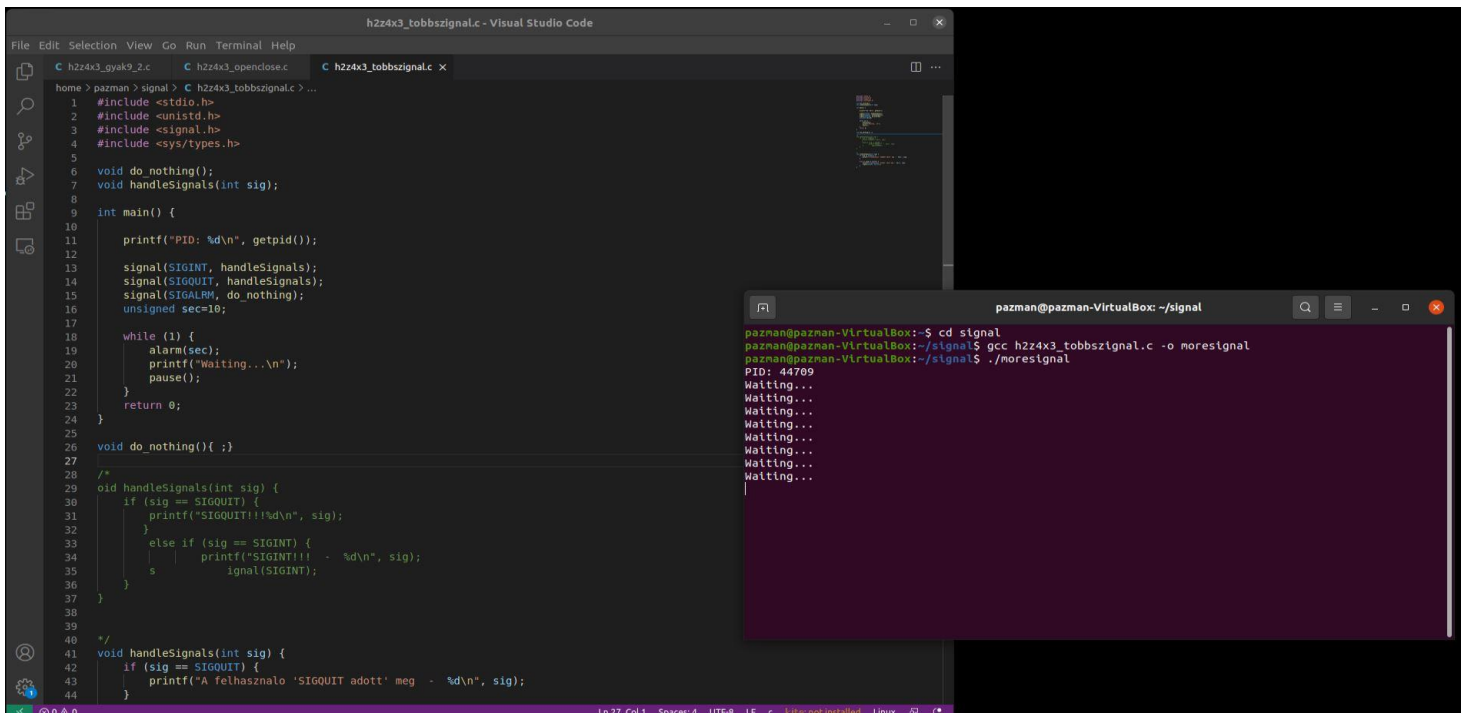
2. Készítse el a következő feladatot, melyben egy szignálkezelő több szignált is tud kezelni:

a.) Készítsen egy szignál kezelőt (handleSignals), amely a SIGINT (CTRL + C) vagy SIGQUIT (CTRL + \) jelek fogására vagy kezelésére képes.

b.) Ha a felhasználó SIGQUIT jelet generál (akár kill paranccsal, akár billentyűzetről a CTRL + \) a kezelő egyszerűen kiírja az üzenetet visszatérési értékét – a konzolra.

c.) Ha a felhasználó először generálja a SIGINT jelet (akár kill paranccsal, akár billentyűzetről a CTRL + C), akkor a jelet úgy módosítja, hogy a következő alkalommal alapértelmezett műveletet hajtson végre (a SIG\_DFL) – kiírás a konzolra. d.) Ha a felhasználó másodszor generálja a SIGINT jelet, akkor végrehajt egy alapértelmezett műveletet, amely a program befejezése - kiírás a konzolra.

Mentés: neptunkod\_tobbszignal.c



The image shows a Visual Studio Code editor window with a C program named `h2z4x3_tobbszignal.c` and a terminal window showing its execution.

**Visual Studio Code Editor:**

```
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <signal.h>
4 #include <sys/types.h>
5
6 void do_nothing();
7 void handleSignals(int sig);
8
9 int main() {
10     printf("PID: %d\n", getpid());
11
12     signal(SIGINT, handleSignals);
13     signal(SIGQUIT, handleSignals);
14     signal(SIGALRM, do_nothing);
15     unsigned sec=10;
16
17     while (1) {
18         alarm(sec);
19         printf("Waiting...\n");
20         pause();
21     }
22     return 0;
23 }
24
25 void do_nothing(){ ;}
26
27 /*
28 void handleSignals(int sig) {
29     if (sig == SIGQUIT) {
30         printf("SIGQUIT!!!%d\n", sig);
31     }
32     else if (sig == SIGINT) {
33         printf("SIGINT!!! - %d\n", sig);
34         signal(SIGINT);
35     }
36 }
37 */
38
39 void handleSignals(int sig) {
40     if (sig == SIGQUIT) {
41         printf("A felhasznalo 'SIGQUIT adott' meg - %d\n", sig);
42     }
43 }
```

**Terminal Window:**

```
pazman@pazman-VirtualBox: ~/signal
pazman@pazman-VirtualBox:~/signal$ gcc h2z4x3_tobbszignal.c -o moresignal
pazman@pazman-VirtualBox:~/signal$ ./moresignal
PID: 44709
Waiting...
Waiting...
Waiting...
Waiting...
Waiting...
Waiting...
Waiting...
```

3. Adott a következő ütemezési feladat, amit a FCFS, SJF és Round Robin (RR: 4 ms) ütemezési algoritmus alapján határozza meg következő **teljesítmény értékeket, metrikákat** (külön-külön táblázatba): **Külön táblázatba számolja a teljesítmény értékeket!**

CPU kihasználtság: számolni kell a **cs: 0,1(ms)** és **sch: 0,1 (ms)** értékkel is.

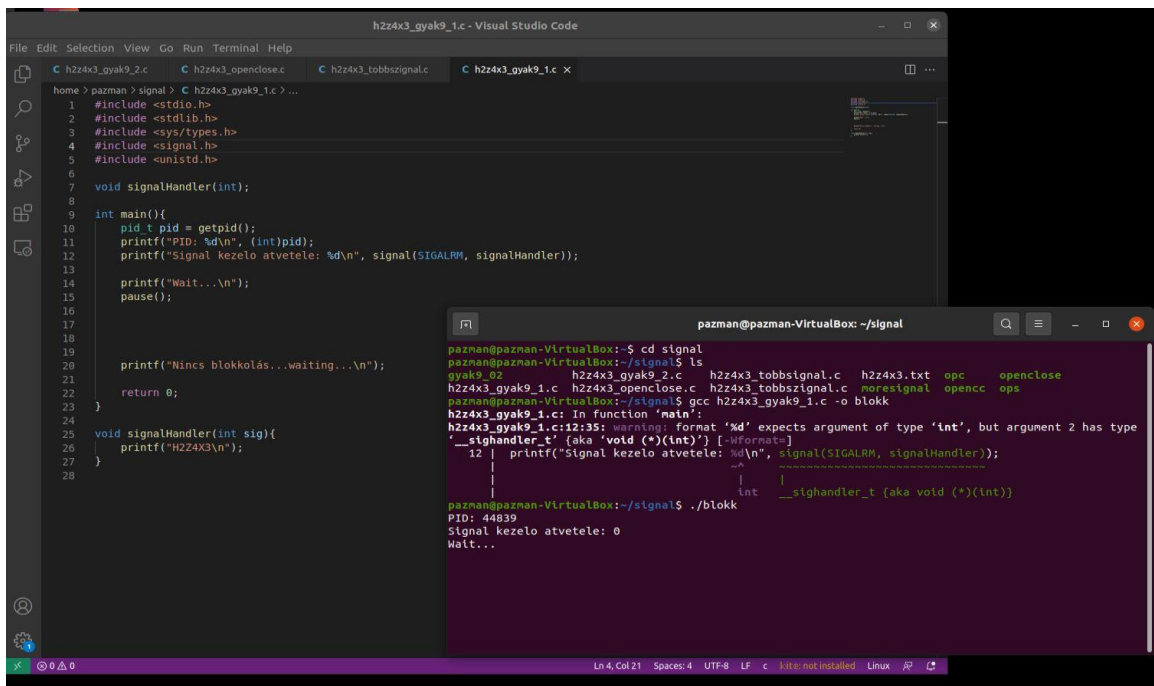
FCFS	P1	P2	P3	P4	Algoritmus neve	
Érkezés	0	0	2	5	Cpu kihasználtság	98,90%
CPU idő	24	3	6	3	Körülfordulási idő	28,25
Indulás	0	24	27	33	Várakozási idők átlaga	19,25
Befejezés	24	27	33	36	Válaszidő átlaga	19,25
Várakozás	0	24	25	28		
Kör.ford	24	27	31	31		
Válasz	0	24	25	28		

SJF	P1	P2	P3	P4	Algoritmus neve	
Érkezés	0	0	2	5	Cpu kihasználtság	98,90%
CPU idő	24	3	6	3	Körülfordulási idő	13,25
Indulás	12	0	3	9	Várakozási idők átlaga	4,25
Befejezés	36	3	9	12	Válaszidő átlaga	4,25
Várakozás	12	0	1	4		
Kör.ford	36	3	7	7		
Válasz	12	0	1	4		

RR	P1	P2	P3	P4	Algoritmus neve	
Érkezés	0,4, 15	0	2,11	5	Cpu kihasználtság	93,70%
CPU idő	24,20,16	3	6,2	3	Körülfordulási idő	18,5
Indulás	0,11,20	4	7,18	15	Várakozási idők átlaga	9,5
Befejezés	0,7,5	7	11,2	18	Válaszidő átlaga	4,75
Várakozás	0	4	5,7	10		
Kör.ford	36	7	18	13		

## Gyakorló feladatok - szignálkezelés

2. Írjon C nyelvű programot, amelyik kill() seg.-vel SIGALRM-et küld egy argumentumként megadott PID-u processznek, egy másik futó program a SIGALRM-hez rendeljen egy fv.-t amely kiírja pl. neptunkodot, továbbá pause() fv.-el blokkolódjon, majd kibillenés után jelezze, hogy kibillent és terminálódjon. Mentés. neptunkod\_gyak9\_1.c

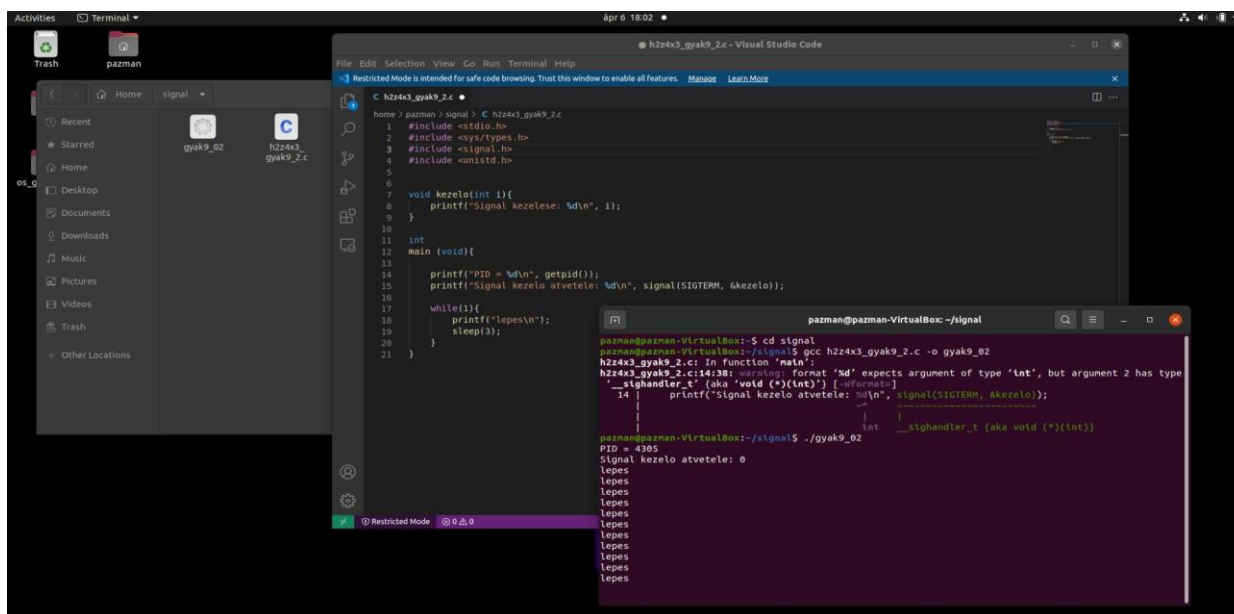


```
h2z4x3_gyak9_1.c - Visual Studio Code
File Edit Selection View Go Run Terminal Help
C h2z4x3_gyak9_2.c C h2z4x3_openclose.c C h2z4x3_tobbsignal.c C h2z4x3_gyak9_1.c X

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/types.h>
4 #include <signal.h>
5 #include <unistd.h>
6
7 void signalHandler(int);
8
9 int main(){
10     pid_t pid = getpid();
11     printf("PID: %d\n", (int)pid);
12     printf("Signal kezelo atvetele: %d\n", signal(SIGALRM, signalHandler));
13
14     printf("Wait...\n");
15     pause();
16
17
18     printf("Nincs blokkolas...waiting...\n");
19
20     return 0;
21 }
22
23 void signalHandler(int sig){
24     printf("H2Z4X3\n");
25 }
26
27
28
```

```
pazman@pazman-VirtualBox: ~/signal
pazman@pazman-VirtualBox:~$ cd signal
pazman@pazman-VirtualBox:~/signal$ ls
gyak9_02 h2z4x3_gyak9_2.c h2z4x3_tobbsignal.c h2z4x3.txt opc openclose
h2z4x3_gyak9_1.c h2z4x3_openclose.c h2z4x3_tobbsignal.c moresignal opencc ops
pazman@pazman-VirtualBox:~/signal$ gcc h2z4x3_gyak9_1.c -o blokk
h2z4x3_gyak9_1.c:12:35: warning: format '%d' expects argument of type 'int', but argument 2 has type
'__sighandler_t' (aka 'void (*)(int)') [-Wformat=]
12 | printf("Signal kezelo atvetele: %d\n", signal(SIGALRM, signalHandler));
    |                                ^~
    |                                |
    |                                int      __sighandler_t (aka void (*)(int))
pazman@pazman-VirtualBox:~/signal$ ./blokk
PID: 44839
Signal kezelo atvetele: 0
Wait...
```

3. Írjon C nyelvű programot, amelyik a SIGTERM-hez hozzárendel egy fv.-t., amelyik kiírja az int paraméter értéket, majd végtelen ciklusban fusson, 3 sec-ig állandóan blokkolódva elindítás után egy másik shell-ben kill parancssal (SIGTERM) próbálja terminálni, majd SIGKILL-el.” Mentés. neptunkod\_gyak9\_2.c



```
h2z4x3_gyak9_2.c - Visual Studio Code
File Edit Selection View Go Run Terminal Help
C h2z4x3_gyak9_2.c
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

1 #include <stdio.h>
2 #include <sys/types.h>
3 #include <signal.h>
4 #include <unistd.h>
5
6
7 void kezelo(int i){
8     printf("Signal kezelese: %d\n", i);
9 }
10
11 int
12 main (void){
13
14     printf("PID = %d\n", getpid());
15     printf("Signal kezelo atvetele: %d\n", signal(SIGTERM, &kezo));
16
17     while(1){
18         printf("lepes\n");
19         sleep(3);
20     }
21 }
22
```

```
pazman@pazman-VirtualBox: ~/signal
pazman@pazman-VirtualBox:~$ cd signal
pazman@pazman-VirtualBox:~/signal$ ls
gyak9_02 h2z4x3_gyak9_2.c h2z4x3_tobbsignal.c h2z4x3.txt opc openclose
h2z4x3_gyak9_1.c h2z4x3_openclose.c h2z4x3_tobbsignal.c moresignal opencc ops
pazman@pazman-VirtualBox:~/signal$ gcc h2z4x3_gyak9_2.c -o gyak9_02
h2z4x3_gyak9_2.c:14:35: warning: format '%d' expects argument of type 'int', but argument 2 has type
'__sighandler_t' (aka 'void (*)(int)') [-Wformat=]
14 | printf("Signal kezelo atvetele: %d\n", signal(SIGTERM, &kezo));
    |                                ^~
    |                                |
    |                                int      __sighandler_t (aka void (*)(int))
pazman@pazman-VirtualBox:~/signal$ ./gyak9_02
PID = 4305
Signal kezelo atvetele: 0
lepes
lepes
lepes
lepes
lepes
lepes
lepes
lepes
lepes
lepes
```