

Operációs Rendszerek BSc

8. gyak.

2021. 03. 31.

Készítette:

Nyíri Beáta

Programtervező Informatikus

I40FDC

Miskolc, 2021

1. Tanulmányozzák a Dr. Vadász Dénes: Operációs rendszerek, 2006. ME, jegyzet Szignálok fejezetet 61-69 oldalig, majd...

Értelmezzék a mintapéldákat és oldják meg: *alarm.c*; *alarm_ado.c*; *alarmra_var.c* - szintén a jegyzet 68. oldalán található.

Mentés: *neptunkod_alarm.c*; *neptunkod_alarm_ado.c*; *neptunkod_alarmra_var.c*

i40fdc_alarm.c

```
1  /***** Vadász Dénes jegyzete alapján *****/
2
3  Abstract: Példa signal kezelésre.
4            ctrl/break signal hatására a do_int handler
5            működik először, majd ignorálódik ez a
6            signal.
7            A ciklusban másodpercenként alarm generálódik,
8            az alarm signal handler-e a do_nothing.
9            A pause felfüggeszti a process-t, amíg kap
10           egy signalt.
11
12  Internals: signal(), alarm(), pause()
13
14  *****/
15
16  #include <stdio.h>
17  #include <stdlib.h>
18  #include <unistd.h>
19  #include <signal.h>
20  #define SECOND 1
21
22  void do_nothing();
23  void do_int();
24
25  int main()
26  {
27      int i;
28      unsigned sec=1;
29
30      signal(SIGINT, do_int);
31
32      for(i=1; i<8; i++){
33          alarm(sec);
34          signal(SIGALRM, do_nothing);
35          printf(" %d varok de meddig?\n", i);
36          pause();
37      }
38  }
39
40  void do_nothing(){ ;}
41
42  void do_int(){
43      printf(" int jott ");
44      signal(SIGINT, SIG_IGN);
45  }
46
```

```

1 varok de meddig?
2 varok de meddig?
3 varok de meddig?
4 varok de meddig?
5 varok de meddig?
6 varok de meddig?
7 varok de meddig?

Process returned 0 (0x0)   execution time : 7.005 s
Press ENTER to continue.

```

Eredmény:

1 másodpercenként ír ki egy sort, amíg el nem ér 8-ig.

i40fdc_alarmra_var.c

```

1  /***** Vadász Dénes jegyzete alapján *****/
2
3  Abstract:  Példa signal kezelésre.
4             Az alarm signal handler-e a so_nothing.
5
6  Internals: signal(), pause()
7
8  *****/
9
10 #include <unistd.h>
11 #include <signal.h>
12
13 void do_nothing();
14
15 main()
16 {
17     signal(SIGALRM, do_nothing);
18     printf(" %d varok de meddig?\n");
19     pause();
20     printf(" Vegre, itt az alarm\n");
21 }
22
23 void do_nothing() { ; }
24

```

i40fdc_alarm_ado.c

```
1  /***** Vadász Dénes jegyzete alapján *****/
2
3  Abstract: Példa signal kezelésre.
4
5  Internals: atoi(), perror(), exit(), kill()
6
7  *****/
8
9  #include <stdio.h>
10 #include <stdlib.h>
11 #include <sys/types.h>
12 #include <signal.h>
13
14 int main(int argc, char **argv)
15 {
16     int pid;
17
18     if(argc < 1){
19         perror(" Nincs kinek ");
20         exit(1);
21     }
22
23     pid = atoi(argv[1]);
24
25     kill(pid, SIGALRM);
26 }
```

2. Készítse el a következő feladatot, melyben egy szignálkezelő több szignált is tud kezelni:
- Készítsen egy szignál kezelőt (`handleSignals`), amely a `SIGINT` (CTRL + C) vagy `SIGQUIT` (CTRL + \) jelek fogására vagy kezelésére képes.
 - Ha a felhasználó `SIGQUIT` jelet generál (akár `kill` paranccsal, akár billentyűzetről a CTRL + \) a kezelő egyszerűen kiírja az üzenetet visszatérési értékét – a konzolra.
 - 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.
 - Ha a felhasználó másodszor generálja a `SIGINT` jelet, akkor végrehajt egy alapértelmezett műveletet, amely a program befejezése - kiírás a konzolra.

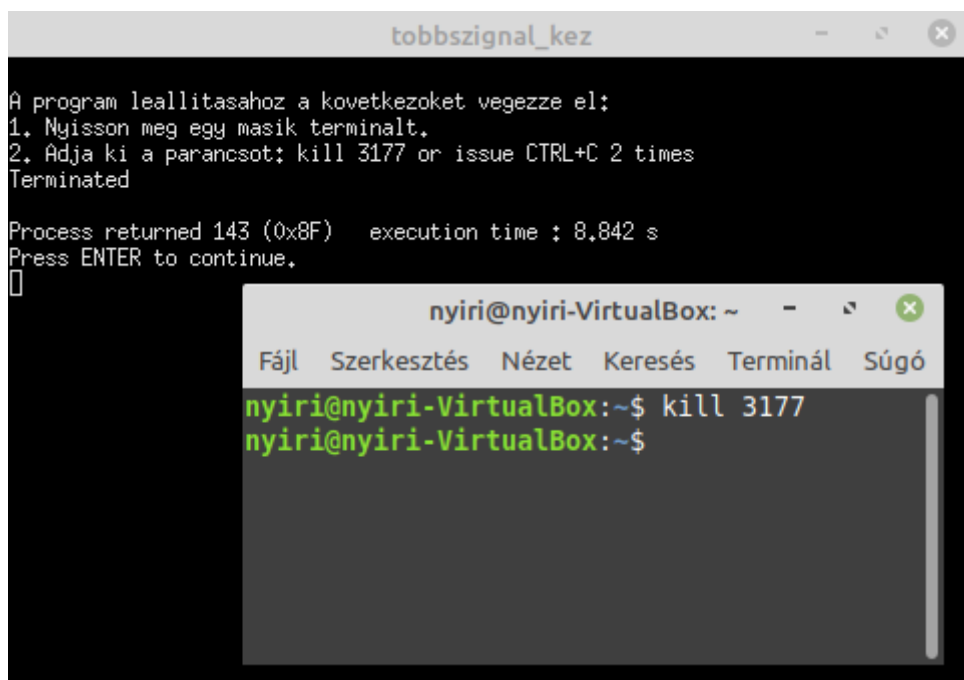
Mentés: `neptunkod_tobbsignal_kez.c`

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <signal.h>
5
6  void handleSignals(int signum);
7
8  int main(void)
9  {
10     void (*sigHandlerInterrupt)(int);
11     void (*sigHandlerQuit)(int);
12     void (*sigHandlerReturn)(int);
13     sigHandlerInterrupt = sigHandlerQuit = handleSignals;
14     sigHandlerReturn = signal(SIGINT, sigHandlerInterrupt);
15     if(sigHandlerReturn == SIG_ERR){
16         perror("signal error: ");
17         return 1;
18     }
19     sigHandlerReturn = signal(SIGQUIT, sigHandlerQuit);
20
21     if(sigHandlerReturn == SIG_ERR){
22         perror("signal error: ");
23         return 1;
24     }
25     while (1){
26         printf("\nA program leállításához a következőket vegezze el:\n");
27         printf("1. Nyisson meg egy másik terminált.\n");
28         printf("2. Adja ki a parancsot: kill %d or issue CTRL+C 2 times\n", getpid());
29         sleep(10);
30     }
31     return 0;
32 }
33
34 void handleSignals(int signum){
35     switch(signum){
36         case SIGINT:
37             printf("\nNyomja meg a CTRL+C billentyükombinációt!\n");
38             printf("Most visszaállítja a SIGINT jelet az alapértelmezett műveletre\n");
39             signal(SIGINT, SIG_DFL);
40             break;
41         case SIGQUIT:
42             printf("\nNyomja meg a CTRL+C billentyükombinációt!\n");
43             break;
44         default:
45             printf("\nFogadott jel száma %d\n", signum);
46             break;
47     }
48     return;
49 }
50
```

```
A program leallitasahoz a kovetkezoeket vegezze el:
1. Nyisson meg egy masik terminalt.
2. Adja ki a parancsot: kill 3150 or issue CTRL+C 2 times
^C
Nyomja meg a CTRL+C billentyukombinaciot!
Most visszaallitja a SIGINT jelet az alapertelmezett muveletre

A program leallitasahoz a kovetkezoeket vegezze el:
1. Nyisson meg egy masik terminalt.
2. Adja ki a parancsot: kill 3150 or issue CTRL+C 2 times
^C
Process returned -1 (0xFFFFFFFF)   execution time : 17.839 s
Press ENTER to continue.
```

CTRL+C esetén



```
tobbsignal_kez

A program leallitasahoz a kovetkezoeket vegezze el:
1. Nyisson meg egy masik terminalt.
2. Adja ki a parancsot: kill 3177 or issue CTRL+C 2 times
Terminated

Process returned 143 (0x8F)   execution time : 8.842 s
Press ENTER to continue.
[]
```

nyiri@nyiri-VirtualBox: ~

Fájl Szerkesztés Nézet Keresés Terminál Súgó

```
nyiri@nyiri-VirtualBox:~$ kill 3177
nyiri@nyiri-VirtualBox:~$
```

kill „pid” esetén