

# **Operációs rendszerek BSc**

8. gyakorlat

2021. április 12. (hétfő) 16:00 – 18:00

**Készítette:**

Szeli Márk

Gazdaságinformatikus

B8VNQ7

**1. feladat:** 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

*B8VNQ7\_alarm.c:*

```
1  #include <unistd.h>
2  #include <signal.h>
3  #define SECOND 1
4
5  void do_nothing();
6  void do_int();
7
8  main ()
9  {
10     int i;
11     unsigned sec = 1;
12
13     signal(SIGINT, do_int);
14
15     for (i = 1; i < 8; i++) {
16         alarm(sec);
17         signal(SIGALRM, do_nothing);
18         printf("%d varakozik, meddig?\n", i);
19         pause();
20     }
21 }
22 void do_nothing(){ ;}
23
24 void do_int() {
25     printf("int érkezett");
26     signal(SIGINT,SIG_IGN);
27 }
28
29
```

Eredmény:

```
mark@mark-K72Jr:~$ ./B8VNQ7_alarm.out
1 varakozik, meddig?
2 varakozik, meddig?
3 varakozik, meddig?
4 varakozik, meddig?
5 varakozik, meddig?
6 varakozik, meddig?
7 varakozik, meddig?
```

*B8VNQ7\_alarm\_ado.c:*

```
1  #include <sys/types.h>
2  #include <signal.h>
3
4  main(int argc, char **argv)
5  {
6      int pid;
7
8      if (argc < 1)
9      {
10         perror("Nincs kinek!");
11         exit(1);
12     }
13
14     pid = atoi(argv[1]);
15
16     kill(pid, SIGALRM);
17 }
```

Eredmény:

```
mark@mark-K72Jr:~$ ./B8VNQ7_alarm_ado.out
Szegmentálási hiba (core készült)
```

*B8VNQ7\_alarmra\_var.c:*

```
1  #include <unistd.h>
2  #include <signal.h>
3
4  void do_nothing();
5
6  main ()
7  {
8      signal(SIGALRM, do_nothing);
9      printf("%d varok, de meddig?\n");
10     pause();
11     printf("Vegre, itt az alarm!\n");
12 }
13 void do_nothing(){ ;}
```

Eredmény:

```
mark@mark-K72Jr:~$ ./B8VNQ7_alarmra_var.out
-2116082144 varok, de meddig?
```

**2. feladat:** Készítsen 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 értéként – 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\_tobbsignal\_kez.c

*B8VNQ7\_tobbsignal\_kez.c*

```
1  #include <signal.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <stdio.h>
5  #include <string.h>
6
7  void handleSignal();
8
9  int main(){
10
11     signal(SIGINT, handleSignal);
12     signal(SIGQUIT, handleSignal);
13
14     for (int i = 0; i < 10; i++){
15         printf("Varakozas...\n");
16         pause();
17     }
18
19     return 0;
20 }
21
22 void handleSignal(int sig){
23     char desc[50];
24
25     switch(sig){
26     case 1: {
27         strcpy(desc, "Hangup Signal\n");
28         break;
29     }
30
31     case 2: {
32         strcpy(desc, "Interrupt Signal\n");
33         signal(SIGINT, SIG_DFL);
34         printf("\nA 'CTRL + C' parancs ki fogja leptetni a programbol!\n");
35         break;
36     }
37
38     case 3: {
39         strcpy(desc, "Quit Signal\n");
40         break;
41     }
42 }
```

```
42
43     case 4: {
44         strcpy(desc, "Illegal Instruction Signal\n");
45         break;
46     }
47
48     case 5: {
49         strcpy(desc, "Trace Trap Signal\n");
50         break;
51     }
52
53     case 6: {
54         strcpy(desc, "Abort Signal\n");
55         break;
56     }
57
58     printf("\nAz elkapott jel: %s\n", desc);
59 }
60 }
```

Eredmény:

```
~$ ./tobbszignal-kez.out
Varakozas...
^C
A 'CTRL + C' parancs ki fogja leptetni a programbol!
Varakozas...
^C
```

**3. feladat:** Processz családon belül szinkronizáljon szignál kezeléssel. Szülő processz küldjön pl. 10 másodpercenként jelzést (pl. SIGALRM, SIGINT, SIGUSR1, SIGUSR2) gyermekének. A gyermek a szignált futásának megváltoztatására használja - szintén a jegyzet 68. oldalán található. Specifikáljon erre feladatokat! Mentés: *neptunkod\_szink\_sigkez.c*

*B8VNQ7\_szink\_sigkez.c:*

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <signal.h>
5  #include <sys/types.h>
6
7  void alarmkap()
8  {
9      printf("SIGUSR1 kuldes a gyermekprocessznek...\n");
10 }
11
12 void signalusr()
13 {
14     printf("SIGUSR1 kapott a gyermekprocessz!\n");
15 }
16
17 int main(void)
18 {
19     pid_t pid;
20     int stop = 5;
21     int time = 10;
22
23     if((pid = fork()) < 0)
24     {
25         perror("Fork error!\n");
26     }
27
28     else if(pid == 0)
29     {
30         printf("A gyermekprocessz kimenete.\n");
31         signal(SIGUSR1, signalusr);
32
33         for(int i = 0; i < stop; i++)
34         {
35             pause();
36             printf("A gyermekprocessz állpota megváltozott!\n");
37         }
38
39         printf("A gyermekprocessz leallt!\n");
40     }
41
42     else
43     {
44         printf("A szuloprocessz kimenetele.\n");
45         signal(SIGALRM, alarmkap);
```

```
47         for(int i = 0; i < stop; i++)
48         {
49             alarm(time);
50             pause();
51             printf("PID erteke: %d\n", pid);
52             kill(pid, SIGUSR1);
53         }
54
55         printf("A szuloprocessz leallt!\n");
56     }
57
58     exit(0);
59 }
```

Eredmény:

```
mark@mark-K72Jr:~$ ./B8VNQ7_szink_sigkez.out
A szuloprocessz kimenetele.
A gyermekprocessz kimenetele.
SIGUSR1 kuldes a gyermekprocessznek...
PID erteke: 2839
SIGUSR1 kapott a gyermekprocessz!
A gyermekprocessz allpota megvaltozott!
SIGUSR1 kuldes a gyermekprocessznek...
PID erteke: 2839
SIGUSR1 kapott a gyermekprocessz!
A gyermekprocessz allpota megvaltozott!
SIGUSR1 kuldes a gyermekprocessznek...
PID erteke: 2839
SIGUSR1 kapott a gyermekprocessz!
A gyermekprocessz allpota megvaltozott!
SIGUSR1 kuldes a gyermekprocessznek...
PID erteke: 2839
A szuloprocessz leallt!
SIGUSR1 kapott a gyermekprocessz!
A gyermekprocessz allpota megvaltozott!
A gyermekprocessz leallt!
```