

# Operációs rendszerek

## Nyolcadik gyakorlat

Készítette: Énekes Zoltán László

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

### **alarm.c**

```
#include <unistd.h>
#include <signal.h>
#define SECOND 1

void do_nothing();
void do_int();

main ()
{
    int i;
    unsigned sec=1;

    signal(SIGINT, do_int);

    for (i=1;i<8;i++) {
        alarm(sec);
        signal(SIGALRM, do_nothing);
        printf(" %d Varok de meddig?\n",i);
        pause();
    }
}

void do_nothing(){ ;}

void do_break(){
    printf("Megszakitas jott!\n");
    signal(SIGINT, SIG_IGN);
}
```

### **alarm\_ado**

```
#include <sys/types.h>
#include <signal.h>

main(int argc, char **argv)
{
    int pid;

    if (argc < 1)
    {
        perror("Hiba!\n");
        exit(1);
    }

    pid = atoi(argv[1]);
```

```

        kill(pid, SIGALRM);

    return 0;
}

```

### alarmra\_var

```

#include <unistd.h>
#include <signal.h>
#include <stdio.h>
#include <stdlib.h>

void do_nothing();

int main(){

    unsigned sec = 5;
    signal(SIGALRM, do_nothing);
    alarm(sec);

    printf("%d varok.\n",1);
    pause();

    printf("Megerkezett az alarm.\n");

    return 0;
}

void do_nothing(){;}

```

**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\_tobbsignal\_kez.c

tobbsignal\_kez

```

#include <stdio.h>
#include <stdlib.h>

#include <unistd.h>
#include <signal.h>

```

```

void InterruptHandler(int sig);
void QuitHandler(int sig);

unsigned int Interrupts = 0;

int main(void)
{
    if (signal(SIGINT, InterruptHandler) == SIG_ERR)
    {
        printf("Signal hiba!\n");
        return 0;
    }

    if (signal(SIGQUIT, QuitHandler) == SIG_ERR)
    {
        printf("Jel hiba!\n");
        return 0;
    }

    while(Interrupts < 2)
    {
        printf("Varakozas jelre...\n");
        sleep(1);
    }

    printf("Megerkezett a masodik jel!");
    return 0;
}

void InterruptHandler(int sig)
{
    printf("Kapott jelek száma %d\n", sig);
    Interrupts++;
}

void QuitHandler(int sig)
{
    printf("Kilépési jelek %d\n", sig);
}

```