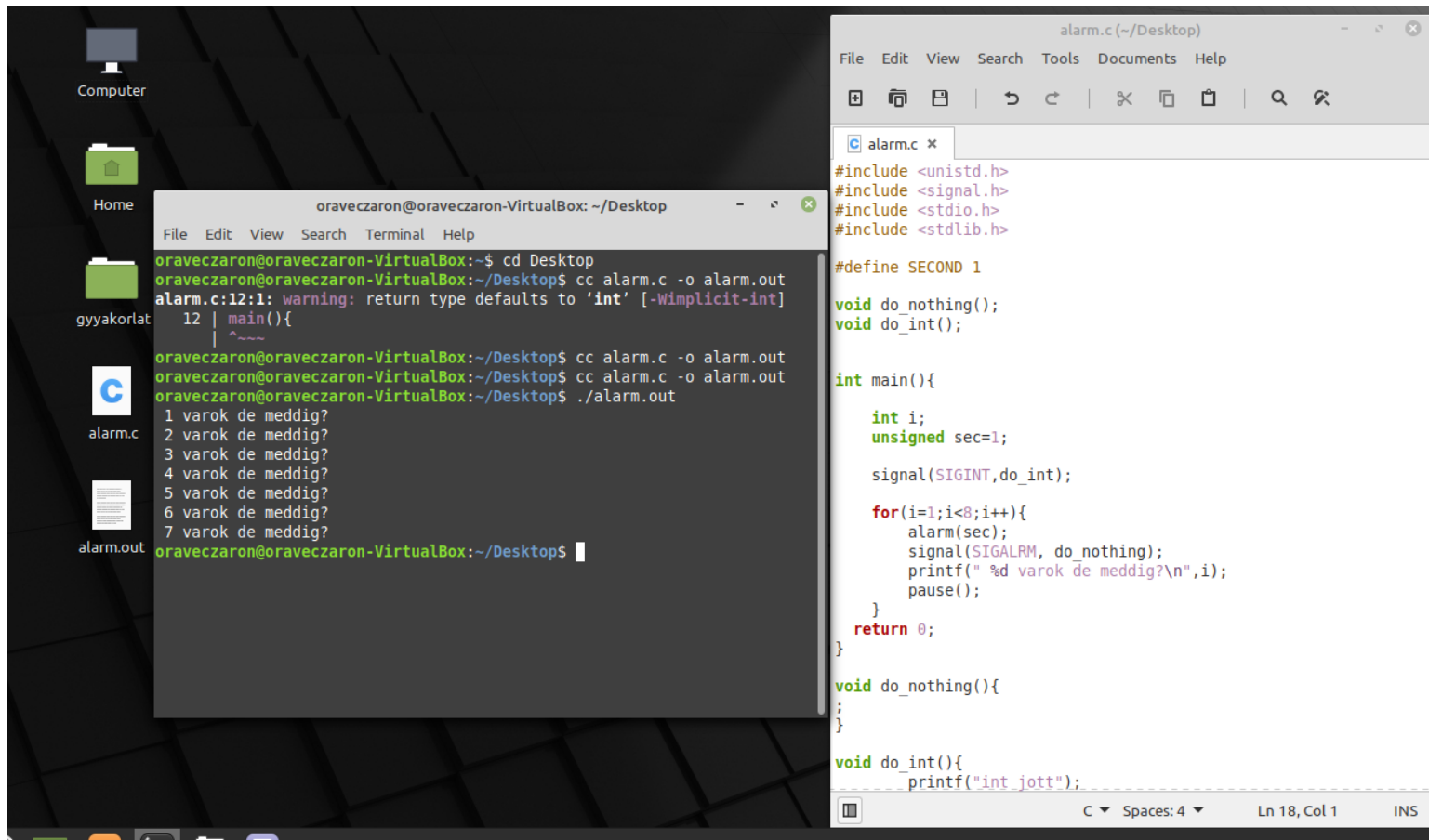


1, feladat:

alarm.c



```
oraveczaron@oraveczaron-VirtualBox: ~/Desktop
File Edit View Search Terminal Help
oraveczaron@oraveczaron-VirtualBox:~$ cd Desktop
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ cc alarm.c -o alarm.out
alarm.c:12:1: warning: return type defaults to 'int' [-Wimplicit-int]
12 | main(){
    | ^~~~~
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ cc alarm.c -o alarm.out
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ cc alarm.c -o alarm.out
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ ./alarm.out
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?
oraveczaron@oraveczaron-VirtualBox:~/Desktop$
```

```
alarm.c (~/Desktop)
File Edit View Search Tools Documents Help
alarm.c x
#include <unistd.h>
#include <signal.h>
#include <stdio.h>
#include <stdlib.h>

#define SECOND 1

void do_nothing();
void do_int();

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

    signal(SIGINT,do_int);

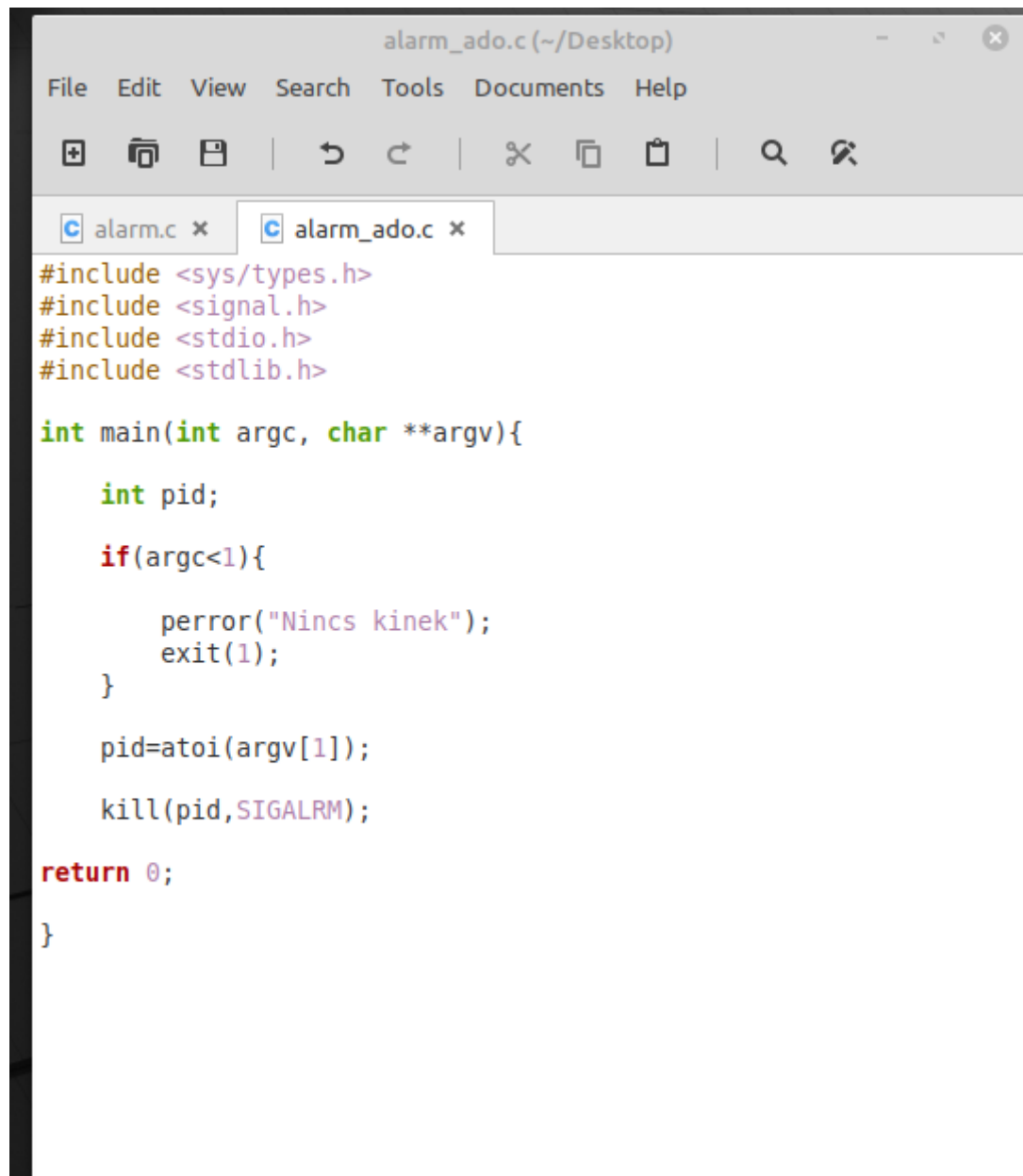
    for(i=1;i<8;i++){
        alarm(sec);
        signal(SIGALRM, do_nothing);
        printf(" %d varok de meddig?\n",i);
        pause();
    }
    return 0;
}

void do_nothing(){
    ;
}

void do_int(){
    printf("int jott");
}
```

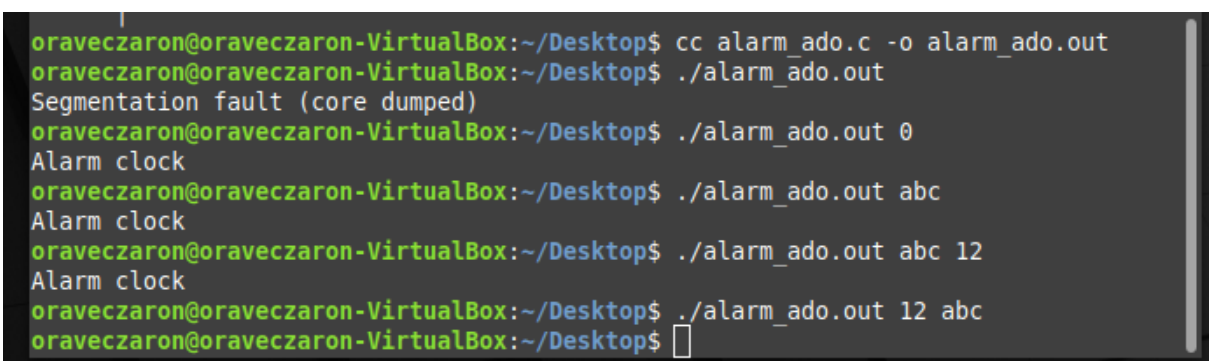
- Figyeljük, hogy kap-e jelzést a processz ha kap nevelődik, de nem tudjuk meddig, majd kap egy újabb jelzést és ezeket a jelzéseket addig számolja meg míg a programba el nem érjük a megadott maximumot.

alarm_ado.c



```
alarm_ado.c (~/Desktop)
File Edit View Search Tools Documents Help
+ [icon] [icon] [icon] | [icon] [icon] | [icon] [icon] [icon] | [icon] [icon]
alarm.c x alarm_ado.c x
#include <sys/types.h>
#include <signal.h>
#include <stdio.h>
#include <stdlib.h>

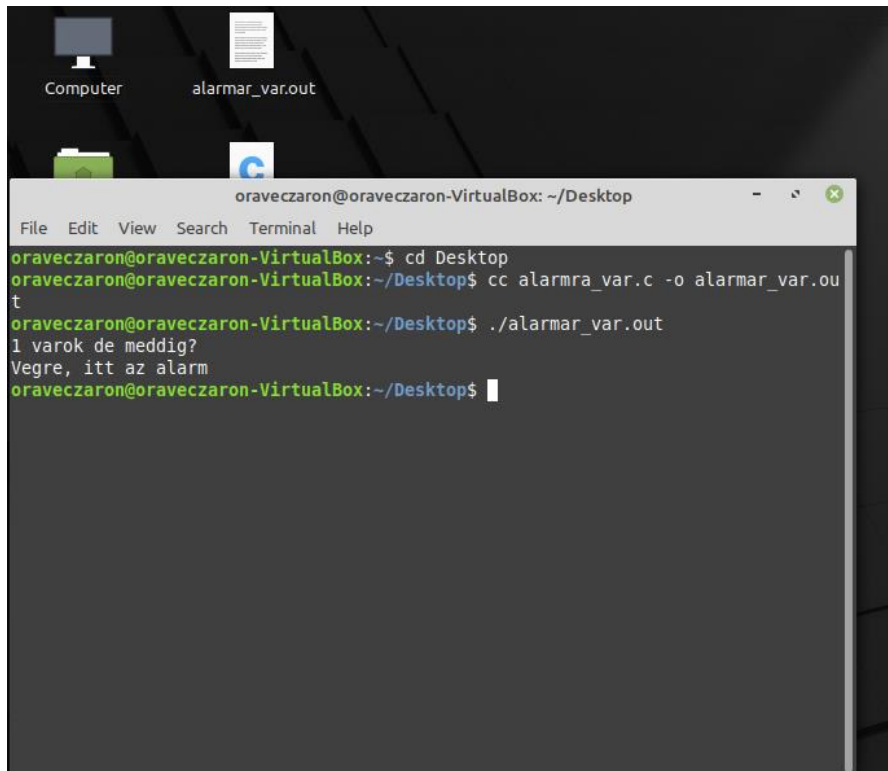
int main(int argc, char **argv){
    int pid;
    if(argc<1){
        perror("Nincs kinek");
        exit(1);
    }
    pid=atoi(argv[1]);
    kill(pid,SIGALRM);
    return 0;
}
```



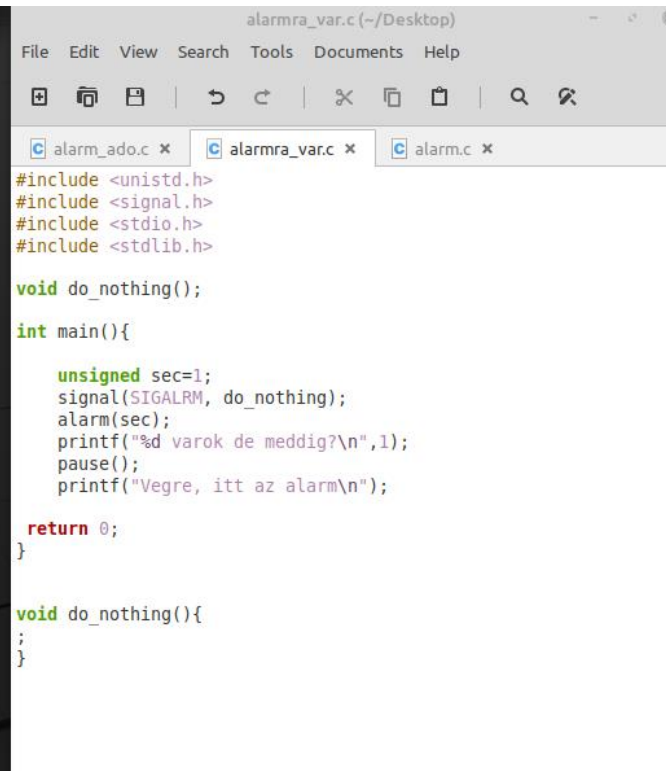
```
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ cc alarm_ado.c -o alarm_ado.out
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ ./alarm_ado.out
Segmentation fault (core dumped)
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ ./alarm_ado.out 0
Alarm clock
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ ./alarm_ado.out abc
Alarm clock
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ ./alarm_ado.out abc 12
Alarm clock
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ ./alarm_ado.out 12 abc
oraveczaron@oraveczaron-VirtualBox:~/Desktop$
```

- kiírtatásnál kérünk számot akkor „Alarm clock”-ot ír ki.

alarmra_var.c



```
oraveczaron@oraveczaron-VirtualBox: ~/Desktop
File Edit View Search Terminal Help
oraveczaron@oraveczaron-VirtualBox:~$ cd Desktop
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ cc alarmra_var.c -o alarmra_var.out
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ ./alarmra_var.out
1 varok de meddig?
Vegre, itt az alarm
oraveczaron@oraveczaron-VirtualBox:~/Desktop$
```



```
alarmra_var.c (~/Desktop)
File Edit View Search Tools Documents Help
alarm_ado.c x alarmra_var.c x alarm.c x
#include <unistd.h>
#include <signal.h>
#include <stdio.h>
#include <stdlib.h>

void do_nothing();

int main(){

    unsigned sec=1;
    signal(SIGALRM, do_nothing);
    alarm(sec);
    printf("%d varok de meddig?\n",1);
    pause();
    printf("Vegre, itt az alarm\n");

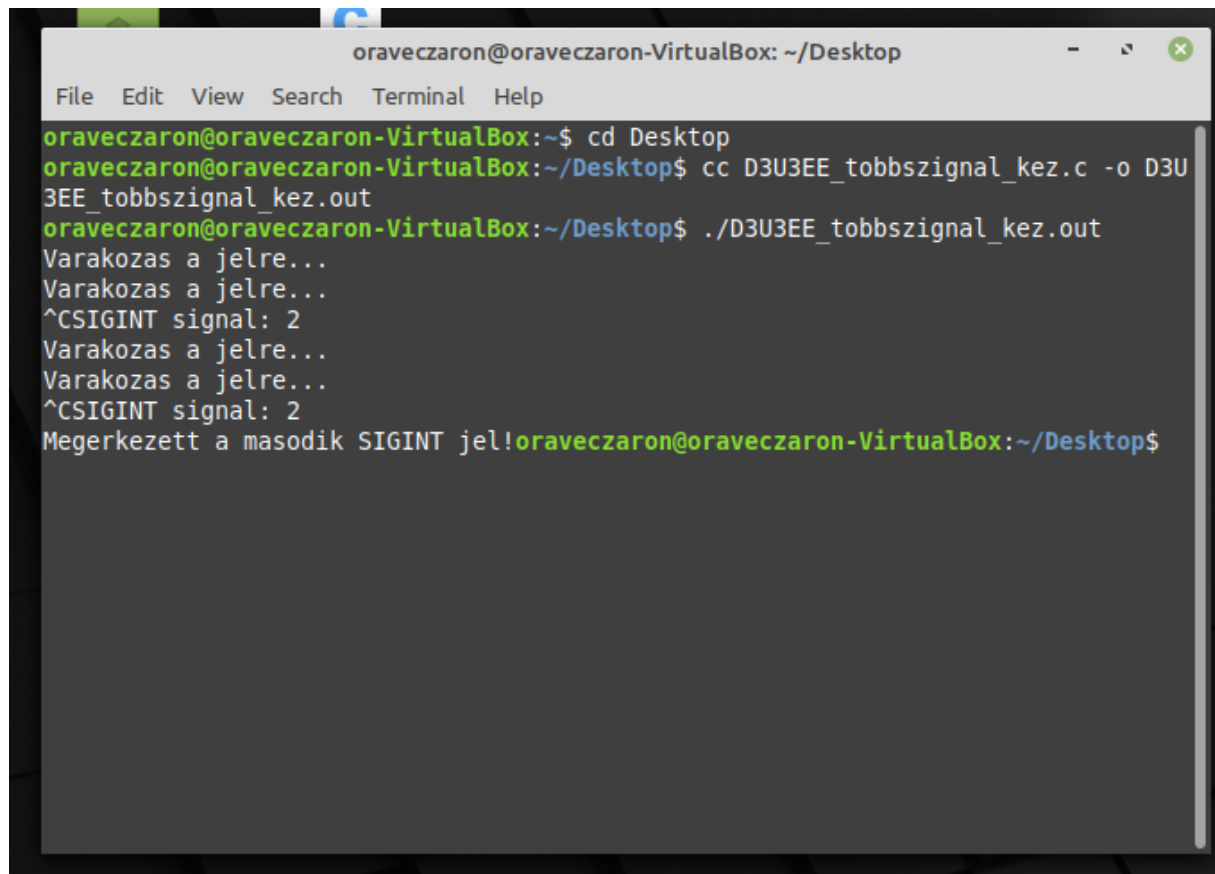
    return 0;
}

void do_nothing(){
;
}
```

Megvárja azt az idővallumot amit mi megadtunk secnek és utána kiírja hog „Végre itt aaz alarm” ...

2, feladat.:

D3U3EE_tobbszignal_kez.c



```
oraveczaron@oraveczaron-VirtualBox: ~/Desktop
File Edit View Search Terminal Help
oraveczaron@oraveczaron-VirtualBox:~$ cd Desktop
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ cc D3U3EE_tobbszsignal_kez.c -o D3U3EE_tobbszsignal_kez.out
oraveczaron@oraveczaron-VirtualBox:~/Desktop$ ./D3U3EE_tobbszsignal_kez.out
Varakozas a jelre...
Varakozas a jelre...
^CSIGINT signal: 2
Varakozas a jelre...
Varakozas a jelre...
^CSIGINT signal: 2
Megerkezett a masodik SIGINT jel!oraveczaron@oraveczaron-VirtualBox:~/Desktop$
```

```

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

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

void Megszakitas(int jel);
void kilepes(int jel);

unsigned int Megszakitasok = 0;

int main(void)
{
    if (signal(SIGINT, Megszakitas) == SIG_ERR)
    {
        printf("Nem sikerult handlert allitani a(z) SIGINT jelre!\n");
        return 0;
    }

    if (signal(SIGQUIT, kilepes) == SIG_ERR)
    {
        printf("Nem sikerult handlert allitani a(z) SIGQUIT jelre!\n");
        return 0;
    }

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

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

void Megszakitas(int jel)
{
    printf("SIGINT signal: %d\n", jel);
    Megszakitasok++;
}

void kilepes(int jel)
{
    printf("SIGQUIT signal: %d\n", jel);
}

```

Program futásakor a program vár egy jelre addig míg az meg nem adjuk ctrl+c -vel majd kér egy második jelet is és vége a programnak ha mindkettő jelet megkapta.