„1.

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

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

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

void do_nothing();
void do_int();

int main(void)
{
    unsigned sec = 1;
    signal(SIGINT, do_int);

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

void do_nothing() { }

void do_int()
{
    printf("int jott");
    signal(SIGINT, SIG_IGN);
}
```

alarm_ado.c

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

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

    if (argc < 1)
    {
        perror("Nincs kinek");
        exit(1);
    }

    pid = atoi(argv[1]);
    kill(pid, SIGALRM);
}
```

alarmra_var.c

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

void do_nothing();

int main(void)
{
    signal(SIGALRM, do_nothing);
    printf("%d varok de meddig?\n");
    pause();
    printf("Vegre, itt az alarm \n");
}

void do_nothing() {}
```

A 2. és a 3. feladat opcionális, vagy az egyiket vagy a másikat kell megoldani. 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.

tobbszignal_kez.c

```c
#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("Nem sikerult handlert allitani a(z) \"SIGINT\" jelre!\n");
        return 0;
    }

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

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

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

void InterruptHandler(int sig)
{
    printf("SIGINT signal: %d\n", sig);
    Interrupts++;
}

void QuitHandler(int sig)
{
    printf("SIGQUIT signal: %d\n", sig);
}
```