Operációs rendszerek BSc

8.gyak.

2021. 03. 31.

Készítette:

Tóth József BProf
Üzemmérnök-informatikus alapszak
WI2GDP

Miskolc, 2021

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

   neptunkod_alarm.c:

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

   A main-en belül addig megyünk végig a SIGALRM lekérdezés alatt, amíg nem történik valamilyen signal megszakítás, mely elindítja a do_int függvényt. A pause felfüggeszti a processzt, amíg kap egy signalt.

neptunkod_alarm_ado.c és a neptunkod_alarm_var.c:

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

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

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

    pid = atoi(argv[1]);

    kill(pid, SIGALRM);
}
```

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


void do_nothing();


main ()
{


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

A két forrásfájl együtt dolgozik. Az alarm_ado elindításánál vár egy bemeneti paramétert, ami megfelel a pid-nek, majd ezt az értéket a kill() paranccsal átpostázza a SIGALRM-nak. Így az alarm_var-ból a do_nothing metódus a SIGALRM-nak megfelelően fog lefutni.

2. **feladat -** *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_tobbszignal_kez.c

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

void handleSignals(int signum);

int main() {
    void(*sigHandlerInterrupt)(int);
    void(*sigHandlerQuit)(int);
    void(*sigHandlerReturn)(int);
    sigHandlerInterrupt = sigHandlerQuit = handleSignals;
    sigHandlerReturn = signal(SIGINT, sigHandlerInterrupt);

    if (sigHandlerReturn == SIG_ERR) {
        perror( s: "Signal error");
        return 1;
    }

    sigHandlerReturn = signal(SIGQUIT, sigHandlerQuit);

    if (sigHandlerReturn == SIG_ERR) {
        perror( s: "Signal error");
        return 1;
    }
    for(;;) {
        printf( format: "\nA program leallitasahoz a kovetkezoket vegezze el: \n");
        printf( format: "1. Nyisson meg egy másik terminalt.\n");
        printf( format: "2. Adja ki a parancsot: kill: %d \n", getpid());
        sleep( seconds: 10);
    }
    return 0;
}
```

```c
void handleSignals(int signum) {
    switch(signum) {
        case SIGINT:
            printf( format: "\n CTRL+C-t eszlelt\n");
            signal(SIGINT, SIG_DFL);
            break;
        case SIGQUIT:
            printf( format: "SIQUIT aktivalodott\n");
            break;
        default:
            printf( format: "\nFogadott jel szama: %d\n", signum);
            break;
    }
    return ;
}
```

```
A program leallitasahoz a kovetkezoket vegezze el:
1. Nyisson meg egy másik terminalt.
2. Adja ki a parancsot: kill: 5139
joseph@joseph-virtual-machine:~/Documents/WI2GDPGyak/WI2GDPOsGyak/WI2GDP_0331$ gcc WI2GDP_tobbszignal_kez.c -o teszt
joseph@joseph-virtual-machine:~/Documents/WI2GDPGyak/WI2GDPOsGyak/WI2GDP_0331$ ./teszt

A program leallitasahoz a kovetkezoket vegezze el:
1. Nyisson meg egy másik terminalt.
2. Adja ki a parancsot: kill: 5204
Terminated
joseph@joseph-virtual-machine:~/Documents/WI2GDPGyak/WI2GDPOsGyak/WI2GDP_0331$
```

```
bash: 1: command not found
joseph@joseph-virtual-machine:~$ kill 5204
joseph@joseph-virtual-machine:~$
```