

Operációs rendszerek BSc

8.gyak.

2021. 03. 31.

Készítette:
Kovács Krisztián
Programtervező informatikus
WIQPM2

Miskolc, 2021

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

```
signal(SIGINT, do_int);

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

Amikor a felhasználó kiadja a signalt, akkor a do_int függvény hajtódik végre. A ciklus nyolcszor lefut, amiben másodpercenként alrmet küldünk, amit lekezelünk, a do_nothing fog lefutni.

```
void do_nothing() { ;}

void do_int() {
    printf(" Megszakítás jött ");
    signal(SIGINT, SIG_IGN);
}
```

Itt látható hogy a do_nothing lefutásakor semmi se történik, viszont a do_int úgy kezeli le a signalt, hogy kiír egy szöveget és ugynennek a signalnak most lecseréljük a handlerjét, arra, hogy ignoráljuk.

```
int pid=3347;

printf(" %s ", argv[0]);
if(argc < 1){
    perror("Nincs kinek");
    exit(1);
}

pid = atoi(argv[1]);

kill(pid, SIGALRM);
```

A második részfeladatot nem tudtam lefuttatni, nem tudom miért, viszont értem nagyjából hogy működik. Lekértem a pid-et a varónak, és ha van kinek, akkor elküld egy signalt, gondolom a kill segítségével.

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

void do_nothing() { ;}
```

Megkapja a signalt és lekezeli.

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

```
^C
A CTRL + C kovetkezo lenyomasakor az alapertelmezett kezelo fog futni
^C
Process returned -1 (0xFFFFFFFF)   execution time : 23,196 s
Press ENTER to continue.
```

```
^C
A CTRL + C kovetkezo lenyomasakor az alapertelmezett kezelo fog futni
^\\
Az elkapott jel: Quit
Process returned -1 (0xFFFFFFFF)   execution time : 17,415 s
Press ENTER to continue.
```

```
^\\
Az elkapott jel: Quit
Process returned -1 (0xFFFFFFFF)   execution time : 3,555 s
Press ENTER to continue.
```