

8. operációs rendszerek gyakorlat

1. Feladat:

```
#include <stdio.h>
#include <unistd.h>
#include <signal.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();
    }
}

void do_nothing(){ ;}

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

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <sys/types.h>
4  #include <signal.h>
5
6  int main(int argc, char **argv)
7  {
8      int pid;
9
10     if (argc < 1)
11     {
12         perror(" Nincs kinek");
13         exit(1);
14     }
15
16     pid = atoi(argv[1]);
17
18     kill(pid, SIGALRM);
19 }
```

```
1  #include <stdio.h>
2  #include <unistd.h>
3  #include <signal.h>
4
5  void do_nothing();
6
7  int main ()
8  {
9      signal(SIGALRM, do_nothing);
10     printf(" %d varok de meddig?\n");
11     pause();
12     printf(" Vegre, itt az alarm \n");
13 }
14 void do_nothing(){ ;}
15
```

2. Feladat:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <signal.h>
5  #include <sys/wait.h>
6
7  int main()
8  {
9      pid_t cpid;
10     cpid = fork();
11
12     if(cpid < 0)
13     {
14         printf("Sikertelen rendszer hívás!");
15     }
16     else if(cpid == 0)
17     {
18         for (alarm(20); pause() != SIGALRM;);
19     }
20     else
21     {
22         exit(1);
23     }
24
25     return 0;
26 }
```

```
void handleSignals(int signum);

int main(void) {
    void (*sigHandlerInterrupt)(int);
    void (*sigHandlerQuit)(int);
    void (*sigHandlerReturn)(int);
    sigHandlerInterrupt = sigHandlerQuit = handleSignals;
    sigHandlerReturn = signal(SIGINT, sigHandlerInterrupt);
    if (sigHandlerReturn == SIG_ERR) {
        perror("signal error: ");
        return 1;
    }
    sigHandlerReturn = signal(SIGQUIT, sigHandlerQuit);

    if (sigHandlerReturn == SIG_ERR) {
        perror("signal error: ");
        return 1;
    }
    while (1) {
        printf("\nA program leállításához a következőket végezze el: \n");
        printf("1. Nyisson meg egy másik terminált.\n");
        printf("2. Adja ki a parancsot: kill %d or issue CTRL+C 2 times (másodszor is megszűnik)\n", getpid());
        sleep(10);
    }
}
```

```
    sleep(10);
}
return 0;
}

void handleSignals(int signum) {
    switch(signum) {
        case SIGINT:
            printf("\nNyomja meg a CTRL + C billentyűkombinációt\n");
            printf("Most visszaállítja a SIGINT jelet az alapértelmezett műveletre\n");
            signal(SIGINT, SIG_DFL);
            break;
        case SIGQUIT:
            printf("\nNyomta meg a CTRL+\n");
            break;
        default:
            printf("\nFogadott jel száma %d\n", signum);
            break;
    }
    return;
}
```