

```

#include <signal.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/errno.h>

void my_sighandlerINT();
void my_sighandlerQUIT();

int main()
{
    int a=1;

    printf("Wybierz sposob dzialania programu:\n"
        "1 - Wykonanie operacji domyslniej\n"
        "2 - Ignorowanie sygnalu\n"
        "3 - Przechwycenie sygnalu\n"
        "4 - Wysluj do siebie SIGINT za pomoca funkcji kill\n");
    scanf("%d",&a);

    switch(a){
        case 1:
            printf("Program odpowie domyslnie na sygnal\n");
            if (signal(SIGQUIT,SIG_DFL) == SIG_ERR){ /* po otrzymaniu sygnalu
SIGQUIT wykona sie SIG_DFL */
                perror("Funkcja signal ma problem z SIGQUIT");
                exit(EXIT_FAILURE);
            }
            if (signal(SIGINT,SIG_DFL) == SIG_ERR){
                perror("Funkcja signal ma problem z SIGINT");
                exit(EXIT_FAILURE);
            }
            if (pause() < 0){
                perror("ERROR: sygnal nie powoduje zakoczenia procesu");
                exit(EXIT_FAILURE);
            }
            break;
        case 2:
            printf("Program zignoruje sygnal\n");
            if (signal(SIGQUIT,SIG_IGN) == SIG_ERR){
                perror("Funkcja signal ma problem z SIGQUIT");
                exit(EXIT_FAILURE);
            }
            if (signal(SIGINT,SIG_IGN) == SIG_ERR){
                perror("Funkcja signal ma problem z SIGINT");
                exit(EXIT_FAILURE);
            }
            /* Tutaj pause nie ma sensu bo sygnaly zostana zignorowane */
            /* aby popchnac program do przodu trzeba uzyc SIGKILL, lub
            wpisac dowolny znak i nacisnac enter */
            printf("Aby zakonczyc program uzyj SIGKILL, badz wprowadz\n"
                "dowolny znak i nacisnij enter\n");
            scanf("%d",&a);
            /*if (pause() < 0){
                perror("ERROR: sygnal nie powoduje zakoczenia procesu");
                exit(EXIT_FAILURE);
            }*/
            break;
        case 3:
            printf("Program wypisze rodzaj otrzymanego sygnalu\n");
            if (signal(SIGQUIT,my_sighandlerQUIT) == SIG_ERR){

```

```

        perror("Funkcja signal ma problem z SIGQUIT");
        exit(EXIT_FAILURE);
    }
    if (signal(SIGINT, my_sighandlerINT) == SIG_ERR){
        perror("Funkcja signal ma problem z SIGINT");
        exit(EXIT_FAILURE);
    }
    pause(); /*tutaj na pewno pause zwroci blad, nasza obsluga sygnalu nie
    pozwala na zakonczenie programu! Dlatego ignorujemy zwracana przez pause
    wartosc */
    break;
case 4:
    printf("Wysylam do siebie SIGINT!\n");
    sleep(1);
    if(kill(getpid(), SIGINT)){
        printf("ERROR: blad funkcji kill\n");
        exit(EXIT_FAILURE);
    }
    break;
default:
    printf("bledny wybor menu, konczenie pracy\n");
    exit(1);
}

return 0;
}

void my_sighandlerINT(int sig){
    printf("Otrzymano SIGINT\n");
    exit(EXIT_SUCCESS);
}
void my_sighandlerQUIT(int sig){
    printf("Otrzymano SIGQUIT\n");
    exit(EXIT_SUCCESS);
}

```