

Procesy - sygnały

Zadanie 1

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

#define KOM_SIGINT "Bład funkcji signal (SIGINT)\n"
#define KOM_SIGCONT "Bład funkcji signal (SIGCONT)\n"
#define KOM_SIGQUIT "Bład funkcji signal (SIGQUIT)\n"

void Przechwycenie_Sygnalu(int sygnal)
{
    char xx[256];
    sprintf(xx, "PID %d: Przechwycilem sygnal %d\n", getpid(), sygnal);
    write(STDERR_FILENO, xx, strlen(xx));

    /* W niektórych systemach tutaj należy powtorzyc wywołanie funkcji
    signal */
}

main()
{
    /* Przechwycenie SIGINT(2) Ctrl-C */
    if (signal(SIGINT, Przechwycenie_Sygnalu) == SIG_ERR) {
        write(STDERR_FILENO, KOM_SIGINT, strlen(KOM_SIGINT));
        return(1); }

    /* Przechwycenie SIGCONT (18) */
    if (signal(SIGCONT, Przechwycenie_Sygnalu) == SIG_ERR) {
        write(STDERR_FILENO, KOM_SIGCONT, strlen(KOM_SIGCONT));
        return(1); }

    /* Przechwycenie SIGQUIT(3) Ctrl-\ */
    if (signal(SIGQUIT, Przechwycenie_Sygnalu) == SIG_ERR) {
        write(STDERR_FILENO, KOM_SIGQUIT, strlen(KOM_SIGQUIT));
        return(1); }

    while (1)
    { ; /* tutaj mozna wstawic ciało procesu */
    }
}
```

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

/* PROGRAM PRZECHWYTUJE TRZY SYGNALY ZADANE PRZEZ PARAMETRY */

#define KOM "Bład funkcji signal\n"

void Przechwycenie_Sygnalu(int sygnal)
{
    char xx[256];
    sprintf(xx, "PID %d: Przechwycilem sygnal %d\n", getpid(), sygnal);
    write(STDERR_FILENO, xx, strlen(xx));

    /* W niektórych systemach tutaj należy powtorzyc wywołanie funkcji
    signal */
}

int main(int arg, char **arc)
{
    int p1, p2, p3;
    char xx[100];

    if (arg != 4) {
        sprintf(xx, "\n Proces %s należy wywołać z trzema parametrami\n
        Ilość podanych parametrow: %d\n\n", arc[0], arg-1);
        write(STDOUT_FILENO, xx, strlen(xx));
        return(1); }

    p1=atoi(arc[1]);
    p2=atoi(arc[2]);
    p3=atoi(arc[3]);
    sprintf(xx, "\n Proces %s\n Obsługiwane sygnały: %d %d %d\n\n",
        arc[0], p1, p2, p3);
    write(STDOUT_FILENO, xx, strlen(xx));

    if (signal(p1, Przechwycenie_Sygnalu) == SIG_ERR) {
        write(STDERR_FILENO, KOM, strlen(KOM));
        return(1); }
    if (signal(p2, Przechwycenie_Sygnalu) == SIG_ERR) {
        write(STDERR_FILENO, KOM, strlen(KOM));
        return(1); }
    if (signal(p3, Przechwycenie_Sygnalu) == SIG_ERR) {
        write(STDERR_FILENO, KOM, strlen(KOM));
        return(1); }

    while (1)
    { ; /* tutaj można wstawić ciało procesu */
    }
}
```

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

/* PROGRAM PRZECHWYTUJE SYGNAŁY ZADANE PRZEZ PARAMETRY WYWOŁANIA
   W ilości nie większej niż 10 */

#define KOM "Błąd funkcji signal\n"
void Przechwycenie_Sygnału(int sygnal)
{
    char xx[256];
    sprintf(xx, "PID %d: Przechwycilem sygnal %d\n", getpid(), sygnal);
    write(STDERR_FILENO, xx, strlen(xx));

    /* W niektórych systemach tutaj należy powtorzyć wywołanie funkcji
       signal */
}

int main(int arg, char **arc)
{
    int sygnal[10], i;
    char xx[100];

    if (arg > 11) {
        sprintf(xx, "\n Proces %s należy wywołać z nie więcej niż 10
parametrami\n Ilość podanych parametrow: %d\n\n", arc[0], arg-1);
        write(STDOUT_FILENO, xx, strlen(xx));
        return(1); }

    for (i=0; i<arg-1; i++)
        sygnal[i]=atoi(arc[i+1]);

    sprintf(xx, "\n Proces %s\n Obsługiwane sygnały:", arc[0]);
    write(STDOUT_FILENO, xx, strlen(xx));
    for (i=0; i<arg-1; i++) {
        sprintf(xx, "%d ", sygnal[i]);
        write(STDOUT_FILENO, xx, strlen(xx)); }
    sprintf(xx, "\n");
    write(STDOUT_FILENO, xx, strlen(xx));

    for (i=0; i<arg-1; i++) {
        if (signal(sygnal[i], Przechwycenie_Sygnału) == SIG_ERR) {
            write(STDERR_FILENO, KOM, strlen(KOM));
            return(1); }
    }

    while (1)
    { ; /* tutaj można wstawić ciało procesu */
    }
}
```