

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Национальный исследовательский ядерный университет «МИФИ»  
(НИЯУ МИФИ)  
Институт интеллектуальных кибернетических систем  
Кафедра Кибернетики

**Лабораторная работа №2 по курсу  
«Разработка ПО ОС UNIX»**

**Выполнил студент группы Б17-511:**

Чудновец Иван

**Проверил:**

Ктитров С.В.

Москва, 2020

## Задание

Разработать программу, которая запускает задаваемую программу, возможно, с аргументами, а затем, в диалоге с пользователем, приостанавливает ее, возобновляет выполнение, прекращает её работу. Программу оформить как утилиту командной строки.

## Код программы

```
#include <unistd.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <signal.h>

void dialog(){

    printf("To stop the process, write the 's' symbol in the console.\n");

    printf("To continue the process, write the 'c' symbol in the console.\n");

    printf("To kill the process, write the 'k' symbol in the console.\n");

    printf("To exit from the program, write the 'e' symbol in the console.\n");

}

int main(int argc, char *argv[]) {

    pid_t pid = fork();

    if (pid == -1){

        printf("Error with starting process.\n");

        exit(-1);

    }

    if (pid == 0){

        int len = 0;

        while (argv[2 + len] != NULL) len++;

        char **args;

        int str_sizeof = sizeof(char *);

        if (len != 0){

            args = (char **)malloc((len + 2) * str_sizeof);

            int i = 1;
```

```

        for(; i <= len + 1; i++){
            args[i] = argv[1 + i];
        }
    }
    else{
        args = (char **)malloc(2 * str_sizeof);
        args[1] = NULL;
    }
    args[0] = argv[0];
    execv(argv[1], args);
}
else{
    char sym;
    dialog();
    while ((sym = getchar()) != 'e'){
        switch(sym){
            case '\n':
                continue;
            case 's':
                printf("SIGSTOP has emitted.\n");
                kill(pid, SIGSTOP);
                break;
            case 'c':
                printf("SIGCONT has emitted.\n");
                kill(pid, SIGCONT);
                break;
            case 'k':
                printf("SIGKILL has emitted.\n");
                kill(pid, SIGKILL);
                break;
            default:
                printf("Unexpected symbol '%c'\n", sym);

```

```
                break;
            }
            dialog();
        }
    }
    return 0;
}
```