

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

## ЛАБОРАТОРНАЯ РАБОТА №2

по курсу операционные системы

I семестр, 2019/20 уч. год

Студент Артемьев Дмитрий Иванович, группа М8О-206Б-18

Преподаватель Соколов Андрей Алексеевич

Оценка

Подпись

Дата

## Условие

23. Родительский процесс считывает две координаты передает их через канал дочернему процессу. Дочерний процесс определяет к какой четверти относится точка, а далее передает результат родительскому процессу.

## Описание программы

Код программы находится в файле main.c.

## Ход выполнения программы

1. Чтение координат в два массива `char`.
2. Передача координат в дочерний процесс.
3. Получение координат дочерним процессом.
4. Определение, к какой четверти относится точка, используя только информацию о ведущем символе ( '-' или число).
5. Передача результата в родительский процесс.
6. Вывод результата родительским процессом в стандартный поток вывода.
7. Завершение работы программы.

## Недочёты

Если на вход подавать лидирующие пробелы, то программа будет работать некорректно.

## Выводы

Я научился работать с процессами в Linux.

## Исходный код

### main.c

```
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <wait.h>

int main() {
    char x[9], y[9];
    char buffer[2] = {'\0'} ;
    int data_processed;
    int x_pipe[2];
    int y_pipe[2];
    int result[2];

    if (pipe(x_pipe) || pipe(y_pipe) || pipe(result)) {
        printf("Error pipe\n");
        exit(2);
    }

    pid_t proc = fork();
    if (proc < 0) {
        printf("Error fork\n");
        exit(1);
    }

    if (proc == 0) {

        close(y_pipe[1]);
        close(x_pipe[1]);

        data_processed = read(x_pipe[0], x, sizeof(x));
        data_processed = read(y_pipe[0], y, sizeof(y));

        close(y_pipe[0]);
        close(x_pipe[0]);

        if (x[0] != '-' && y[0] != '-') {
            buffer[0] = '1';
        }
    }
```

```

    if (x[0] == '-' && y[0] != '-') {
        buffer[0] = '2';
    }
    if (x[0] == '-' && y[0] == '-') {
        buffer[0] = '3';
    }
    if (x[0] != '-' && y[0] == '-') {
        buffer[0] = '4';
    }

    close(result[0]);
    data_processed = write(result[1], buffer, sizeof(buffer));
    close(result[1]);
    exit(EXIT_SUCCESS);
}
else if (proc > 0) {
    read(0, x, sizeof(x));
    read(0, y, sizeof(y));

    close(y_pipe[0]);
    close(x_pipe[0]);

    data_processed = write(x_pipe[1], x, sizeof(x));
    data_processed = write(y_pipe[1], y, sizeof(y));

    close(y_pipe[1]);
    close(x_pipe[1]);

    int how;
    wait(&how);
    close(result[1]);
    data_processed = read(result[0], buffer, sizeof(buffer));
    write(1, buffer, sizeof(buffer));
    close(result[0]);
    exit(EXIT_SUCCESS);
}
exit(EXIT_SUCCESS);
}

```