

# ОС ДЗ 4

Власов Николай Алексеевич, БПИ229

15 февраля 2024 г.

## Описание кода

### Сделал на 10 :)

В начале определяются две константы: размер буфера и стандартный мод для файлов (rw-rw-rw-)

В начале функции *main* мы проверяем, достаточно ли аргументов. Если нет, выводим подсказку.

Создаем буфер (маленький (+1 балл)), открываем файлы (попутно отлавливая ошибки), затем циклом *dowhile* считываем из входного файла данные в буфер и записываем их оттуда в выходной файл.

Затем получаем флаги для входного файла и присваиваем выходному файлу те же флаги. (+1 балл)

## Код программы

Листинг 1: main.c

```
1 #include <fcntl.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <sys/stat.h>
5 #include <unistd.h>
6
7 const int buffer_size = 256;
8 const int default_mode = 0666;
9
10
11 int main(int argc, const char *argv[]) {
12     if (argc < 3) {
13         // wrong arguments number
14         fprintf(stderr, "use %s<input>_<output>\t to write content from <input> to <output>\n", argv[0]);
15         exit(1);
16     }
17     char buffer[buffer_size];
18     // buffer of fixed little size
19     int input_fd = open(argv[1], O_RDONLY, default_mode);
20     // open input file
21     if (input_fd == -1) {
22         // error during opening input file
23         fprintf(stderr, "Error during opening %s\n", argv[1]);
24         exit(1);
25     }
26     int output_fd = open(argv[2], O_WRONLY | O_CREAT | O_TRUNC, default_mode);
27     // open output file
```

Листинг 2: main.c (продолжение)

```

1     ssize_t bytes_read = 0;
2     do {
3         bytes_read = read(input_fd, buffer, buffer_size);
4         // read bytes from file to buffer
5         if (bytes_read == -1) {
6             // error during reading input file
7             fprintf(stderr, "Error_during_reading_%s\n", argv[1]);
8             exit(1);
9         }
10        write(output_fd, buffer, bytes_read);
11        // write bytes to file to buffer
12        } while (bytes_read == buffer_size);
13
14    struct stat input_st;
15
16    if (stat(argv[1], &input_st) != 0) {
17        // error during checking flags of input file
18        fprintf(stderr, "Error_during_checking_flags_of_%s\n", argv[1]);
19        exit(1);
20    }
21
22    close(input_fd);
23    // close input file
24
25    if (fchmod(output_fd, input_st.st_mode) == -1) {
26        // error during change flag of output file
27        fprintf(stderr, "Error_during_changing_mode_of_%s\n", argv[2]);
28        exit(1);
29    }
30
31    close(output_fd);
32    // close output file
33 }

```