

**Министерство науки и высшего образования  
Российской Федерации**

**Федеральное государственное автономное  
образовательное учреждение высшего образования**

**«Национальный исследовательский университет  
ИТМО»**

**Факультет информационных технологий и  
программирования**

Лабораторная работа №15

*Архиватор файлов*

**Выполнила студентка группы № М3118**

Маркозубова Анастасия Кирилловна

**Подпись:**

**Проверил:**

Повышев Владислав Вячеславович

Санкт-Петербург  
2023

## Текст задания

Целью лабораторной работы является разработка программы по архивированию и распаковке нескольких файлов в один архив. Архиватор должен

1. Уметь архивировать несколько (один и более) указанных файлов в архив с расширением **\*.arc**
2. Уметь распаковывать файловых архив, извлекая изначально запакованные файлы
3. Предоставлять список файлов, упакованных в архиве
4. *Сжимать и разжимать данные при архивировании с помощью алгоритма Хаффмана (опциональное задание, оценивается доп баллами)*

Архиватор должен быть выполнен в виде консольного приложения, принимающего в качестве аргументов следующий параметры

- **--file FILE**  
Имя файлового архива, с которым будет работать архиватор
- **--create**  
Команда для создания файлового архива
- **--extract;**  
Команда для извлечения из файлового архива файлов
- **--list**  
Команда для предоставления списка файлов, хранящихся в архиве
- **FILE1 FILE2 .... FILEN**  
Свободные аргументы для передачи списка файлов для запаковки

Примеры использования:

***arc --file data.arc --create a.txt b.bin c.bmp***

***arc --file data.arc --extract***

***arc --file data.arc --list***

## Решение с комментариями

```
#include <stdio.h>
#include <string.h>

// Функция для открытия файла
FILE* openFile(const char* filename, const char* mode) {
    FILE* file = fopen(filename, mode);
    if (!file) {
        fprintf(stderr, "Failed to open file %s\n", filename);
    }
    return file;
}

// Функция для создания архива и добавления файлов в архив
void create(char* archiveName, int count, char* files[], int startIndex) {
    FILE* archiveFile = openFile(archiveName, "ab+");
    if (!archiveFile) return;

    for (int i = startIndex; i < count; i++) {
        FILE* file = openFile(files[i], "rb");
        if (!file) continue;

        // Получение размера файла
        fseek(file, 0, SEEK_END);
        unsigned long long filesize = ftell(file);
        fseek(file, 0, SEEK_SET);

        // Запись в архив имени файла и его размера
        fprintf(archiveFile, "%s %llu\n", files[i], filesize);

        // Копирование содержимого файла в архив
        for (unsigned long long j = 0; j < filesize; j++) {
            int ch = fgetc(file);
            if (ch == EOF) break;
            fputc(ch, archiveFile);
        }

        fclose(file);
    }

    fclose(archiveFile);
}

// Функция для извлечения файлов из архива
void extract(char* nameArchive) {
    FILE* arc = fopen(nameArchive, "rb");
    if (!arc) {
        fprintf(stderr, "Failed to open archive file for reading\n");
        return;
    }

    char filename[128];
    unsigned long long int filesize;

    // Чтение из архива имени файла и его размера
    while (fscanf(arc, "%127s %llu\n", filename, &filesize) == 2) {
        FILE* file = fopen(filename, "wb");
        if (!file) {
            fprintf(stderr, "Failed to open file for writing\n");
            continue;
        }

        // Извлечение данных файла из архива
        while (filesize-- > 0) {
            int ch = fgetc(arc);
```

```

        if (ch == EOF) break;
        fputc(ch, file);
    }

    fclose(file);
}

fclose(arc);
}

// Функция для отображения списка файлов в архиве
void list(char* nameArchive) {
    FILE* arc = fopen(nameArchive, "rb");
    if (!arc) {
        fprintf(stderr, "Failed to open archive file for reading\n");
        return;
    }

    char filename[128];
    unsigned long long int filesize;

    // Вывод на консоль названия файлов в архиве и их размера
    while (fscanf(arc, "%127s %llu\n", filename, &filesize) == 2) {
        printf("%s size: %llu bytes\n", filename, filesize);

        // Пропуск содержимого файла
        fseek(arc, filesize, SEEK_CUR);

        // Пропуск пустых строк
        int ch = fgetc(arc);
        while (ch == '\n' || ch == '\r') {
            ch = fgetc(arc);
        }

        // Возврат символа обратно в поток, если это не конец файла
        if (ch != EOF) {
            ungetc(ch, arc);
        }
    }

    fclose(arc);
}

int main(int argc, char* argv[]) {
    char* nameArchive = NULL;
    int create_flag = 0, extract_flag = 0, list_flag = 0;
    int files_start = 0; // Индекс начала списка файлов для архивации
    int files_end = argc; // Индекс конца списка файлов

    // Обработка аргументов командной строки
    for (int i = 1; i < argc; i++) {
        if (strcmp(argv[i], "--file") == 0 && i + 1 < argc) {
            nameArchive = argv[++i];
            files_end = i - 1; // Конец списка файлов до "--file"
        }
        else if (strcmp(argv[i], "--create") == 0) {
            create_flag = 1;
            files_start = i + 1; // Начало списка файлов после "--create"
        }
        else if (strcmp(argv[i], "--extract") == 0) {
            extract_flag = 1;
        }
        else if (strcmp(argv[i], "--list") == 0) {
            list_flag = 1;
        }
    }

```

```

    }

    // Выполнение соответствующей операции в зависимости от флагов
    if (nameArchive) {
        if (create_flag) {
            create(nameArchive, files_end, argv, files_start);
        }
        else if (extract_flag) {
            extract(nameArchive);
        }
        else if (list_flag) {
            list(nameArchive);
        }
    }
    else {
        fprintf(stderr, "Archive file name not specified\n");
    }

    return 0;
}

```









```
D:\OP\LABA 15>main --create test.txt test2.txt test3.txt --file archive.arc
```

```
D:\OP\LABA 15>main --file archive.arc --list
test.txt size: 4 bytes
test2.txt size: 4 bytes
test3.txt size: 4 bytes
```

```
D:\OP\LABA 15>main --create abc.bmp --file archive.arc
```

```
D:\OP\LABA 15>main --file archive.arc --list
test.txt size: 4 bytes
test2.txt size: 4 bytes
test3.txt size: 4 bytes
abc.bmp size: 0 bytes
```

```
D:\OP\LABA 15>main --extract --file archive.arc
```

 abc.bmp	20.12.2023 3:28	Файл "BMP"	0 КБ
 CMakeLists.txt	10.12.2023 20:27	Текстовый документ	1 КБ
 main.c	20.12.2023 3:25	JetBrains CLion	11 КБ
 test.txt	20.12.2023 3:28	Текстовый документ	1 КБ
 test2.txt	20.12.2023 3:28	Текстовый документ	1 КБ
 test3.txt	20.12.2023 3:28	Текстовый документ	1 КБ
 main.exe	20.12.2023 3:25	Приложение	124 КБ
 archive.arc	20.12.2023 3:27	Файл "ARC"	1 КБ