Министр 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

Выполнил студент группы № M3119
Черных Арсений Игоревич
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
Программа состоит из 3 основных функций. Создание архива, извлечение файлов, вывод списка файлов. В createArchive создаю архив. Для каждого из файлов: открываем файл, записываем размер, длину, считываем файл в буфер и записываем в архив. Extract функция извлечения из архива. В функции лист, для каждого файла в архиве проводим операции: считывание всех параметров, перемещение указателя в архиве на след файл. В Мейне работаем с командной строкой, просто вызываем эти функции и обрабатываем ошибки, если есть.

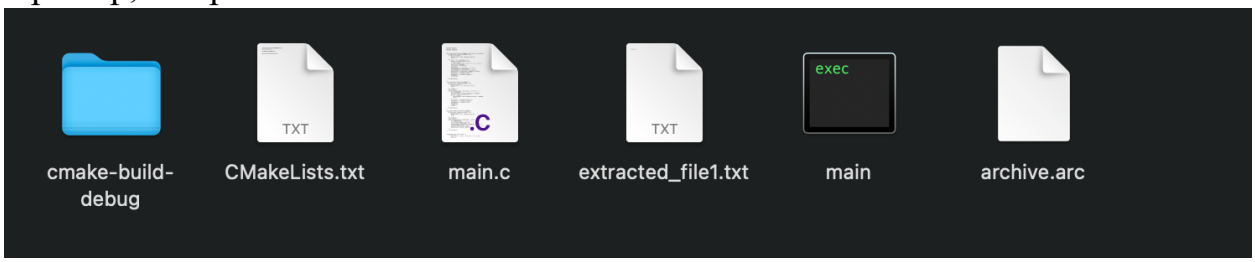
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>


void createArchive(char* archiveName, char* files[], int numFiles) {
    FILE* archive = fopen(archiveName, "wb");
    if (archive == NULL) {
        fprintf(stderr, "Error opening archive\n");
        return;
    }
    for (int i = 0; i < numFiles; i++) {
        FILE* file = fopen(files[i], "rb");
        if (file == NULL)
            fprintf(stderr,"Error opening file %s\n", files[i]);
        fseek(file, 0, SEEK_END);
        long sizeFile = ftell(file);
        rewind(file);
        fwrite(&sizeFile, sizeof(long), 1, archive);
        int lengthFile = strlen(files[i]);
        fwrite(&lengthFile, sizeof(int), 1, archive);
        fwrite(files[i], sizeof(char), lengthFile, archive);
        char* buffer = (char*)malloc(sizeFile);
        fread(buffer, 1, sizeFile, file);
        fwrite(buffer, 1, sizeFile, archive);
        free(buffer);
        fclose(file);
    }
    fclose(archive);
}
```

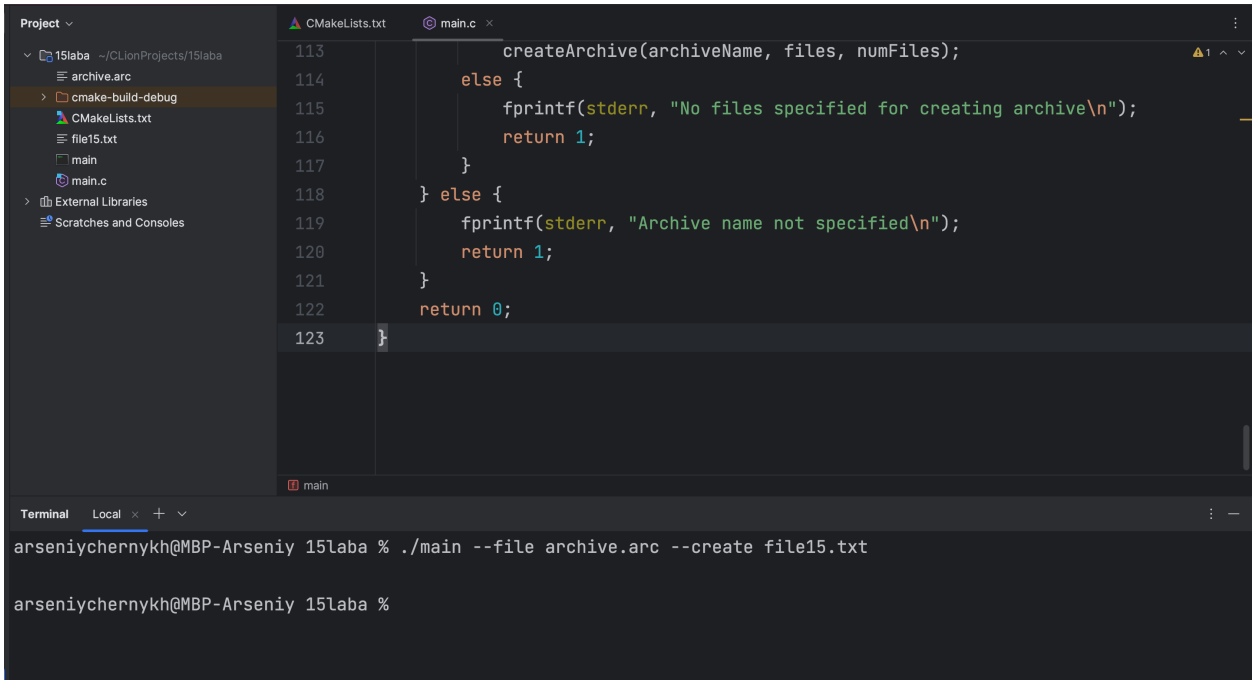
```
void extractArchive(char* archiveName) {
    FILE* archive = fopen(archiveName, "rb");
    if (archive == NULL) {
        fprintf(stderr, "Error opening archive\n");
        return;
    }
    int fileNum = 1;
    long sizeFile;
    while (fread(&sizeFile, sizeof(long), 1, archive) == 1) {
        char filename[50];
        sprintf(filename, "extracted_file%d.txt", fileNum);
        FILE* file = fopen(filename, "wb");
        if (file == NULL) {
            fprintf(stderr, "Error creating file %s\n", filename);
            return;
        }
        char* buffer = (char*)malloc(sizeFile);
        fread(buffer, 1, sizeFile, archive);
        fwrite(buffer, 1, sizeFile, file);
        free(buffer);
        fclose(file);
        fileNum++;
    }
    fclose(archive);
}

void listFilesInArchive(char* archiveName) {
    FILE* archive = fopen(archiveName, "rb");
    if (archive == NULL) {
        fprintf(stderr, "Error opening archive\n");
        return;
    }
    int fileNum = 1;
    long sizeFile;
    while (fread(&sizeFile, sizeof(long), 1, archive) == 1) {
        int fileNameLength;
        fread(&fileNameLength, sizeof(int), 1, archive);
        char filename[fileNameLength + 1];
        fread(filename, sizeof(char), fileNameLength, archive);
        filename[fileNameLength] = '\0';
        printf("File %d: %s, size %ld bytes\n", fileNum++, filename, sizeFile);
```

```
        fseek(archive, sizeFile, SEEK_CUR);
    }
    fclose(archive);
}


int main(int argc, char* argv[]) {
    if (argc < 4) {
        fprintf(stderr, "ERROR, the program is not working");
        return 1;
    }

    char* archiveName = NULL;
    char* files[100];
    int numFiles = 0;
    int extractingArchive = 0;
    int listingFiles = 0;
    for (int i = 1; i < argc; i++) {
        if (strcmp(argv[i], "--file") == 0) {
            archiveName = argv[i+1];
            i++;
        }
        else if (strcmp(argv[i], "--create") == 0) {
            for (int j = i+1; j < argc; j++) {
                if (argv[j][0] != '-')
                    files[numFiles++] = argv[j];
                else break;
            }
        }
        else if (strcmp(argv[i], "--extract") == 0)
            extractingArchive = 1;
        else if (strcmp(argv[i], "--list") == 0)
            listingFiles = 1;
    }
    if (archiveName) {
        if (extractingArchive)
            extractArchive(archiveName);
        else if (listingFiles)
            listFilesInArchive(archiveName);
        else if (numFiles > 0)
            createArchive(archiveName, files, numFiles);
        else {
```

```
            fprintf(stderr, "No files specified for creating archive\n");
            return 1;
        }
    } else {
        fprintf(stderr, "Archive name not specified\n");
        return 1;
    }
    return 0;
}
```

Пример, как работает extract





Пример работы создания архива