**Министр науки и высшего образования Российской̆ Федерации**

**Федеральное государственное автономное образовательное учреждение высшего образования**

**«Национальный исследовательский университет ИТМО»**

Факультет информационных технологий и программирования

Лабораторная работа №14

**Выполнил студент группы № M3104**

Гурман Тимофей Владимирович

**Подпись:**

****

**Проверил:**

Повышев Владислав Вячеславович

Санкт-Петербург

2020Текст 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 запаковонные

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

Arch.h

#ifndef LAB15\_ARCH\_H  
#define LAB15\_ARCH\_H  
void Extract(char\* fileName);  
void List(char\* file\_name);  
void Create(char\* fileName, int argc, char\* argv[]);  
#endif //LAB15\_ARCH\_H

Arch.c

#include "arch.h"  
#include <stdio.h>  
  
void Extract(char\* fileName) {  
 FILE\* f\_Arc = fopen(fileName, "rb");  
 unsigned long long int now = 0;  
 unsigned long long int start = 0;  
 int chr;  
 while ((chr = getc(f\_Arc)) != EOF) {  
 start++;  
 if (chr == '\n')  
 break;  
 }  
  
 fseek(f\_Arc, 0, SEEK\_SET);  
 char name[128] = { 0 };  
 unsigned long long int sizef;  
 FILE\* f;  
 while (fscanf(f\_Arc, "file: %s size: %llu ", name, &sizef) != 0) {  
 f = fopen(name, "wb");  
 if (f == NULL)  
 break;  
 now = ftell(f\_Arc);  
 fseek(f\_Arc, start, SEEK\_SET);  
 start += sizef;  
 while ((sizef--) > 0) {  
 chr = getc(f\_Arc);  
 putc(chr, f);  
 }  
 fseek(f\_Arc, now, SEEK\_SET);  
 fclose(f);  
 }  
 printf("extraction completed");  
}  
  
void List(char\* file\_name) {  
 FILE\* f\_Arc = fopen(file\_name, "rb");  
 char name[128];  
 int i;  
 while (1) {  
 fscanf(f\_Arc, "%s", name);  
 if (name[0] == '!')  
 break;  
 fscanf(f\_Arc, "%s", name);  
 printf("%s\n", name);  
 fscanf(f\_Arc, "%s", name);  
 fscanf(f\_Arc, "%s", name);  
 }  
 fclose(f\_Arc);  
}  
  
void Create(char\* fileName, int argc, char\* argv[]) {  
 int i;  
 int tmp;  
 FILE\* f\_Arc = fopen(fileName, "wb");  
 FILE\* f;  
 unsigned long long int nameSize[128];  
 for (i = 4; i < argc; i++) {  
 f = fopen(argv[i], "rb");  
 if (f == NULL) {  
 continue;  
 }  
 fseek(f, 0, SEEK\_END);  
 nameSize[i - 4] = ftell(f);  
 fclose(f);  
 }  
 for (i = 0; i < argc - 4; i++) {  
 fprintf(f\_Arc, "file: %s size: %llu ", argv[i + 4], nameSize[i]);  
 }  
 fprintf(f\_Arc, "!\n");  
 for (i = 4; i < argc; i++) {  
 f = fopen(argv[i], "rb");  
 if (f == NULL) {  
 printf("Open error %s\n", argv[i]);  
 continue;  
 }  
 while ((tmp = getc(f)) != EOF)  
 putc(tmp, f\_Arc);  
 fclose(f);  
 }  
 printf("archivation finished");  
}

Main.c

#include <stdio.h>  
#include "arch.h"  
  
int main(int argc, char\* argv[]) {  
 int i;  
 char\* fileName;  
 printf("\n");  
 for (i = 0; i < argc; i++) {  
 if (argv[i][2] == 'f') {  
 fileName = argv[i + 1];  
 }  
 if (argv[i][2] == 'c') {  
 Create(fileName, argc, argv);  
 break;  
 }  
 if (argv[i][2] == 'e') {  
 Extract(fileName);  
 }  
 if (argv[i][2] == 'l') {  
 List(fileName);  
 }  
 }  
  
 return 0;  
}