Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Курсовая работа по курсу «Дискретный анализ»: Методы сжатия данных

Студент: О. Р. Лисовский
Преподаватель: Н. А. Зацепин
Группа: М8О-408Б
Дата:
Оценка:
Подпись:

Москва, 2020

## Условие

Необходимо реализовать два известных метода сжатия данных для сжатия одного файла.

Формат запуска должен быть аналогичен формату запуска программы gzip, должны быть поддержаны следующие ключи: -c, -d, -k, -l, -r, -t -1 -9. Должно поддерживаться указание символа дефиса в качестве стандартного ввода.

## Метод решения

Как и требуется в условии запуск программы аналогичен запуску утилиты gzip: ./main <ключи> <файлы> <ключи> <файлы> ...

### Обработка входных данных

Программа начинается с обработки строки стандартного ввода. Строка обрабатывается по словам. Если слово начинается с символа «-», то предполагается что это набор ключей и ключи передаются в специальную функцию, чтобы исключить противоречия работы ключей. Возможные сочетания и глвенство одних ключей над другими описано в таблице ниже.

| | d | k | l | r | t | 1 | 9 |
|---|---|---|---|---|---|---|---|
| c | нет блока | c блокирует k | l блокирует c | нет блока | t блокирует c | нет блока | нет блока |
| d | - | нет блока | l блокирует d | нет блока | t блокирует d | d блокирует 1 | d блокирует 9 |
| k | - | - | l блокирует k | нет блока | t блокирует k | нет блока | нет блока |
| l | - | - | - | нет блока | l блокирует t | l блокирует 1 | l блокирует 9 |
| r | - | - | - | - | нет блока | нет блока | нет блока |
| t | - | - | - | - | - | t блокирует 1 | t блокирует 9 |
| 1 | - | - | - | - | - | - | последний полученный блокирует прошлые |

В случае если полученное слово начинается с другого символа, то программа предполагает что это имя файла или директории и добавляет его в список для дальнейшей обработки.

### Интерфейс

После установления активных ключей и заполнения списка объектов компрессии/-декомпрессии, программа начинает работу этим самым списком. В случае отсутствия ключа -r все директории не рассматриваются. При активации ключа всё содержимое директории рекурсивно обрабатывается программой. При работе с файлами проверяется наличие/отсутствие (в зависимости от ключа -d) файла с расширением .gz и в случае необходимости программа спрашивает у пользователя право на перезапись соответствующего файла.

При подготовке непосредственно компрессии проверяется наличие ключа -1 или -9 для определения необходимого алгоритма. В случае их отсутствия используются оба алгоритма и выбирается лучший результат.

При подготовке непосредственно декомпрессии читается первый байт файла для установления алгоритма декодирования.

### Постобработка

При окончании работы компрессии/декомпрессии программа получает сигнал об их завершении. Если этот сигнал соответствует ошибке то работа с конкретным файлом аварийно прекращается и обрабатывается следующий файл. Дальнейшие действия обусловлены введёнными ключами.

Далее незакодированный файл будет упоминаться как файл, а закодированный файл как архив.

### Арифметическая компрессия

### Арифметическая декомпрессия

### LZ77 компрессия

### LZ77 декомпрессия

По окончании чтения архива, количество байт, которое было в изначальном файле, сверяется с тем, сколько было записано в его новую версию. При несовпадении выводится соответствующее сообщение, и декомпрессия завершается неудачно.

## Описание файлов программы

Код программы разбит на 9 файлов:

1. Algorithms.h - Содержит базовую информацию о классах TACC и TLZ77, необходимых для работы компрессии и декомпрессии соответствующих алгоритмов.

2. Algorithms.cpp - Содержит реализацию классов TACC и TLZ77.

3. BFile.h - Содержит базовую информацию о классах TOutBinary и класса TInBinary, необходимых для работы с файлами.

4. BFile.cpp - Содержит реализацию классов TOutBinary и TInBinary.

5. interface.h - Содержит в себе перечисление и описание всех функций необходимых для взаимодействия программы и алгоритмов сжатия данных.

6. interface.cpp - Содержит реализацию всех функций, описанных в файле interface.h.

7. Library.h - Содержит в себе ключи, необходимые для работы алгоритмов, и библиотеки для работы всей программы.

8. main.cpp - Файл запуска.

9. Makefile - Сборочный файл.

## Основные типы данных

1. TOutBinary - класс, обеспечивающий запись необходимого количества байт в файл.

2. TInBinary - класс обеспечивающий считывание необходимого количества байт из файла.

3. TLZ77 - класс, описывающий работу алгоритма LZ77.

4. TACC - класс, описывающий работу арифметического алгоритма.

## Описание методов и функций программы

### Основные свойства и методы класса TACC

public:

1.

2.

3.

4.

5.

6.

7.

8.

private:

1.
2.
3.
4.
5.
6.
7.
8.

**Основные свойства и методы класса TOutBinary**

public:

1.
2.
3.
4.
5.
6.
7.
8.

private:

1.
2.
3.
4.
5.
6.
7.
8.

**Основные свойства и методы класса TInBinary**

public:

1.
2.
3.
4.
5.
6.
7.
8.

private:

1.
2.
3.
4.
5.
6.
7.
8.

**Основные свойства и методы класса TLZ77**

public:

1.
2.
3.
4.

5.

6.

7.

8.

private:

1.

2.

3.

4.

5.

6.

7.

8.

**Прочие функции**

1. void FileIterator(std::map<std::string, int>) - Осуществляет проход по всем папкам и файлам для их компрессии/декомпрессии.
2. bool Parser(std::map<std::string, int>*, std::string) - фильтрует полученные при вводе аргументы. При получении некорректного аргумента возвращает false.
3. bool AskDir(std::string, bool) - Проверка на существование директории. При существовании возвращает true, в любом ином случае false.
4. void DirectoryWork(std::string) - В случае наличия ключа -r осуществляет работу с внутренними файлами и директориями указанной директории.
5. void DeComPress(std::string) - Помогает определить действия по отношению к указанному файлу: совершить компрессию, декомпрессию или посмотреть информацию об архиве.
6. bool Rewrite(std::string) - В случае возможного повторения имён файлов при компрессии/декомпрессии принимает решение о перезаписи.

7. void ErrorNotes(std::string) - Показывает сообщения об ошибках, возникших при работе с указанной директорией.

8. bool KeyL(TInBinary*, std::string) - Осуществляет работу ключа -l - вывод информации об архиве.

9. void PreCompress(TInBinary*, std::string) - Осуществляет подготовку указанного файла к сжатию в соответствии с указанными ключами.

10. unsigned long long int Compress(std::string, TInBinary*, bool) - Непосредственно активирует указанный алгоритм сжатия. Возвращает размер полученного архива или 0 в случае ошибки.

11. void PreDecompress(TInBinary*, std::string) - Осуществляет подготовку указанного файла к разжатию в соответствии с указанными ключами.

## Исходный код

## Тест производительности

| Файл | Размер исходного файла | Алгоритм | Время сжатия (с) | Время декомпрессии (с) | Размер сжатого файла | Коэффициент сжатия |
|---|---|---|---|---|---|---|
| world95.txt | | | | | | |
| world95.txt | | | | | | |
| world95.txt | | | | | | |
| enwik8 | | | | | | |
| enwik8 | | | | | | |
| enwik8 | | | | | | |
| enwik9 | | | | | | |
| enwik9 | | | | | | |
| enwik9 | | | | | | |

## Выводы

В процессе выполнения данной работы я освоил 2 вида кодирования: арифметическое и LZ77. Было обнаружены как сходства, так и различия. К примеру

Благодаря освоению двух алгоритмов сразу у меня появились представления о рабоче прочих алгоритмов кодирования и стали очевидны различные требования к их работе и результату. Были существенно улучшены навыки работы с файлами: проверка наличия, запись, чтение, перепись.

## Список литературы


1. Алгоритм LZ77 [Электронный ресурс]: mf.grsu.by URL: http://mf.grsu.by/UchProc/livak/po/comprsite/theory_lz77.html (дата обращения 10.08.2020)

2. Алгоритмы LZW, LZ77 и LZ78 [Электронный ресурс]: habr.com URL: https://habr.com/ru/post/132683/ (дата обращения 23.08.2020)

3. Арифметическое кодирование [Электронный ресурс]: mf.grsu.by URL: http://mf.grsu.by/UchProc/livak/po/comprsite/theory_arithmetic.html (дата обращения 30.08.2020)

4. Идея арифметического кодирования [Электронный ресурс]: algolist.ru URL: http://algolist.ru/compress/standard/arithm.php (дата обращения 02.09.2020)

5. Arithmetic coding - integer implementation [Электронный ресурс]: stringology.org URL: http://www.stringology.org/DataCompression/ak-int/index_en.html (дата обращения 26.09.2020)