

Barch – “Boldir archive program”.

Оглавление

Описание.....	2
Детальное описание формата файла архива.....	2
Алгоритм сжатия Хаффмана.....	3
Вызов программы из консоли.....	3
Добавления файлов в архив.....	3
Удаление файлов из архива.....	3
Извлечение файлов из архива.....	4
Целостность архива.....	4
Несовпадение хеш-кода файла.....	4
Вывод информации о файлах, хранящихся в архиве.....	4
Заметки.....	4

Описание.

Barch — архиватор с функцией сжатия файлов по алгоритму Хаффмана.

Архив состоит из заголовка и файлов, которые в свою очередь также содержит заголовок. Для обеспечения целостности архива содержит хеш-код, как всех файлов вместе, так и каждого файла отдельно. Сжатию подвергается каждый файл отдельно, поэтому, если пропала какая-то часть данных, остальные от этого не пострадают и добавление нового файла не приводит к изменению всего архива.

После выполнения программы, на экран будет напечатан список ошибок, возникших в ходе работы программы.

Детальное описание формата файла архива.

Таблица 1. Заголовок архива.

Название	Тип	Размер (байт)	Описание
magic	Строка	6	6 символов для распознавания того, что это архив Barch. Всегда равно «BARCH1».
size	Целое	4	Размер данных после заголовка.
numObjects	Целое	4	Кол-во файлов.
hash	Целое	4	Хеш-код всех данных после заголовка (алгоритм crc32).

После заголовка архива идут numObjects секций, каждая из которых состоит из своего заголовка (называемым заголовком файла) и сжатого данных (файла).

Таблица 2. Заголовок файла.

Название	Тип	Размер (байт)	Описание
nameLength	Целое	4	Кол-во символов в имени файла.
name	Строка	nameLength	Имя файла (включая путь), без завершающего нуля.
size	Целое	4	Размер данных (сжатого файла)
user	Целое	4	Имя пользователя, указанное в правах файла.
group	Целое	4	Группа, указанная в правах файла.
permissions	Целое	4	Права файла в виде трех цифр, например, 777.
hash	Целое	4	Хеш-код данных (сжатого файла) после (алгоритм crc32).

Все целые значения являются без знаковыми.

[TODO: Хранение дерева]

Алгоритм сжатия Хаффмана.

Сжимая файл по алгоритму Хаффмана первое что мы должны сделать - это необходимо прочитать файл полностью и подсчитать сколько раз встречается каждый символ из расширенного набора ASCII. Если мы будем учитывать все 256 символов, то для нас не будет разницы в сжатии текстового и EXE файла.

После подсчета частоты вхождения каждого символа, необходимо просмотреть таблицу кодов ASCII и сформировать мнимую компоновку между кодами по убыванию. То есть не меняя местонахождение каждого символа из таблицы в памяти отсортировать таблицу ссылок на них по убыванию. Каждую ссылку из последней таблицы назовем "узлом".

[TODO]

Вызов программы из консоли.

Для вызова программы из консоли вводите имя программы (я компилирую в barch.exe), а затем дополнительные флаги. Формат вызова:

```
barch.exe -[axdlt] archive.barch file_1 file_2 ... file_n
```

Поддерживаемые операции:

- **a** - поместить файл(ы) в архив;
- **x** - извлечь файл(ы) из архива;
- **d** - удалить файл(ы) из архива;
- **l** - вывести информацию о файлах, хранящихся в архиве;
- **t** - проверить целостность архива.

Добавления файлов в архив.

Если архив уже существует, то в память грузиться всё содержимое заголовков. И создается копия оригинального архива. Если при чтении произошла ошибка, например, несовпадения хеш-кода, то выполнение программы будет приостановлено, в ожидание ответа пользователя, необходимо ли продолжать работу. Если же архива не существует, то в памяти компьютера создается пустая структура и заполняется нулями. Далее для каждого файла, который пользователь хочет добавить в архив: файл грузиться в память, сжимается по алгоритму Хаффмана и в копию архива заносится информация о файле и его содержимое, включая дерево. Если файла не существует, то есть программа его не может считать, то он пропускается, и в стек ошибок заносится сообщение об этом. После выполнения этих действий для всех файлов, создается новый архив, который заменит старый.

Удаление файлов из архива.

Первым делом считывается архив, если его не существует, то выводится программа заканчивает выполнение. Если при чтении произошла ошибка, например, несовпадения хеш-кода, то выполнение программы будет приостановлено, в ожидание ответа пользователя, необходимо ли продолжать работу. Далее для каждого введенного файла: если в архиве нету файла, то в стек ошибок заносится сообщение об этом. Иначе, из памяти удаляется вся информация о существовании данного файла. После выполнения этих действий для всех введенных файлов, будет создан новый архив, который заменит старый.

Извлечение файлов из архива.

Первым делом считывается архив, если его не существует, то выводится программа заканчивает выполнение. Если при чтении произошла ошибка, например, несовпадения хеш-кода, то выполнение программы будет приостановлено, в ожидание ответа пользователя, необходимо ли продолжать работу. Далее для каждого введенного файла, он декодируется и записывается в файловую систему. Если пользователь не ввел ни одного файла, то извлекутся все файлы, содержащиеся в архиве.

Целостность архива.

Проверяется хеш-код массива структур файлов и хеш-код, считанный из заголовка архива, а также хеш-код каждой отдельной структуры файла и хеш-код, считанный из заголовков файлов.

Несовпадение хеш-кода файла.

При несовпадении хеш-кода файла, пользователь получит вопрос, необходимо ли продолжать работу приложения. Варианты ответа:

- «n» - программа закончит выполнение.
- «y» - в ходе выполнения программы файлы будут учитываться так, как есть.
- «i» - удаление «плохих» файлов.

Вывод информации о файлах, хранящихся в архиве.

На экран пользователя выводится список файлов, содержащиеся в архиве. Файлы, у которых нет совпадения хеш-кода, помечаются восклицательным знаком.

Заметки.

Файлы, содержащиеся в архиве, никогда не загружаются в память, за исключением тех, над которыми необходимо провести какие-либо операции.