УДК 621.327:629.391

Студ. К. Д. Цыбулько
Науч. рук. доц. Н. В. Пацей
(кафедра программной инженерии, БГТУ)

АНАЛИЗ АЛГОРИТМОВ СЖАТИЯ ИНФОРМАЦИИ

Сжатие данных - это процесс, который используется для уменьшения физического размера блока информации. Задача сжатия состоит из двух компонентов: алгоритма кодирования, который принимает сообщение и генерирует сжатое представление (возможно, с меньшим количеством битов), и алгоритма декодирования, который восстанавливает исходное сообщение или некоторую его аппроксимацию из сжатого представления. Алгоритмы сжатия делятся на два класса − алгоритмы сжатия без потерь и алгоритмы с потерями. В данной работе были рассмотрены три часто используемых метода: кодирование Хаффмана, Лемпеля-Зива и RunLength.

Целью исследования является сравнение методов сжатия данных без потерь, чтобы установить следующее:

- степень (коэффициент) сжатия, которая может быть достигнута с помощью каждого из алгоритмов;
- уровень эффективности различных алгоритмов;
- сравнение методов по типу данных.

Метод кодирования Хаффмана назначает более короткие коды символам, которые встречаются чаще, а более длинные коды - тем, которые встречаются реже. Прежде чем мы сможем назначить битовые комбинации для каждого символа, мы назначаем каждому символу вес в зависимости от частоты его использования. Кодирование Хаффмана использует специальный метод выбора представления для каждого символа, в результате чего получается код префикса (иногда называемый «кодами без префикса»), то есть строка битов, представляющая некоторый конкретный символ. Для набора символов с равномерным распределением вероятностей и числом членов, которое является степенью двойки, кодирование Хаффмана эквивалентно простому двоичному блочному кодированию, например, кодированию ASCII.

Техника сжатия Run-Length, вероятно, является наиболее простым методом сжатия. Он может использоваться для сжатия данных, составленных из любой комбинации символов, и может быть достаточно эффективным, если данные представлены в виде 0 и 1. Общая идея этого метода состоит в том, чтобы заменить последовательные повторяющиеся вхождения символа одним вхождением символа, за которым следует количество вхождений.

Метод сжатия Lempel-Ziv (LZ) является одними из самых популярных алгоритмов без потерь. Код, который выводит алгоритм LZ,

может иметь любую произвольную длину, но должен содержать больше одного символа. Первые 256 кодов (при использовании восьми битных символов) по умолчанию назначаются стандартному набору символов. Остальные коды присваиваются строкам в процессе работы алгоритма.

LZ-кодирование относится к категории алгоритмов, называемых словарным кодированием. Идея состоит в том, чтобы создать словарь (таблицу) строк, используемых во время сеанса связи. Если и отправитель, и получатель имеют копию словаря, то ранее встреченные строки можно заменить их индексом в словаре, чтобы уменьшить объем передаваемой информации. Есть два одновременных события: создание индексированного словаря и сжатие строки символов. Алгоритм извлекает наименьшую подстроку, которую невозможно найти в словаре, из оставшейся несжатой строки. Затем он сохраняет копию этой подстроки в словаре в качестве новой записи и присваивает ей значение индекса. Сжатие происходит, когда подстрока, за исключением последнего символа, заменяется индексом, найденным в словаре. Программа для архивирования/разархивирования была разработана с использованием delphi. Данные, которые использовались для тестирования трех (3) сравниваемых методов сжатия: текстовый документ, документ Microsoft Word, веб-документ и изображение.

После запуска тестов были получены данные по коэффициентам сжатия, представленные на рисунке 1.

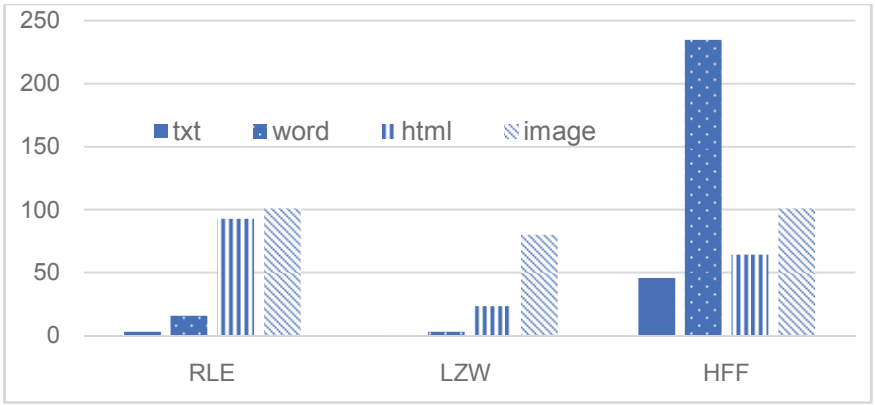

Рисунок 1 - Исследование коэффициента сжатия

Как видно из диаграммы для сжатия тестовых документов лучше подходит RLE алгоритм. Для сжатия документов в формате doc, а также html наименьший коэффициент сжатия дает алгоритм LZW. Для сжатия изображений исследуемые алгоритмы мало эффективны, так минимальны коэффициент сжатия -80% был получен при использовании алгоритма LZW.

Таким образом, метод Lempel-Ziv оказался наиболее эффективным и дает хорошие результаты для текстовых и веб-документов.