

Ю. В. Муругова

Научный руководитель: доцент Ю. А. Подгорнова
*Муромский институт (филиал) федерального государственного образовательного
учреждения высшего образования «Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
602264, г. Муром, Владимирская обл., ул. Орловская, 23*

Кодирование данных методом LZW

Сжатие данных — процесс преобразования данных посредством алгоритма, служащий для уменьшения их объема, а также для более эффективного использования устройств, передачи и хранения данных. Этот процесс необходим для устранения избыточности, которая содержится в исходных данных. Восстановление данных является процедурой обратной сжатия данных. Так же ее называют распаковкой или декомпрессией.

LZW-сжатие можно назвать наиболее универсальным методом сжатия данных. Он получил распространение благодаря своей простоте и гибкости. Этот алгоритм назван по имени его создателей Абрахама Лемпеля, Якоба Зива и Терри Велча [1].

Исходный метод сжатия данных впервые был заявлен в 1978 г. Над ним работали только двое ученых, поэтому ему присвоили название LZ78 (Lempel-Ziv). Позже, алгоритм был усовершенствован Терри Велчем, и опубликовал уже в 1984 г., как LZW-алгоритм.

LZW-сжатие производят для обработки файлов изображения в формате GIF, а также предлагается в качестве опции для форматов TIFF и PostScript.

Процесс сжатия состоит в следующем: алгоритмом динамически создаётся таблица, которая будет содержать преобразованные строки. Определённым символьным последовательностям ставятся соответствующие им группы бит имеющие определенную длину. Обычно, она является фиксированной и составляет 12 бит.

Происходит инициализация таблицы всеми одно-символьными строками. Алгоритм, во время кодирования, просматривает каждый символ текста, сохраняя каждую новую, уникальную двух-символьную строку в таблицу. Каждая строка представляет собой пару: код-символ, где код является ссылкой на соответствующий первый символ.

После сохранения в таблицу новой двух-символьной строки, происходит передача на выход кода первого символа. В процессе считывания, для очередного символа на входе, по таблице сопоставляется уже существующая строка с максимальной длиной бит, затем в таблицу записывается код этой строки и следующий символом на входе. В результате код этой строки выдается на выход, а в качестве начала новой применяется следующий символ.

Исходя из того, что любое гнездо словаря содержащее одно-символьные цепочки, исключая самые первые, содержит так же и копию некоторого другого гнезда, с одним символом, приписанным в конце, при реализации этого алгоритма на практике, существует возможность обойтись простой списочной структурой, имеющей одну связь.

Рассмотрим процесс сжатия, который выглядит следующим образом:

- 1) происходит последовательное считывание символов входного потока;
- 2) проверка на существование необходимой строки в созданной;
- 3) в случае ее существования, считывается следующий символ, в обратном случае в поток заносится код созданный для предыдущей найденной строки;
- 4) строка сохраняется в таблицу, а поиск начинается снова [2].

Для осуществления процесса декодирования, алгоритму на входе достаточно иметь лишь закодированный текст, так как существует возможность воссоздания необходимой таблицы преобразования опираясь на закодированный текст. Во время работы алгоритм генерирует однозначно декодируемый код. Это объясняется тем, что каждый раз, во время генерирования нового кода и добавлении строки в таблицу строк, LZW-алгоритм выполняет проверку, уточняя является ли строка уже известной. Если это так, то алгоритм не генерирует новый код, а выводит существующий. В результате, любая строка в таблице будет существовать строго в единственном экземпляре и иметь свой уникальный номер. Таким образом, при получении

нового кода, при дешифровании создается новая строка, а при получении уже известного, строка извлекается из словаря.

Преимущества LZW-алгоритма: высокая скорость работы при осуществлении упаковки или распаковки, требования малого объема памяти и сравнительно не сложная реализация на аппаратном уровне. Недостатки: степень сжатия информации значительно ниже, чем сжатие по схеме двухступенчатого кодирования. Это объясняется тем, что при переполнении словаря алгоритм может либо прекратить его заполнение, либо очистить (полностью или частично).

Литература

1. Ватолин Д. – Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. - М.: ДИАЛОГ-МИФИ, 2002. -384 с.
2. Материалы сайта habrahabr.ru //URL: <https://habrahabr.ru> (дата обращения: 30.03.2018)