**Сжатие информации**

Зачем нужно сжимать информацию и какие существуют способы это сделать.

А действительно, зачем? Посчитаем, к примеру, сколько займет памяти изображение, по качеству близкое к телевизионному. Пусть его разрешение -- 800х6009 пиксел, а число оттенков цвета около 16 тысяч (High Color), т. е. цвет каждого пиксела представляется двухбайтовым кодом. 800x600=480000 элементов. 480000x2 байт = 960000 байт -- это чуть меньше 1 мегабайта. Кажется, не так много -- на лазерном диске поместится больше 650 таких картинок. Ну, а если речь идет о фильме? Стандартная скорость кинопроекции -- 24 кадра в секунду. Значит на компакт-диске можно записать фрагмент длительностью 650:24=27 секунд. Куда это годится?! А ведь это далеко не единственный случай, когда информации "слишком много". Таким образом, одна из причин использования сжатия данных -- желание поместить больше информации в память того же объема. Есть и вторая причина. Сжатие информации ускоряет ее передачу. Но об этом -- в следующей главе.

Существует несколько методов сжатия (компрессии10) данных. Все их можно разделить на две группы -- сжатие без потерь и с потерями. В первом случае распакованное сообщение точно повторяет исходное. Естественно, так можно обрабатывать любую информацию. Сжатие же с потерями возможно только в тех случаях, когда допустимы некоторые искажения -- какие именно, зависит от конкретного типа данных.

Практически все методы сжатия без потерь основаны на одной из двух довольно простых идей.

Одна из них впервые появилась в методе сжатия текстовой информации, предложенном в 1952 году Хафманом. Вы знаете, что стандартно каждый символ текста кодируется одним байтом. Но дело в том, что одни буквы встречаются чаще, а другие реже. Например, в тексте, написанном на русском языке, в каждой тысяче символов в среднем будет 90 букв "о", 72 -- "е" и только 2 -- "ф". Больше же всего окажется пробелов: сто семьдесят четыре. Если для наиболее распространенных символов использовать более короткие коды (меньше 8 бит), а для менее распространенных -- длинные (больше 8 бит), текст в целом займет меньше памяти, чем при стандартной кодировке.

Несколько методов сжатия основаны на учете повторяющихся байтов или последовательностей байт. Простейший из них -- RLE11 -- широко используется при сжатии изображений. В файле, сжатом таким методом, записывается, сколько раз повторяются одинаковые байты. Например, вместо "RRRRRGGGBBBBBBRRRBBRRRRRRR" будет храниться "5R3G6B3R2B7R"12. Очевидно, что такой метод лучше всего работает, когда изображение содержит большие участки с однотонной закраской.

Другие методы основаны на том, что если некоторая последовательность байт встречается в файле многократно, ее можно записать один раз в особую таблицу, а потом просто указывать: "взять столько-то байт из такого-то места таблицы"13.

Методы сжатия без потерь уменьшают размер файлов не очень сильно. Обычно коэффициент сжатия не превосходит 1/3—1/4. Гораздо лучших результатов можно добиться, используя сжатие с потерями. В этом случае на основе специальных исследований определяется, какой информацией можно пожертвовать.

Например, установлено, что человеческое зрение очень чувствительно к изменению яркости и гораздо меньше, к цветовому тону. Поэтому при сжатии фотографических изображений (и вообще, изображений, в которых нет резких границ между цветами) можно исключить информацию о цвете части пикселов. При распаковке же определять его по соседним. На практике чаще всего применяется метод, использующий более сложную обработку, -- JPEG14. Он позволяет сжимать изображение в десятки раз. С учетом особенностей восприятия человеком информации строятся также методы сжатия с потерями видеоизображения (наиболее распространены сейчас методы MPEG15) и звука.

Естественно, сжатие с потерями может использоваться только программами, предназначеными для обработки конкретных видов данных (например, графическими редакторами). А вот методы сжатия без потерь применяются и для любых произвольных файлов (широко известны программы-компрессоры ARJ, ZIP, RAR, StuffIt и др).

Заметим, что не стоит пытаться сжать файлы, которые уже были сжаты: размер их либо уменьшится совсем незначительно, либо даже увеличится.

**Примечания**

На самом деле, в телевизионном изображении 625 строк.

Compressus (лат.) -- сжимание.

Run-Length Encoding (англ.) -- кодирование длины последовательности.

На самом деле, конечно, используются коды цветов и коды, указывающие либо сколько раз повторяется следующий байт, либо сколько следующих байтов -- неповторяющиеся.

На этой идее основан широко использующийся для сжатия различных данных метод LZW, названный так по первым буквам фамилий его разработчиков: Lempel, Ziv и Welch.

Joint Photographic Experts Group (англ.) -- Объединенная группа экспертов по фотографии, разработавшая одноименный метод сжатия изображений.

Moving Picture Experts Group (англ.) -- Группа экспертов по движущимся изображениям