

# Лабораторная работа № 1

## Сжатие информации методом RLE

### 1. Цель работы

Изучить метод сжатия информации RLE, получить практические навыки сжатия файлов стандартными архиваторами.

### 2. Общие сведения

Несмотря на то, что объёмы внешней памяти ЭВМ постоянно растут, потребность в сжатии информации не уменьшается. Это объясняется тем, что сжатие необходимо не только для экономии места в памяти ЭВМ, но и для быстрой передачи информации по Сети (например, для трансляции спортивного соревнования в реальном масштабе времени).

Кроме того, возможность отказа магнитных и полупроводниковых носителей информации, разрушающее действие вирусов заставляют пользователей делать резервное копирование ценной информации на другие (запасные) носители информации. Очевидно, что разумнее информацию хранить сжатой.

**Сжатие информации (архивация)** — это такое преобразование информации, при котором объем файла уменьшается, а количество информации, содержащейся в архиве, остаётся прежним.

Степень сжатия информации зависит от содержимого файла и формата файла, а также от выбранного метода сжатия информации. Степень (качество) сжатия файлов характеризуется **коэффициентом сжатия**  $K_c$ , определяемым как отношение объёма исходного файла  $V_o$  к объёму сжатого файла  $V_c$ :

$$K_c = \frac{V_o}{V_c}.$$

Чем больше величина  $K_c$ , тем выше степень сжатия информации.

Все существующие методы сжатия информации можно разделить на два класса: сжатие без потерь информации (обратимый алгоритм) и сжатие с потерей информации (необратимый алгоритм). В первом случае исходную информацию можно точно восстановить по имеющейся упакованной информации. Во втором случае распакованное сообщение (например, фотография) будет незначительно отличаться от исходного сообщения.

Одна из идей сжатия состоит в учёте того факта, что в файлах часто встречаются несколько подряд идущих одинаковых байтов, а некоторые последовательности байтов повторяются многократно. При архивации такие места файла можно заменить командами вида «повторить данный байт  $n$  раз» или «взять часть данных длиной  $k$  байтов, которая встречалась  $m$  байтов назад». Такой алгоритм архивации называется методом сжатия информации RLE (**Run Length Encoding** — кодирование путём учёта повторений или кодирование длин серий).

Рассмотрим основную идею сжатия информации методом RLE.

Упакованная методом RLE последовательность состоит из **управляющих байтов**, за которыми следуют один или несколько байтов данных. При этом, если старший бит управляющего байта равен 1, то следующий за ним байт данных нужно повторить при декодировании столько раз, сколько указано в оставшихся 7 битах управляющего байта.

Например, управляющий байт 10001001 говорит, что следующий за ним байт нужно повторить 9 раз, так как  $1001_2 = 9_{10}$ .

Если старший бит управляющего байта равен 0, то при декодировании нужно взять несколько следующих байтов без изменений. Число байтов, которые берутся без изменений, указывается в оставшихся 7 битах. Например, управляющий байт 00000011 говорит, что следующие за ним 3 байта нужно взять без изменений.

Рассмотрим пример сжатия методом RLE.

Пусть дана некоторая последовательность из 12 байтов:

```
11111111 11111111 11111111 11111111 11111111 11110000
00001111 11000011 10101010 10101010 10101010 10101010.
```

В начале исходной двоичной последовательности 5 раз повторяется байт 11111111. Чтобы упаковать эти 5 байтов, нужно записать сначала управляющий байт 10000101, а затем повторяемый байт 11111111. В результате сжатия этого фрагмента данных выигрыш составит 3 байта. Далее идут 3 разных (неповторяющихся) байта: 11110000 00001111 и 11000011. Чтобы их «упаковать», нужно записать управляющий байт 00000011, а затем указать эти 3 неповторяющихся байта. В результате архивации этого фрагмента двоичной последовательности получается увеличение объема архива по сравнению с исходными данными на 1 байт. Далее в последовательности 4 раза повторяется байт 10101010. Для архивации этого фрагмента двоичных данных нужно сформировать управляющий байт 10000100 и записать один раз повторяемый байт 10101010. Сжатие последнего фрагмента даст выигрыш 2 байта.

В результате такой архивации получена новая последовательность данных (архив), состоящая из 8 байтов:

```
10000101 11111111 00000011 11110000
00001111 11000011 10000100 10101010.
```

Таким образом, 12 байт исходной двоичной последовательности удалось сжать до 8 байт.

$$K_c = \frac{12}{8} = 1,5.$$

### 3. Задания на выполнение лабораторной работы

#### 3.1. Задание 1. Исследование эффективности сжатия файлов различных форматов

С помощью архиватора (WinZip, WinRar, 7-Zip и т.п.) выполнить сжатие различных документов, формат которых указан в таблице 3.1.1

Табл. 3.1.1

Документ	Расширение	Объем файла до архивации, Кбайт	Объем файла после архивации, Кбайт	Коэффициент сжатия
Текст	.doc			
Рисунок	.jpg			
Рисунок	.bmp			
Видео	.avi			
Звук	.mp3			
Звук	.wav			
Web-стр	.html			

В качестве текстового документа нужно взять файл, который не содержит рисунков. Число символов должно быть более 3000 знаков.

Фотографии формата JPG можно взять, например, на сайте samara.psati.ru в соответствии с вариантом (таблица 3.1.2).

Табл. 3.1.2

Вар	Раздел	Вар	Раздел
1	Администрация города	9	Животный мир
2	Учебные заведения	10	Набережные, пляжи
3	Парки, сады, скверы	11	Ночной город
4	Водоёмы	12	Спортивные сооружения
5	Вокзалы	13	Культовые сооружения
6	Растительный мир	14	Памятники и скульптуры
7	Дворцы, дома	15	Театры, концертные залы
8	Музеи, выставки	16	Улицы, проспекты

Рисунок формата BMP следует нарисовать в растровом графическом редакторе. Рисунок должен содержать изображение флага государства в соответствии с таблицей 3.1.3.

Табл.3.1.3

Вариант	Страна	Вариант	Страна
1.	Россия	9.	Германия
2.	Украина	10.	Польша
3.	Франция	11.	Голландия
4.	Япония	12.	Болгария
5.	Венгрия	13.	Литва
6.	Италия	14.	Эстония
7.	Армения	15.	Румыния
8.	Латвии	16.	Австрия

Видеокалип формата AVI желательнo снять самостоятелно (с помощью видеокамеры, цифрового фотоаппарата, мобильного телефона, планшета).

В качестве звукового файла нужно использовать своё любимое музыкальное произведение или сделать самостоятельную запись речи.

### 3.2. Задание 2. Сжатие информации методом RLE

Выполнить ручную кодирование сообщения методом RLE. В качестве исходной фразы взять текст из табл. 3.2.1. С помощью таблицы CP-1251 (см. Приложение 1) перевести символы заданной фразы в двоичные числа. Выполнить сжатие информации, вычислить контрольную сумму и коэффициент сжатия.

Табл. 3.2.1.

Вар	Текст	Вар	Текст
1	Кредитка 2235555666122	9	Счет 95122244445333333
2	Паспорт 25700000333215	10	Касса 1478885555233333
3	ИНН 78888255555488856	11	Прошло 11100002 секунд
4	Пароль 17775556666612	12	Пролетели 82223333352 м
5	Пароль abcWWWZZZq	13	Вес 159755553333331 кг
6	Автомобиль 78999994441	14	Цена 2598888666611 коп
7	Алло это 456555544488	15	Мощность 3574444555 Вт
8	Удостоверение 265444111	16	Выиграл 10000555 рублей

## 4. Порядок выполнения лабораторной работы

### 4.1. Методические указания к заданию 3.1

Работа со стандартными архиваторами, как правило, осуществляется с помощью контекстного меню. После щелчка по архивируемому файлу правой кнопкой мыши нужно следовать инструкциям (Add to “Имя файла”.rar). Для сжатия допустимо использовать любой архиватор.

### 4.2. Методические указания к заданию 3.2

Выполним сжатие сообщения методом RLE. Пусть текст сообщения будет таким: ИНН 22223133333

Текст	Десятичный код (таблица CP-1251)	Двоичный код	Архив
И	200	11001000	00000001
Н	205	11001101	11001000
Н	205	11001101	10000010
Пробел	32	00100000	11001101
2	50	00110010	00000001
2	50	00110010	00100000
2	50	00110010	10000100
2	50	00110010	00110010
3	51	00110011	00000010
1	49	00110001	00110011
3	51	00110011	00110001
3	51	00110011	10000101
3	51	00110011	00110011
3	51	00110011	
3	51	00110011	
КС двоичная			1010111В
КС шестнадц.			A7H

Таким образом, пятнадцать байт исходного текста сжаты до тринадцати байт. Коэффициент сжатия в данном случае составил:

$$K_c = \frac{15}{13} = 1,154.$$

Для упрощения проверки результата сжатия были вычислены контрольные суммы (КС) в двоичной и шестнадцатеричной системах счисления. Для этого одноименные разряды архива были просуммированы по правилу Исключающее ИЛИ.

## **5. Требования к отчёту**

Отчёт подготавливается в электронном виде. Он должен содержать рисунок флага, таблицу с результатами экспериментальных исследований эффективности сжатия файлов разного формата, а также результаты кодирования методом RLE.

## **6. Контрольные вопросы**

- 6.1. Перечислите известные Вам методы сжатия информации без потерь.
- 6.2. В чём состоит отличие методов сжатия с потерями и без потерь?
- 6.3. Сколько бит в управляющем байте отводят для указания числа повторяющихся байтов при сжатии методом кодирования длин серий?
- 6.4. О чём говорит равенство единице старшего бита в управляющем байте при сжатии методом кодирования длин серий?
- 6.5. Перечислите известные Вам архиваторы.
- 6.6. Целесообразно ли выполнять сжатие файлов формата JPEG, MP3, MPEG?
- 6.7. Рисунок какого формата будет сжат сильнее BMP или JPEG?
- 6.8. Какой код является неравномерным: RLE или Хаффмана?
- 6.9. Что называется кодом?

## **7. Список литературы**

1. Алексеев А.П. Информатика 2015: учебное пособие/ Алексеев А.П. – М: СОЛОН-Пресс, 2015. – 400 с. ISBN 978-5-91359-158-6.