

# Лабораторная работа

## Тема: Сжатие информации.

### 1. Цель.

Целью лабораторной работы является получение навыков работы с архиваторами RAR, ARJ и ZIP, и ознакомление с основными алгоритмами сжатия информации.

### 2. Лабораторное задание.

1. Найдите на компьютере не менее 5-ти текстовых файлов (расширение .txt)
2. Произведите их сжатие архиватором RAR в обычный и SFX-архив.
3. Зафиксируйте размер файла до сжатия и после него.
4. Вычислите коэффициент сжатия (отношение размера исходного файла к размеру сжатого файла)
5. Повторите пункты 1-4 для графических файлов (расширение .bmp)
6. Повторите пункты 1-4 для графических файлов (расширение .jpg)
7. Повторите пункты 1-4 для звуковых файлов (расширение .wav)
8. Сведите полученные результаты в таблицу. Сделайте выводы о том, какие файлы сжимаются лучше.
9. Напишите отчет о проделанной работе.

### 3. Содержание отчета.

Отчет должен содержать следующие разделы:

- ✓ Ответы на контрольные вопросы.
- ✓ Результаты сжатия файлов в виде таблицы.
- ✓ Выводы о проделанной работе.

### 4. Методические указания по выполнению лабораторной работы.

При эксплуатации персональных компьютеров по самым различным причинам возможны порча или потеря информации на магнитных дисках. Это может произойти из-за физической порчи магнитного диска, неправильной корректировки или случайного уничтожения файлов, разрушения информации компьютерным вирусом и т.д. Для того чтобы уменьшить потери в таких ситуациях, следует иметь **архивные копии** используемых файлов и систематически обновлять копии изменяемых файлов. Для хранения архивов данных можно использовать внешние запоминающие устройства большой емкости, которые дают возможность легко скопировать жесткий диск (например, магнитооптика, стримеры, "Арвид" и др.)

Однако при этом резервные копии занимают столько же места, сколько занимают исходные файлы, и для копирования нужных файлов может потребоваться много дискет.

Более удобно для создания архивных копий использовать специально разработанные **программы архивации файлов**, которые сжимают информацию. При архивировании степень сжатия файлов сильно зависит от их формата. Некоторые форматы данных (графические, Page Maker и др.) имеют упакованные разновидности, при этом сжатие производится создающей исходный файл программой, однако лучшие архиваторы способны поджать и их. Совсем другая картина наблюдается при архивации текстовых файлов. Текстовые файлы обычно сжимаются на 50-70%, а программы на 20-30%.

**Принцип работы архиваторов** основан на поиске в файле "избыточной" информации и последующем ее кодировании с целью получения минимального объема. Самым известным методом архивации файлов является сжатие последовательностей одинаковых символов. Например, внутри вашего файла находятся последовательности байтов, которые часто повторяются. Вместо того чтобы хранить каждый байт, фиксируется количество повторяющихся символов и их позиция. Для наглядности приведем следующий пример:

Упаковываемый файл занимает 15 байт и состоит из следующей последовательности символов: **B BBBB LLLLLL AAAAA**

В шестнадцатиричной системе :

**42 42 42 42 42 4C 4C 4C 4C 4C 41 41 41 41 41**

Архиватор может представить этот файл в следующем шестнадцатиричном виде: **01 05 42 06 05 4C 0A 05 41**

Эти последовательности можно интерпретировать следующим образом: с первой позиции 5 раз повторяется знак B, с шестой позиции 5 раз повторяется знак L и с позиции 11 5 раз повторяется знак A. Согласитесь, очень простая демонстрация алгоритма архивации. Очевидно, что для хранения файла в его последней форме требуется лишь 9 байт - меньше на 6 байт.

Описанный метод является простым и очень эффективным способом сжатия файлов. Однако он не обеспечивает большой экономии объема, если обрабатываемый текст содержит небольшое количество последовательностей повторяющихся символов.

**Существуют два основных способа проведения сжатия:**

- статистический
- словарный.

Лучшие статистические методы применяют арифметическое кодирование, лучшие словарные - метод Зива-Лемпела. В **статистическом сжатии** каждому символу присваивается код, основанный на вероятности его появления в тексте. Высоко вероятные символы получают короткие коды, и наоборот. Такой способ сжатия называют оптимальным префиксным кодом. Для его построения используют алгоритмы Хаффмана или Шеннона-Фано. Например, анализируя любой английский текст, можно установить, что буква E встречается гораздо чаще, чем Z, а X и Q относятся к наименее встречающимся. Таким образом, используя специальную таблицу соответствия, можно закодировать каждую букву E меньшим числом бит, используя более длинный код для более редких букв, тогда как в обычных кодировках любому символу

соответствует битовая последовательность фиксированной длины (как правило, кратной байту).

В *словарном методе* группы последовательных символов или "фраз" заменяются кодом. Замененная фраза может быть найдена в некотором "словаре".

Популярные архиваторы **ARJ**, **RAR** работают на основе алгоритма Лемпела-Зива. Сущность алгоритмов Зива и Лемпела состоит в том, что фразы заменяются указателем на то место, где они в тексте уже ранее появлялись. Это семейство алгоритмов обозначается как LZ-сжатие. Такой метод быстро приспосабливается к структуре текста и может кодировать короткие функциональные слова, т.к. они очень часто в нем появляются. Новые слова и фразы могут также формироваться из частей ранее встреченных слов.

Декодирование сжатого текста осуществляется напрямую - происходит простая замена указателя готовой фразой из словаря, на которую тот указывает. На практике LZ-метод добивается хорошего сжатия, его важным свойством является очень быстрая работа декодировщика.

Одной из форм такого указателя является пара  $(m,l)$ , которая заменяет фразу из  $l$  символов, начинающуюся со смещения  $m$  во входном потоке. Например, указатель  $(7,2)$  адресует 7-ой и 8-ой символы исходной строки. Используя это обозначение, строка "abbaabbbabab" будет закодирована как "abba(1,3)(3,2)(8,3)". Заметим, что несмотря на рекурсию в последнем указателе, производимое кодирование не будет двусмысленным.

Распространено не верное представление, что за понятием LZ-метода стоит единственный алгоритм. Из-за большого числа вариантов этого метода лучшее описание можно осуществить только через его растущую семью, где каждый член отражает свое решение разработчика. Эти версии отличаются друг от друга в двух главных факторах:

- есть ли предел обратного хода указателя,
- и на какие подстроки из этого множества он может ссылаться.

Продвижение указателя в ранее просмотренную часть текста может быть неограниченным (расширяющееся окно) или ограничено окном постоянного размера из  $N$  предшествующих символов, где  $N$  обычно составляет несколько тысяч. Выбранные подстроки также могут быть неограниченным или ограниченным множеством фраз, выбранных согласно некоторому замыслу. Каждая комбинация этих условий является компромиссом между скоростью выполнения, объемом требуемой ОП и качеством сжатия.

*Расширяющееся окно* предлагает лучшее сжатие за счет организации доступа к большему количеству подстрок. Но по мере роста окна, кодировщик может замедлить свою работу из-за возрастания времени поиска соответствующих подстрок, а сжатие может ухудшиться из-за увеличения размеров указателей. Если памяти для окна будет не хватать, произойдет сброс процесса, что также ухудшит сжатие до поры нового увеличения окна. *Окно постоянного размера* лишено этих проблем, но содержит меньше подстрок, доступных указателю. Ограничение множества доступных подстрок

размерами фиксированного окна уменьшает размер указателей и убыстряет кодирование.

### **К основным функциям архиваторов относятся:**

- архивация указанных файлов или всего текущего каталога;
- извлечение отдельных или всех файлов из архива;
- просмотр содержимого архивного файла;
- проверка целостности архивов;
- восстановление поврежденных архивов;
- ведение многотомных архивов;
- вывод файлов из архива на экран или на печать;
- парольная защита архива.

Архиватор **ARJ** не имеет графического интерфейса, и вся работа с ним осуществляется с командной строки. Формат команд имеет следующий вид:

**arj <команда> [- <спецификация1> [- <спецификация2>]...] <имя архива> [<имя файла>...]**

Подробную информацию о списке команд архиватора можно получить, набрав в командной строке: **arj /?**

Рассмотрим наиболее популярные команды архиватора:

Для архивации файлов: **arj a <имя архива> <имя файла1> <имя файла2>**

Для извлечения файлов из архива: **arj e <имя архива> <имя файла1> <имя файла2>**

Для просмотра содержимого архивного файла: **arj l <имя архива>**

Для проверки целостности архива: **arj t <имя архива>**

Для восстановления испорченного архива: **arj -jr <имя архива>**

Для создания многотомного архива: **arj a -v<размер тома> <имя архива> <имя файла1> <имя файла2>**

Для вывода файла из архива на экран: **arj p <имя архива> <имя файла>**

Для создания архива с паролем: **arj a -g<пароль> <имя архива> <имя файла>** или **arj a -g? <имя архива> <имя файла>** в последнем случае пароль будет запрошен отдельной строкой.

Для создания самораспаковывающихся архивов: **arj a -je <имя архива> <имя файла1> <имя файла2>**

Архиватор RAR имеет версии, как для **Dos, Win 3.XX** так и для **Windows 95/98**. Последние версии **WinRar** имеют графический интерфейс и работа с ними очень проста и понятна. Данный архиватор позволяет создавать как архивы \*.rar так и архивы \*.zip

К достоинствам данного архиватора можно отнести:

- графический интерфейс;
- высокую степень сжатия, даже мультимедийных файлов;
- возможность оценить размер архива, не производя архивирование.
- большую вероятность восстановления поврежденных архивов.

## **5. Контрольные вопросы.**

1. Зачем нужно архивировать информацию?
2. На чем основана работа архиваторов. По какому принципу они сжимают информацию.
3. Каковы функции архиваторов.
4. Чем отличаются SFX – архивы.