Сжатие данных

1.Теоретические основы сжатия данных
2.Обратимость сжатия
3.Алгоритмы сжатия данных.
4.Программные средства сжатия данных
5.Базовые требования к диспетчерам архивов

1.Теоретические основы сжатия данных

Характерной особенностью большинства типов данных, является избыточность.

Степень избыточности зависит от типа данных. У видеоданных степень избыточности в несколько раз больше, чем у графических, а степень избыточности графических данных в несколько раз больше, чем текстовых. Степень избыточности данных зависит от принятой системы кодирования.

При обработке информации избыточность также играет важную роль. Так, например, при преобразовании информации избыточность используют для повышения ее качества (актуальности, адекватности и т. п.).

Объекты сжатия

В зависимости от того, в каком объекте размещены данные, подвергаемые сжатию, различают:

- уплотнение(архивацию) файлов;
- уплотнение (архивацию) папок;
- уплотнение дисков.
- *Уплотнение файлов* применяют для уменьшения их размеров при подготовке к передаче по каналам электронных сетей или к транспортировке на внешнем носителе малой емкости
- *Уплотнение папок* используют как средство архивации данных перед длительным хранением, в частности при резервном копировании.
- *Уплотнение дисков* служит для повышения эффективности использования их рабочего пространства.

2. Обратимость сжатия

Существует три способа сжатия данных.

Это изменение содержания данных, изменение их структуры, либо и то и другое вместе.

Если при сжатии данных происходит изменение их содержания, метод сжатия необратим и при восстановлении данных из сжатого файла не происходит полного восстановления исходной последовательности. Такие методы называют ***методами сжатия с регулируемой потерей информации***. Они применимы только для типов данных, для которых формальная утрата части содержания не приводит к значительному снижению потребительских свойств. Это относится к мультимедийным данным: видеорядам, музыкальным записям, звукозаписям и рисункам.

Методы сжатия с потерей информации обеспечивают более высокую степень сжатия, чем обратимые методы, но их нельзя применять к текстовым документам, базам данных и, к программному коду.

Характерными форматами сжатия с потерей информации являются:

.JPG для графических данных;

.MPG для видеоданных;

.MP3 для звуковых данных.

Если при сжатии данных происходит только изменение их структуры, то метод сжатия обратим.

Из результирующего кода можно восстановить исходный массив путем применения обратного метода. Обратимые методы применяют для сжатия любых типов данных.

Характерными форматами сжатия без потери информации являются:

- .GIF, .TIP, .PCX и многие другие для графических данных;
- .AVI для видеоданных;
- .ZIP, .ARJ, .RAR, .LZH, .LH, .CAB и многие другие для любых типов данных.

Алгоритмы обратимых методов

При исследовании методов сжатия данных следует иметь в виду существование следующих доказанных теорем.

- **Для любой последовательности данных существует теоретический предел сжатия, который не может быть превышен без потери части информации.**
- **Для любого алгоритма сжатия можно указать такую последовательность данных, для которой он обеспечит лучшую степень сжатия, чем другие методы.**
- **Для любого алгоритма сжатия можно указать такую последовательность данных, для которой данный алгоритм вообще не позволит получить сжатия.**

Существует достаточно много обратимых методов сжатия данных, однако в их основе лежит сравнительно небольшое количество теоретических алгоритмов, представленных в таблице 1.

Свойства алгоритмов сжатия

Свойства алгоритмов сжатия

Алгоритм	Выходная структура	Сфера применения	Примечание
RLE (Run-Length Encoding	Список (вектор данных)	Графические данные	Эффективность алгоритма не зависит от объема данных
KWE (Keyword Encoding)	Таблица данных (словарь)	Текстовые данные	Эффективен для массивов большого объема
Алгоритм Хафмана	Иерархическая структура (дерево кодировки)	Любые данные	Эффективен для массивов большого объема

3. Алгоритмы сжатия данных.

Алгоритм RLE.

В основу алгоритмов *RLE* положен принцип выявления повторяющихся последовательностей данных и замены их простой структурой, в которой указывается код данных и коэффициент повтора.

Например, для последовательности: 0; 0; 0; 127; 127; 0; 255; 255; 255; 255 (всего 10 байтов) образуется следующий вектор:

Значение	Коэффициент повтора
0	3
127	2
0	1
255	4

Программные реализации алгоритмов *RLE* отличаются простотой, высокой скоростью работы, но обеспечивают недостаточное сжатие. Наилучшими объектами для данного алгоритма являются графические файлы, в которых большие одноцветные участки изображения кодируются длинными последовательностями одинаковых байтов. Этот метод также может давать заметный выигрыш на некоторых типах файлов баз данных, имеющих таблицы с фиксированной длиной полей. Для текстовых данных методы *RLE* не эффективны.

При записи в поток имеем: 0; 5; 127; 2; 0; 1; 255; 4 (всего 8 байтов).

В данном примере коэффициент сжатия равен 8/10 (экономия объема составляет 20%).

Программные реализации алгоритмов *RLE* отличаются простотой, высокой скоростью работы, но обеспечивают недостаточное сжатие. Наилучшими объектами для данного алгоритма являются графические файлы, в которых большие одноцветные участки изображения кодируются длинными последовательностями одинаковых байтов. Этот метод также может давать заметный выигрыш на некоторых типах файлов баз данных, имеющих таблицы с фиксированной длиной полей. Для текстовых данных методы *RLE* не эффективны.

Алгоритм KWE

В основу алгоритмов кодирования по ключевым словам *(Keyword Encoding)* положено кодирование лексических единиц исходного документа группами байтов фиксированной длины. Лексическая единица - слово (последовательность символов, справа и слева ограниченная пробелами или символами конца абзаца). Результат кодирования сводится в таблицу, которая прикладывается к результирующему коду и представляет собой словарь. Для англоязычных текстов используется двухбайтная кодировка слов. Образующиеся при этом пары байтов называют **токенами.**

Алгоритм эффективен для англоязычных текстовых документов и файлов баз данных. Для русскоязычных документов, отличающихся увеличенной длиной слов и большим количеством приставок, суффиксов и окончаний, не удается ограничиться двухбайтными **токенами**, и эффективность метода снижается.

Алгоритм Хаффмана

В основе этого алгоритма лежит кодирование не байтами, а битовыми группами.

- Перед началом кодирования производится частотный анализ кода документа и выявляется частота повтора каждого из встречающихся символов.
- Чем чаще встречается тот или иной символ, тем меньшим количеством битов он кодируется (соответственно, чем реже встречается символ, тем длиннее его кодовая битовая последовательность).
- Образующаяся в результате кодирования иерархическая структура прикладывается к сжатому документу в качестве таблицы соответствия.

Пример кодирования символов русского алфавита представлен на рис. 1. Как видно из схемы, представленной на рисунке, используя 16 бит, можно закодировать до 256 различных символов. Однако ничто не мешает использовать и последовательности длиной до 20 бит — тогда можно закодировать до 1024 лексических единиц (это могут быть не символы, а группы символов, слоги и даже слова).

Рис. 1. Пример буквенного кодирования информации по алгоритму Хафмана

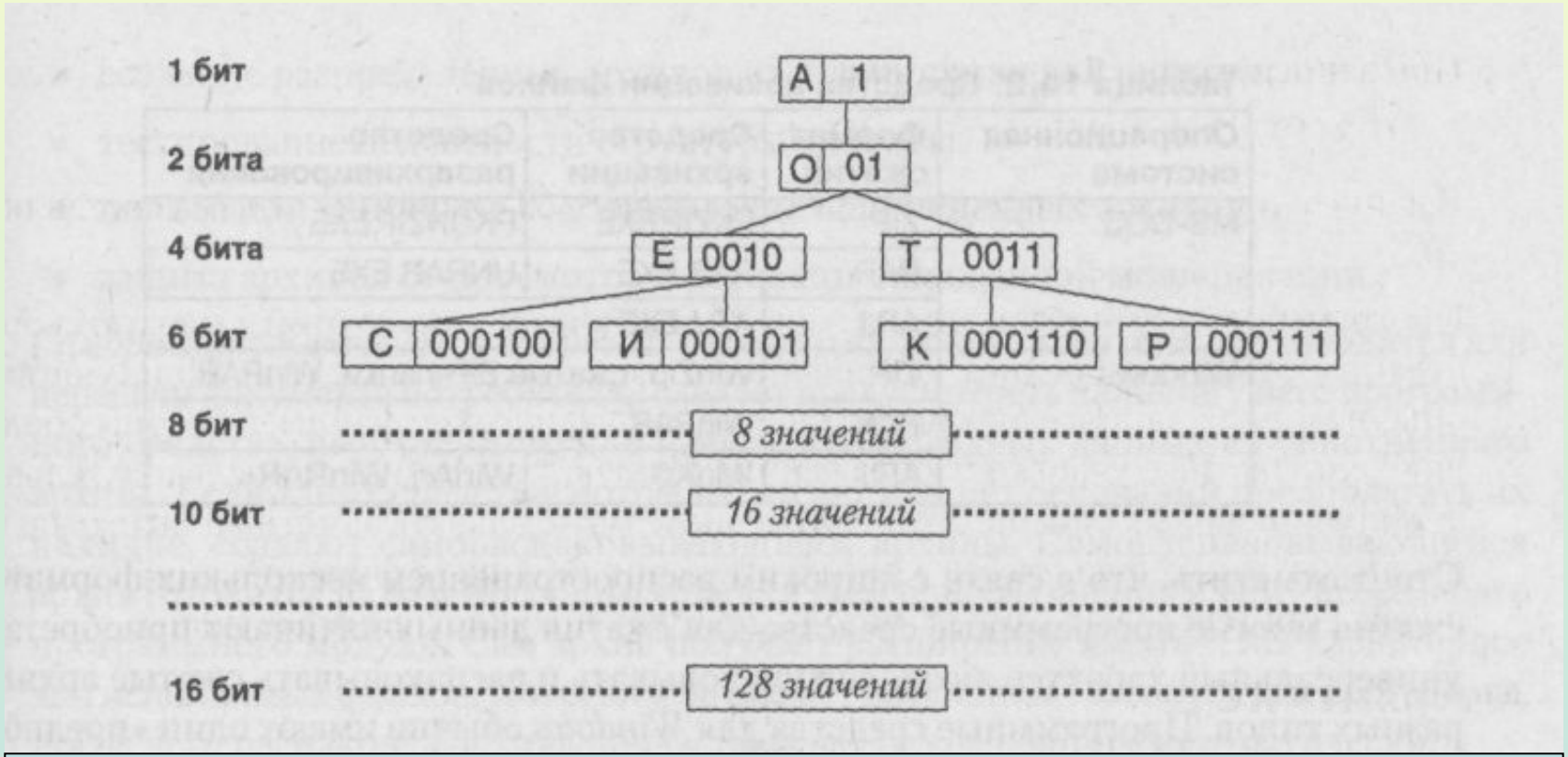

Рис. 1. Пример буквенного кодирования информации по алгоритму Хафмана

В связи с тем, что к сжатому архиву необходимо прикладывать таблицу соответствия, на файлах малых размеров алгоритм Хаффмана малоэффективен. Эффективность алгоритма зависит и от заданной предельной длины кода (размера словаря). Наиболее эффективными оказываются архивы с размером словаря от 512 до 1024 единиц (длина кода до 18-20 бит).

4. Программные средства сжатия данных

«Классическими» форматами сжатия данных являются форматы .ZIP, RAR и .ARJ. Программные средства, предназначенные для создания и обслуживания архивов, выполненных в данных форматах, приведены в табл.

Средства архивации файлов

Средства архивации файлов

Операционная система	Формат сжатия	Средство архивации	Средство разархивирования
MS-DOS	.ZIP	PKZIP.EXE	PKUNZIP.EXE
	.RAR	RAR.EXE	UNRAR.EXE
	.ARJ	ARJ.EXE	
Windows	.ZIP	WinZip, сжатые ZIP-папки, WinRAR	
	.RAR	WinRAR	
	.ARJ	WinArj	WinArj, WinRAR

- Для *Windows* Наиболее распространен формат .ZIP, который является стандартом де-факто для архивов, распространяемых через Интернет. Важную роль в этом играет открытость этого формата. Его использование не требует лицензионных отчислений.
- Операционная система *Windows XP* позволяет рассматривать *ZIP*-архивы как сжатые папки. Все файловые операции можно выполнять в сжатой папке так же, как в обычной. Однако специализированные средства работы с архивами обеспечивают более широкий набор функций.

5. Базовые требования к диспетчерам архивов

Программные средства для создания и обслуживания архивов отличаются большим объемом функциональных возможностей, многие из которых выходят далеко за рамки простого сжатия данных и эффективно дополняют стандартные средства операционной системы. Современные средства архивации данных называют *диспетчерами архивов*.

Базовые функции диспетчеров архивов:

- извлечение файлов из архивов;
- создание новых архивов;
- добавление файлов в имеющийся архив;
- создание самораспаковывающихся архивов;
- создание распределенных архивов на носителях малой емкости;
- тестирование целостности структуры архивов;
- полное или частичное восстановление поврежденных архивов;
- защита архивов от просмотра и несанкционированной модификации.

- *Самораспаковывающиеся архивы.* Самораспаковывающийся архив готовится на базе обычного архива путем присоединения к нему небольшого программного модуля. Сам архив получает расширение имени .EXE, характерное для исполнимых файлов. Потребитель сможет выполнить его запуск как программы, после чего распаковка архива произойдет на его компьютере автоматически.

Распределенные архивы. В тех случаях, когда предполагается передача большого архива на носителях малой емкости, например на гибких дисках, возможно распределение одного архива в виде малых фрагментов на нескольких носителях.

Современные диспетчеры архивов способны выполнить предварительное разбиение архива на фрагменты заданного размера на жестком диске. Впоследствии их можно перенести на внешние носители путем копирования. Все файлы распределенного архива получают разные имена, их последующее упорядочение не вызывает проблем.

Оптимальный режим работы с распределенными архивами следующий:

- создание набора файлов распределенного архива в папке на жестком диске;
- копирование файлов распределенного архива на отдельные сменные носители;
- перенос (перевозка) сменных носителей в место назначения;
- копирование файлов распределенного архива со сменных носителей в одну папку на конечном жестком диске;

Дополнительные требования к диспетчерам архивов

К дополнительным функциям диспетчеров архивов относятся сервисные функции, делающие работу более удобной они обеспечивают:

- просмотр файлов различных форматов без извлечения их из архива;
- поиск файлов и данных внутри архивов;
- установку программ из архивов без предварительной распаковки;
- проверку отсутствия компьютерных вирусов в архиве до его распаковки;
- криптографическую защиту архивной информации;
- декодирование сообщений электронной почты;
- «прозрачное» уплотнение исполнимых файлов .EXE и .DLL;
- создание самораспаковывающихся многотомных архивов;
- выбор или настройку коэффициента сжатия информации.

- *Защита архивов.* В большинстве случаев защиту архивов выполняют с помощью пароля, который запрашивается при попытке просмотреть, распаковать или изменить архив. Теоретически, защита с помощью пароля считается неудовлетворительной и не рекомендуется для особо важной информации. Основные программные средства, используемые для восстановления утраченного пароля (или взлома закрытой информации), используют методы прямого перебора. Работу этих средств можно существенно затруднить и замедлить, если расширить область перебора. Криптостойкость защиты можно повысить за счет знаков препинания, использования символов русского алфавита.

Дополнительные требования к диспетчерам архивов

К дополнительным функциям диспетчеров архивов относятся сервисные функции, делающие работу более удобной. Они часто реализуются внешним подключением дополнительных служебных программ и обеспечивают:

- просмотр файлов различных форматов без извлечения их из архива;
- поиск файлов и данных внутри архивов;
- установку программ из архивов без предварительной распаковки;
- проверку отсутствия компьютерных вирусов в архиве до его распаковки;
- криптографическую защиту архивной информации;
- декодирование сообщений электронной почты;
- создание самораспаковывающихся многотомных архивов;
- выбор или настройку коэффициента сжатия информации.

Программные средства уплотнения носителей

Теоретические основы

В основе уплотнения носителей (например, дисков) также лежит принцип сжатия данных за счет уменьшения избыточности путем изменения структуры, но при этом надо иметь в виду ряд особенностей:

- процесс уплотнения носителей является относительным, происходит сжатие записываемых данных, что вызывает эффект *кажущегося* увеличения емкости носителя;
- процесс сжатия данных происходит под управлением программ, работающих автоматически в фоновом режиме, и, тем самым, он «прозрачен» для пользователя, который никак не ощущает разницы в работе с обычным и уплотненным носителем, но может констатировать факт размещения на диске большего объема данных, чем физическая емкость диска;

- степень сжатия данных зависит, как мы знаем, от типа данных, поэтому наблюдаемое приращение емкости носителя не является величиной постоянной и непрерывно меняется в зависимости от того, какой тип данных добавляется на носитель;
- размер свободного пространства на сжатом томе определяется как произведение реального свободного пространства и предполагаемого (или среднего) коэффициента сжатия и поэтому является приближенной величиной, причем часто такое приближение оказывается очень грубым.

В основе алгоритмов сжатия данных, используемых для уплотнения носителей, не могут лежать необратимые методы. Некоторые типы данных (например, программный код) не допускают потери данных ни в малейшей степени.

Практическая реализация концепции уплотнения дисков

1. На физическом диске создается скрытый файл, предназначенный для записи сжатых данных. Данный файл называют *файлом сжатого тома,* а физический диск, на котором он размещен, называют *несущим диском.*
2. На уровне операционной системы происходит объявление файла сжатого тома в качестве нового *уплотненного диска.* Данные, которые записываются на уплотненный диск, на самом деле заносятся в файл сжатого тома, расположенный на несущем диске.

3. Если файл сжатого тома занимает весь несущий диск, то несущий диск делается скрытым и его место в операционной системе занимает уплотненный диск.
4. Весь обмен информацией с уплотненным диском происходит не под управлением стандартных средств операционной системы, а под управлением специальной программы — *драйвера сжатого тома,* которая интегрируется в операционную систему и организует ее взаимодействие с нестандартной файловой системой, созданной внутри файла сжатого тома.

Оценивая возможность уплотнения носителей, следует иметь в виду, что наличие такого носителя в компьютерной системе затрудняет ее обслуживание и заметно снижает надежность, в первую очередь в связи с особой сложностью восстановления информации в случае неожиданных повреждений аппаратного или программного обеспечения.