МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра информационных систем и технологий

УТВЕРЖДАЮ
Проректор по учебной работе
_____ О.Г.Локтионова
«_4_» _04_ 2019г.

Теория информационных процессов и систем

Методические указания
по выполнению практических работ
для студентов направления подготовки 02.03.03.

Курск 2019

2

УДК 681.3(075)
Составитель: Л.А. Лисицин

Рецензент
Кандидат технических наук, доцент *Халин Ю.А.*.

Теория информационных процессов и систем [Текст]: методические указания по выполнению практических работ / Юго-Зап. гос. ун-т; сост.: Л.А. Лисицин. Курск, 2019. 78 с.: ил. 16. табл. 6. Библиогр. с. 78.

Содержат сведения по технологиям сбора, хранения, обработки и передачи информации. Материал ориентирован на практическую работу студентов в компьютерной среде.

Отражен порядок выполнения практических работ и правила оформления отчетов.

Методические указания предназначены для студентов, обучающихся по направлению 02.03.03 «Математическое обеспечение и администрирование информационных систем».

Методические указания соответствуют требованиям программы, утвержденной учебно-методическим объединением по специальности «Математическое обеспечение и администрирование информационных систем».

Текст печатается в авторской редакции

Подписано в печать 04.04.19 Формат 60x84 1/16.
Усл.печ. л. 4,1. Уч.-изд. л. 3,8. Тираж 100 экз. Заказ 198. Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Оглавление

Практическое занятие №1

Формальное представление знаний. Виды информации. Способы хранения, обработки и передачи информации.

Цель: научиться сохранять, обрабатывать и передавать данные при помощи технических средств информации**.**

Оборудование: ПК, сканер, фотокамера, USB-накопитель.

Программное обеспечение: операционная система, программа для работы с видеоинформацией.

Теоретические основы

1. Технологии сбора и хранения информации

Сбор предполагает получение максимально выверенной исходной информации и является одним из самых ответственных этапов в работе с информацией, поскольку от цели сбора и методов последующей обработки полностью зависит конечный результат работы всей информационной системы.

Технология сбора подразумевает использование определенных методов сбора информации и технических средств, выбираемых в зависимости от вида информации и применяемых методов ее сбора. На заключительном этапе сбора, когда информация преобразуется в данные, т. е. в информацию, представленную в формализованном виде, пригодном для компьютерной обработки, осуществляется ее ввод в систему Для сбора данных необходимо сначала определить технические средства, позволяющие осуществлять сбор быстро и высококачественно и поддерживающие операции ввода информации и представления данных в электронной форме.

2. Технологический процесс обработки информации

Технологический процесс обработки информации — есть строго определенная последовательность взаимосвязанных процедур, выполняемых для преобразования первичной информации с момента ее возникновения до получения требуемого результата.

Технологический процесс призван автоматизировать обработку исходной информации за счет привлечения технических средств базовой информационной технологии, сократить финансовые и трудовые затраты, обеспечить высокую степень достоверности результатной информации. Для конкретной задачи той или

иной предметной области технологический процесс обработки информации разрабатывается индивидуально. Совокупность процедур зависит от следующих факторов:

- характер и сложность решаемой задачи;
- алгоритм преобразования информации;
- используемые технические средства;
- сроки обработки данных;
- используемые системы контроля;
- число пользователей и т. д.

В общем случае технологический процесс обработки информации включает процедуры.

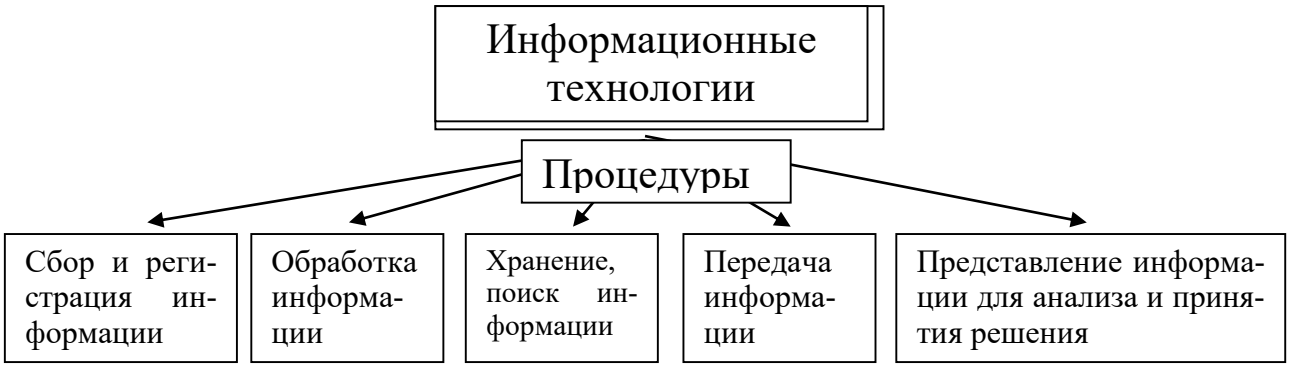


Рисунок1.1 Технологический процесс обработки информации.

3. Способы обработки информации

Современные информационные технологии позволяют обрабатывать информацию централизованным и децентрализованным (т. е. распределенным) способами.

Централизованный способ предполагает сосредоточение данных в информационно-вычислительном центре, выполняющем все основные действия технологического процесса обработки информации. Основное достоинство централизованного способа — сравнительная дешевизна обработки больших объемов информации за счет повышения загрузки вычислительных средств.

Децентрализованный способ характеризуется рассредоточением информационно-вычислительных ресурсов и распределением технологического процесса обработки информации по местам возникновения и потребления информации. Достоинством децентрализованного способа является повышение оперативности обработки информации и решения поставленных задач за счет автоматизации деятельности на конкретных рабочих местах, применения надежных

средств передачи информации, организации сбора первичных документов и ввода исходных данных в местах их возникновения [6].

Децентрализованный способ обработки информации может быть реализован автономным или сетевым методом. При автономной обработке информации передача документов и данных на электронных носителях осуществляется по почте либо курьером, а при сетевой — через современные каналы связи.

На практике применяют смешанный способ обработки информации, для которого характерны признаки двух способов одновременно (централизованный с частичной децентрализацией или децентрализованный с частичной централизацией).

4. Режимы обработки информации на компьютере

Вычислительные средства участвуют в процессе обработки информации в двух основных режимах: пакетном или диалоговом.

В случае, когда технология обработки информации на компьютере представляет собой заранее определенную последовательность операций, не требующую вмешательства человека, и диалог с пользователем отсутствует, информация обрабатывается в так называемом пакетном режиме. Суть его состоит в том, что программы обработки данных последовательно выполняются под управлением операционной системы как совокупность (пакет) заданий. Операционная система обеспечивает ввод данных, вызов требуемых программ, включение необходимых внешних устройств, координацию и управление технологическим процессом обработки информации.

Задачи, решаемые в пакетном режиме, характеризуются следующими свойствами:

• алгоритм решения задачи формализован, вмешательства пользователя не требуется;

• наличие большого объема входных и выходных данных, в основном хранящихся на устройствах хранения информации (например, жестких дисках компьютеров);

• расчет выполняется для большинства записей входных файлов;

• длительное время решения задачи — как правило, обусловлено большими объемами обрабатываемых данных;

• регламентность — задачи решаются с заданной периодичностью.

Пакетный режим возник первым и широко использовался с середины XX в., когда обработка информации на ЭВМ осуществлялась в специально создаваемых вычислительных центрах. Заказчики

подготавливали исходные данные (обычно на перфокартах или перфолентах) и отправляли их в вычислительный центр, где данные обрабатывались и результаты обработки возвращались заказчику. С развитием персональных ЭВМ (начиная с 80-х гг. прошлого века) обработка данных стала осуществляться, в основном, непосредственно потребителями, поэтому в настоящее время пакетный режим используется достаточно редко. Сегодня более распространен диалоговый режим, когда необходимо непосредственное взаимодействие пользователя с компьютером и на каждое свое действие пользователь получает немедленные ответные действия компьютера. Диалоговый режим позволяет пользователю интерактивно управлять порядком обработки информации и получать результатные данные в виде необходимых документов либо файлов.

5. Технологии передачи и представления информации

Информационные процессы невозможны без средств передачи и представления информации, поскольку зачастую информация требуется в месте, территориально удаленном от источника ее возникновения, и должна быть представлена в виде символов, образов и сигналов, пригодных для восприятия потребителем.

Современные средства связи способны передавать информацию в любой форме: телефонные, телевизионные, телеграфные сообщения, массивы данных, печатные материалы, фотографии и т. д. В соответствии со спецификой передаваемых сообщений организуется канал передачи информации — совокупность технических средств, обеспечивающих передачу сигналов от источника к потребителю.

Основная характеристика канала передачи — скорость передачи информации, а ее предельно допустимое значение называют емкостью канала, которая ограничивается шириной полосы канала и шумом.

Канал связи соединяет передатчик и приемник с помощью линии связи, которая может быть проводной, кабельной, радио, микроволновой, оптической или спутниковой. Примерами линий связи являются телефонные и вычислительные сети, сети телевизионного и радиовещания, мобильной связи, спутниковые технологии передачи данных.

В современных цифровых системах связи функции передатчика и приемника выполняет модем. Основное достоинство передачи информации в цифровой форме заключается в возможности ис-

пользования кодированных сигналов, обеспечения защиты информации и наилучшего способа приема.

Для представления переданной или хранимой информации потребителю используются процессы воспроизведения и отображения.

Воспроизведение информации — это процесс, при котором ранее записанная на носителе информация считывается устройством воспроизведения.

Отображение информации — есть представление информации, т. е. генерация сигналов на основе исходных данных, а также правил и алгоритмов их преобразования в форме, приемлемой для непосредственного восприятия человеком.

Потребителем информации наиболее часто выступает человек, и для принятия решений ему необходимы результаты обработки информации. Тем не менее человек не способен ощутить машинное представление информации, а может воспринимать ее лишь органами чувств (органами зрения, слуха, осязания, обоняния и т. д.), поэтому для организации взаимодействия человека с информационными моделями объектов информационная система должна быть наделена специальными средствами отображения данных.

Поскольку зрение используется для восприятия информации наиболее активно, то средства отображения в современных ИС должны представлять информацию в лучшей форме для визуального наблюдения. Заметим, что мультимедиа-системы позволяют также представлять информацию в форме аудио- и видеосигналов, однако для управленческих информационных систем наиболее характерно отображение информации в текстовой и графической форме, что осуществляется за счет использования мониторов и печатающих устройств (например, принтеров, плоттеров).

Прежде чем, например, на мониторе, появится информация в доступной для человека форме, компьютером будет автоматически осуществлена следующая последовательность операций:

- преобразование данных, представленных в машинной форме, в вид, приемлемый для экранного отображения;
- согласование формы представления данных с параметрами монитора;
- воспроизведение в соответствии с возможностями воспроизводящего устройства (т. е. в данном примере — монитора).

Порядок выполнения работы

1. Создание досье группы. Заранее заготовить материал: фотографии, текст.

2. Сфотографировать своих однокурсников.

3. Включить компьютер.

4. Создать общую папку на сервере.

5. Сканировать фотографии и сохранить в общую папку.

6. Включить текстовый редактор. Создать титульный лист с общей фотографией и названием группы: специальность и год.

7. Оформить каждый лист на одного человека. Записать данные: дата рождения, номер школы, хобби.

8. Сохранить данные на жесткий диск в свою папку под именем досье группы.

Отчет должен быть оформлен в текстовом редакторе и содержать:

наименование работы;

цель работы;

задание;

последовательность выполнения работы;

ответы на контрольные вопросы;

вывод о проделанной работе.

Контрольные вопросы

1. Что такое сбор информации и каково его предназначение?
2. Что понимается под технологией сбора информации?
3. Чем отличаются понятия «информация» и «данные»?
4. Назовите основные требования к сбору данных и к хранимым данным.
5. Перечислите основные средства сбора текстовой, графической, звуковой и видеоинформации. Какие еще средства сбора информации вам известны?
6. Какие еще методы сбора данных вам известны?
7. В чем заключается процедура хранения информации?
8. Перечислите основные требования к структурам хранения.
9. Что такое база данных?
10. В чем различие между базой и банком данных?
11. Что такое резервное копирование и для чего оно осуществляется?

12. Что такое архивное копирование и в чем его отличие от резервного копирования?
13. Что такое базовая информационная технология?
14. В чем заключается различие между централизованным и децентрализованным способами обработки информации?
15. Какие режимы обработки информации вам известны?

Практическое занятие № 2

Измерение количества информации. Носители информации.

Цель: научиться измерять и вычислять информацию, а также и работать с носителями информации.

Оборудование: ПК.

Программное обеспечение: операционная система, текстовый редактор.

Теоретические основы

1 байт = 8 битов.

Существуют еще более крупные единицы измерения информации.

Переход к более крупным единицам измерения информации (килобайт, мегабайт, терабайт, петабайт, эксабайт). Байт наиболее удобная единица измерения информационного объема сообщения, состоящего из последовательности символов компьютерного алфавита. Однако она мала при подсчете емкости информационных носителей. По аналогии с физическими единицами измерения (например, 1 килограмм = 1000 грамм) подбираем по таблице целых степеней двойки значение близкое к тысячи. Это значение равно 1024. Поэтому 1 килобайт = 1024 байт = 2^{10} байт, 1 мегабайт = 1024 килобайт = 2^{10} килобайт и т. д.

1 Кбайт (один килобайт) = 1024 байт;

1 Мбайт (один мегабайт) = 1024 Кбайт;

1 Гбайт (один гигабайт) = 1024 Мбайт.

Что представляют единицы измерения:

5 бит – буква в клетке кроссворда.

1 байт – символ, введенный с клавиатуры.

6 байт – средний размер слова, в тексте на русском языке.

50 байт – строка текста.

2 Кбайта – страница машинописного текста.

100 Кбайт – фотография в низком разрешении

1 Мбайт – небольшая художественная книга.

100 Мбайт – метровая книга с полками.

1 Гбайт – прочитывает человек за всю жизнь.

3 Гбайт – час качественной видеозаписи.

Для сохранения информации используют носители информации: флэш-накопители, флэш-карты, диски разных форматов, съемные жесткие диски.

Порядок выполнения работы

Изучение материала и выполнение заданий на компьютере.

Содержательный (вероятностный) подход к определению количества информации

Если заключённые в каком-то сообщении сведения являются для человека новыми, понятными, пополняют его знания, т.е. приводят к уменьшению неопределённости знаний, то сообщение содержит информацию.

1 бит – количество информации, которое содержится в сообщении, которое уменьшает неопределённость знаний в 2 раза.

Пример1. При бросании монеты возможны 2 события (случая) – монета упадёт орлом или решкой, причём оба события равновероятны (при большом количестве бросаний количество случаев падения монеты орлом и решкой одинаковы). После получения сообщения о результате падения монеты неопределённость знаний уменьшилась в 2 раза, и, поэтому, количество информации, полученное при этом равно 1 бит.

Содержательный (вероятностный) подход является субъективным, т.к. одну и ту же информацию разные люди могут оценивать по разному. Для одного человека сведения в сообщении могут быть важными и понятными, для другого бесполезными, непонятными или вредными.

Единицы измерения информации. Перевод единиц измерения.

1 бит – количество информации, которое содержится в сообщении, которое уменьшает неопределённость знаний в 2 раза.

1 бит – наименьшая единица информации. Более крупные единицы – байт, килобайт, мегабайт, гигабайт.

Система единиц измерения информации:

1 байт = 8 бит

1 Кбайт = 210 байт = 1024 байт;

1 Мбайт = 210 Кбайт = 1024 Кбайт = 220 байт;

1 Гбайт = 210 Мбайт = 1024 Мбайт = 220 кбайт = 230 байт

Информационный объём носителей информации: Дискета – 1,44 Мбайт; компакт-диск ≈ 700 Мбайт; DVD-диск – до 17 Гбайт (стандарт – 4,7 Гбайт); жёсткий диск – от 20 Гбайт до 80 Гбайт и более (стандарт 80 Гбайт); Flash-память – 256 Мбайт – 2 Гбайт.

Примеры перевода единиц:

1) 5 байт = 5 * 8 бит = 40 бит;

2) 24 бита = 24*8 байта = 3 байта;

3) 4 Кбайт = 4 * 1024 байт = 4096 байт;

4) 16384 бита = 16384 : 8 байт = 2048 байт;

2048 байт : 1024 = 2 Кбайта.

Вычисление количества информации для равновероятных событий.

Если события равновероятны, то количество информации можно рассчитать по формуле:

$$N = 2^I ,$$

где N – число возможных событий,

I – количество информации в битах.

Формула была предложена американским инженером Р. Хартли в 1928 г.

Задача 1. В коробке 32 карандаша, все карандаши разного цвета. Наугад вытащили красный. Какое количество информации при этом было получено?

Решение.

Так как вытаскивание карандаша любого цвета из имеющихся в коробке 32 карандашей является равновероятным, то число возможных событий

равно 32.

N = 32, I = ?

$N = 2^I$, $32 = 2^5$, I = 5 бит.

Ответ: 5 бит.

Задачи на перевод единиц измерения информации

***Задача 1.* В школьной библиотеке 16 стеллажей с книгами, на каждом – по 8 полок. Ученику сообщили, что нужный учебник находится на 2-ой полке 4-го стеллажа. Какое количество информации получил ученик?

Решение.

1) Число стеллажей (случаев) – 16.

$N1 = 16$, N1 = 2I, $16 = 2^I$, $16 = 2^4$, I1= 4 бита.

2) Число полок на каждом стеллаже (случаев) – 8,

$N2 = 8$, N2 = 2I, $8 = 2^3$, I2 = 3 бит.

3) I = I1 + I2, I = 4 бита + 3 бита = 7 бит.

Ответ: 7 бит.

**Задача 3.* Загадывают число в диапазоне от 1 до 200. Какое наименьшее количество вопросов надо задать, чтобы наверняка отгадать число. На вопросы можно отвечать только «Да» или «Нет».

Решение.

Правильная стратегия состоит в том, чтобы количество вариантов каждый раз уменьшалось вдвое.

Например, загадано число 152.

1 вопрос: Число >100? Да.

2 вопрос: Число < 150? Нет.

3 вопрос: Число > 175? Нет. и т.д.

Количество событий в каждом варианте будет одинаково, и их отгадывание равновероятно. $N = 2^I$, $200 = 2^I$, $7 < I < 8$. Т.к. количество вопросов нецелым числом быть не может, то необходимо задать не более 8 вопросов.

Ответ: 8 вопросов

**Вычисление количества информации для событий с различными вероятностями.

Существует множество ситуаций, когда возможные события имеют различные вероятности реализации. Рассмотрим примеры таких событий.

В коробке 20 карандашей, из них 15 красных и 5 чёрных. Вероятность вытащить наугад красный карандаш больше, чем чёрный.

При случайном падении бутерброда вероятность падения его маслом вниз (более тяжёлой стороной) больше, чем маслом вверх.

В пруду живут 8000 карасей, 2000 щук и 40000 пескарей. Самая большая вероятность для рыбака – поймать в этом пруду пескаря, на втором месте – карася, на третьем – щуку.

Количество информации в сообщении о некотором событии зависит от его вероятности. Чем меньше вероятность события, тем больше информации оно несёт.

$P = K/N$, где К – количество случаев реализации одного из исходов события, N – общее число возможных исходов одного из событий

$2^I = \log2(1/p)$, где I – количество информации, p – вероятность события

Задача. В коробке 50 шаров, из них 40 белых и 10 чёрных. Определить количество информации в сообщении о вытаскивании наугад белого шара и чёрного шара.

Решение.

Вероятность вытаскивания белого шара

P1 = 40/50 = 0,8

Вероятность вытаскивания чёрного шара P2 = 10/50 = 0,2

Количество информации о вытаскивании белого шара

$I_1 = \log2(1/0,8) = \log21,25 = \log1,25/\log2 \approx 0,32$ бит

Количество информации о вытаскивании чёрного шара

$I_2 = \log2(1/0,2) = \log25 = \log5/\log2 \approx 2,32$ бит

Ответ: 0,32 бит, 2,32 бит

Что такое логарифм?

Логарифмом числа a по основанию b называется показатель степени, в которую надо возвести число a, чтобы получить число b.

$a^{\log_a b} = b$, $a > 0$, $b > 0$, $a \neq 1$

Вычисление логарифмов чисел по основанию 2 с помощью электронного калькулятора

$\log_2 6 = \log 6 / \log 2$, где log 6 и log 2 – десятичные логарифмы

Программа вычисления логарифма числа 6 по основанию 2 ($\log_2 6$) с помощью инженерного калькулятора: 6, log, / , 2, log, =

Количество информации в случае различных вероятностей событий определяется по формуле:

Формула Шеннона: (американский учёный, 1948 г.)

где Pi – вероятность i-го события, N – количество возможных событий

$$I=- \sum_{i=1}^{N} p_i log_2 p_i$$

Задача. В озере живут караси и окуни. Подсчитано, что карасей 1500, а окуней - 500. Сколько информации содержится в сообщениях о том, что рыбак поймал карася, окуня, поймал рыбу?

Решение.

События поимки карася или окуня не являются равновероятными, так как окуней в озере меньше, чем карасей.

Общее количество карасей и окуней в пруду 1500 + 500 = 2000.

Вероятность попадания на удочку карася

$p_1 = 1500/2000 = 0,75$, окуня p_2 – 500/2000 = 0,25.

$I_1 = \log_2(1/p1)$, $I_1 = \log2(1/p2)$, где I_1 и I_2 – вероятности поймать карася и окуня соответственно.

$I_1 = \log_2(1 / 0,75) \approx 0,43$ бит, $I_2 = \log2(1 / 0,25) \approx 2$ бит – количество информации в сообщении поймать карася и поймать окуня соответственно.

Количество информации в сообщении поймать рыбу (карася или окуня) рассчитывается по формуле Шеннона

$I = - p_1 \log_2 p1 - p_2 \log_2 p2$

$I = - 0,75*\log 20,75 - 0,25*\log 20,25 = - 0,75*(\log 0,75/\log 2) - 0,25*(\log 0,25/\log 2) =$

$= 0,604$ бит ≈ 0.6 бит.

Ответ: в сообщении содержится 0,6 бит информации

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

наименование работы;

цель работы;

задание;

последовательность выполнения работы;

ответы на контрольные вопросы;

вывод о проделанной работе.

Контрольные вопросы

1. Какое количество информации несет в себе жесткий диск емкостью 4 терабайта, если производитель рассчитывает 1000 за 1024?

2. Чем отличается вероятностный подход к измерению информации от алфавитного?

3. Какие единицы измерения информации используют для флэш-накопителей?

Практическое занятие № 3

Способы измерения информации. Скорость передачи информации.

Цель: научиться измерять и вычислять скорость передачи информацию.

Оборудование: ПК.

Программное обеспечение: операционная система, текстовый редактор.

Теоретические основы

Скорость передачи информации определяется количеством элементов двоичной информации,, передаваемых за 1 с. В синхронной передачи данных по коммутируемым каналам рекомендуется использовать скорости из следующего ряда: 600, 1200, 2400, 4800, 9600 бит/с. Для асинхронной передачи по коммутируемым каналам рекомендуется скорость 300 бит/с.

В синхронной передачи данных по арендованным каналам рекомендуется использовать скорости из следующих рядов:

а) предпочтительные: 600, 1200, 2400, 4800, 9600, 14400 бит/с;

б) дополнительные: 3000, 6000, 7200, 12000 бит/с;

в) допустимого диапазона: Nx600 бит/с (1<=N<=24)

Следует отличать скорость передачи информации от модуляционной (линейной) скорости, измеряемой в Бодах (количество элементов модулированного сигнала, передаваемого за 1 с). Для простых видов модуляции скорость передачи информации совпадает с модуляционной скоростью.

При синхронной передаче скорость должна отличаться от номинального значения не более, чем на 0,01%, а при асинхронной – не более, чем на 2,5%.

Оба компьютера, как правило, могут одновременно обмениваться информацией в обе стороны. Этот режим работы называется полным дуплексным.

Дуплексный режим передачи данных – режим, при котором передача данных осуществляется одновременно в обоих направлениях.

В отличие от дуплексного режима передачи данных, полудуплексный подразумевает передачу в каждый момент времени только в одном направлении.

Кроме собственно модуляции и демодуляции сигналов модемы могут выполнять сжатие и декомпрессию пересылаемой информации, а также заниматься поиском и исправлением ошибок, возникнувших в процессе передачи данных по линиям связи.

Одной из основных характеристик модема является скорость модуляции (modulation speed), которая определяет физическую скорость передачи данных без учета исправления ошибок и сжатия данных. Единицей измерения этого параметра является количество бит в секунду (бит/с), называемое бодом.

Любой канал связи имеет ограниченную пропускную способность (скорость передачи информации), это число ограничивается свойствами аппаратуры и самой линии (кабеля).

Объем переданной информации вычисляется по формуле Q=q*t, где q – пропускная способность канала (в битах в секунду), а t – время передачи

Пример 1. Скорость передачи данных через ADSL-соединение равна 128000 бит/с. Через данное соединение передают файл размером 625 кбайт. Определить время передачи файла в секундах.

Решение:

1) выделим в заданных больших числах степени двойки и переведем размер файла в биты, чтобы «согласовать» единиц измерения:

128000 бит/с = 128 · 1000 бит/с = 27 · 125 · 8 бит/с = 27 · 53 · 23 бит/с = 210·53 бит/с

625 кбайт = 54 кбайт = 54 · 213 бит.

2) чтобы найти время передачи в секундах, нужно разделить размер файла на скорость передачи:

t=(54 · 213)бит / 210·53 бит/с = 40 с.

Ответ: 40 с .

Пример 2. Скорость передачи данных через ADSL-соединение равна 512000 бит/с. Передача файла через это соединение заняла 1 минуту. Определить размер файла в килобайтах.

Решение:

1) выделим в заданных больших числах степени двойки; переведем время в секунды (чтобы «согласовать» единицы измерения), а скорость передачи – в кбайты/с, поскольку ответ нужно получить в кбайтах:

1 мин = 60 с = 4 · 15 с = 22 · 15 с

512000 бит/с = 512 · 1000 бит/с = 29 · 125 · 8 бит/с = 29 · 53 · 23 бит/с = 212 · 53 бит/с = 29 · 53 бит/с = (29 · 53) / 210 кбайт/с = (53 / 2) кбайт/с

2) чтобы найти объем файла, нужно умножить время передачи на скорость передачи:

Q=q*t = 22 · 15 с *(53 / 2) кбайт/с = 3750 кбайт

Ответ: 3750 кбайт.

Пример 3. С помощью модема установлена связь с другим компьютером со скоростью соединения 19200, с коррекцией ошибок и сжатием данных.

а) Можно ли при таком соединении файл размером 2,6 килобайт передать за 1 секунду? Обоснуйте свой ответ.

б) Всегда ли при таком соединении файл размером 2,3 килобайт будет передаваться за 1 секунду? Обоснуйте свой ответ.

в) Можно ли при таком соединении оценить время передачи файла размером 4 Мб? Если можно, то каким образом?

Решение:

а) Для начала узнаем, какое количество килобайт мы можем передать за 1 секунду: 19200/1024/8 = 2,3 (Кбайт). Следовательно, если бы не было сжатия информации, то данный файл за одну секунду при данной скорости соединения было бы невозможно передать. Но сжатие есть, 2.6/2.3 < 4, следовательно, передача возможна.

б) Нет не всегда, так как скорость соединения это максимально возможная скорость передачи данных при этом соединении. Реальная скорость может быть меньше.

в) Можно указать минимальное время передачи этого файла: 4*1024*1024/4/19200, около 55 с (столько времени будет передаваться файл на указанной скорости с максимальной компрессией). Максимальное же время передачи оценить вообще говоря нельзя, так как в любой момент может произойти обрыв связи.

Практические задания

Таблица 1. Задание 1. Решите задачу о передаче информации с помощью модема.

Вариант 1	Скорость передачи данных через ADSL-соединение равна 512000 бит/с. Через данное соединение передают файл размером 1500 Кб. Определите время передачи файла в секундах.

Вариант 2	Скорость передачи данных через ADSL-соединение равна 1024000 бит/с. Через данное соединение передают файл размером 2500 Кб. Определите время передачи файла в секундах.
Вариант 3	Скорость передачи данных через ADSL-соединение равна 1024000 бит/с. Передача файла через данное соединение заняла 5 секунд. Определите размер файла в килобайтах.
Вариант 4	Скорость передачи данных через ADSL-соединение равна 512000 бит/с. Передача файла через данное соединение заняла 8 секунд. Определите размер файла в килобайтах.

Задание 2. Решите задачу о передаче графической информации.

Вариант 1	Определите скорость работы модема, если за 256 с он может передать растровое изображение размером 640х480 пикселей. На каждый пиксель приходится 3 байта.
Вариант 2	Сколько секунд потребуется модему, передающему информацию со скоростью 56 000 бит/с, чтобы передать цветное растровое изображение размером 640 х 480 пикселей, при условии, что цвет каждого пикселя кодируется тремя байтами?
Вариант 3	Определите скорость работы модема, если за 132 с он может передать растровое изображение размером 640х480 пикселей. На каждый пиксель приходится 3 байта.
Вариант 4	Сколько секунд потребуется модему, передающему информацию со скоростью 28800 бит/с, чтобы передать цветное растровое изображение размером 640 х 480 пикселей, при условии, что цвет каждого пикселя кодируется тремя байтами?

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;
- вывод о проделанной работе.

Контрольные вопросы

1. Дайте определение пропускной способности (передача информации)?

2. В чем измеряется пропускная способность?

3. Зависит ли скорость передачи информации от объема передаваемой информации?

Практическое занятие № 4

Способы измерения информации. Поиск энтропии случайных величин.

Цель: научиться вычислять энтропию случайной величины.

Оборудование: ПК.

Программное обеспечение: операционная система, калькулятор, текстовый редактор.

Теоретические основы

Энтропия в теории информации — мера хаотичности информации, неопределённость появления какого-либо символа первичного алфавита. При отсутствии информационных потерь численно равна количеству информации на символ передаваемого сообщения.

Так, возьмём, например, последовательность символов, составляющих какое-либо предложение на русском языке. Каждый символ появляется с разной частотой, следовательно, неопределённость появления для некоторых символов больше, чем для других. Если же учесть, что некоторые сочетания символов встречаются очень редко, то неопределённость ещё более уменьшается (в этом случае говорят об энтропии n-ого порядка. Концепции информации и энтропии имеют глубокие связи друг с другом, но, несмотря на это, разработка теорий в статистической механике и теории информации заняла много лет, чтобы сделать их соответствующими друг другу.

Энтропия независимых случайных событий x с n возможными состояниями (от 1 до n) рассчитывается по формуле:

$$H(x) = -\sum_{i=1}^{n} p_{(i)} log_2 p_{(i)}$$

Эта величина также называется *средней энтропией сообщения*. Величина $log_2 \frac{1}{p_{(i)}}$ называется *частной энтропией*, характеризующей только i-е состояние.

Таким образом, энтропия события x является суммой с противоположным знаком всех произведений относительных частот появления события i, умноженных на их же двоичные логарифмы (основание 2 выбрано только для удобства работы с информацией, представленной в двоичной форме). Это определение для дискретных

случайных событий можно расширить для функции распределения вероятностей.

Шеннон вывел это определение энтропии из следующих предположений:

мера должна быть непрерывной; т. е. изменение значения величины вероятности на малую величину должно вызывать малое результирующее изменение энтропии;

в случае, когда все варианты (буквы в приведенном примере) равновероятны, увеличение количества вариантов (букв) должно всегда увеличивать полную энтропию;

должна быть возможность сделать выбор (в нашем примере букв) в два шага, в которых энтропия конечного результата должна будет является суммой энтропий промежуточных результатов.

Шеннон показал, что любое определение энтропии, удовлетворяющее этим предположениям, должно быть в форме:

$$-K \sum_{i=1}^{n} p_{(i)} log_2 p_{(i)}$$

где K — константа (и в действительности нужна только для выбора единиц измерения).

Шеннон определил, что измерение энтропии ($H = -p_1 \log_2 p_1 - \ldots - p_n \log_2 p_n$), применяемое к источнику информации, может определить требования к минимальной пропускной способности канала, требуемой для надежной передачи информации в виде закодированных двоичных чисел. Для вывода формулы Шеннона необходимо вычислить математическое ожидания «количества информации», содержащегося в цифре из источника информации. Мера энтропии Шеннона выражает неуверенность реализации случайной переменной. Таким образом, энтропия является разницей между информацией, содержащейся в сообщении, и той частью информации, которая точно известна (или хорошо предсказуема) в сообщении. Примером этого является избыточность языка — имеются явные статистические закономерности в появлении букв, пар последовательных букв, троек и т.д.

В общем случае b-арная энтропия (где b равно 2,3,...) источника $\mathcal{S} = (S,P)$ с исходным алфавитом $S = \{a_1, \ldots, a_n\}$ и дискретным распределением вероятности $P = \{p_1, \ldots, p_n\}$ где p_i является вероятностью a_i ($p_i = p(a_i)$) определяется формулой:

$H_b(S) = -\sum_{i=1}^{n} p_{(i)} log_b p_{(i)}$

Определение энтропии Шеннона очень связано с понятием термодинамической энтропии. Больцман и Гиббс проделали большую работу по статистической термодинамике, которая способствовала принятию слова «энтропия» в информационную теорию. Существует связь между понятиями энтропии в термодинамике и теории информации. Например, демон Максвелла также противопоставляет термодинамическую энтропию информации, и получение какого-либо количества информации равно потерянной энтропии.

СВОЙСТВА ЭНТРОПИИ

*1.*Энтропия является вещественной и неотрицательной величиной.

*2.*Энтропия – величина ограниченная.

*3.*Энтропия обращается в нуль лишь в том случае, если вероятность одного из состояний равна единице; тогда вероятности всех остальных состояний, естественно, равны нулю. Это положение соответствует случаю, когда состояние источника полностью определено.

*4.*Энтропия максимальна, когда все состояния источника равновероятны.

5. Энтропия источника и с двумя состояниями u1 и u2 изменяется от нуля до единицы, достигая максимума при равенстве их вероятностей:

p(u1) = p= p(u2) = 1 — p = 0,5.

*6.*Энтропия объединения нескольких статистически независимых источников информации равна сумме энтропии исходных источников.

7. Энтропия характеризует среднюю неопределенность выбора одного состояния из ансамбля. При ее определении используют только вероятности состояний, полностью игнорируя их содержательную сторону. Поэтому энтропия не может служить средством решения любых задач, связанных с неопределенностью.

8. Энтропия как мера неопределенности согласуется с экспериментальными данными, полученными при изучении психологических реакций человека, в частности реакции выбора. Установлено, что время безошибочной реакции на последовательность беспорядочно чередующихся равновероятных раздражителей (например, загорающихся лампочек) растет с увеличением их числа так же, как

энтропия. Это время характеризует неопределенность выбора одного раздражителя. Замена равновероятных раздражителей неравновероятными приводит к снижению среднего времени реакции ровно настолько, насколько уменьшается энтропия.

Дифференциальной энтропией случайной величины X называется величина:

$$H_Д(x)=H(x)-H(y)= -\int_{-\infty}^{+\infty} p_x(x) * \log_2 d * p_x(x)dx$$

Если произвести квантование случайных величин X_1, $X_2 \ldots X_n$ по уровню с числом уровней квантования равным m , то возможное число реализаций длительностью T_n станет конечным и равным $M = m^n$.

Каждая из реализаций C_1, $C_2, \ldots . C_i, \ldots C_m$ будет иметь определенную вероятность появления в эксперименте по наблюдению реализаций. Тогда неопределенность (энтропия) и количество информации в реализации (в среднем по всем реализациям) определяются равенством

$$H_n = -\sum_{i=1}^{M} P(C_i) \log(P(C_i))$$

$$H = \frac{H_n}{n} = -\frac{1}{n}\sum_{i=1}^{M} P(C_i) \log(P(C_i))$$

Энтропия и количество информации на одну степень свободы (на одну выборку) равны

$$H = \frac{H_n}{n} = -\frac{1}{n}\sum_{i=1}^{M} P(C_i) \log(P(C_i))$$

Избыточность показывает, какая доля максимально возможной при заданном объеме алфавита неопределенности не используется источником.

$\mu=(H_{max}-H_u)/H_{max}$,

Где H_u – энтропия рассматриваемого источника, H_{max} – максимально возможное значение его энтропии, которое может быть достигнуто подбором распределения и ликвидацией взаимозависимости элементов алфавита. Так, для дискретного источника с M элементами

$H_{max}=\log M$

Выполнение расчетных задач

Задача №1

Доказать, что H(x,y)≤H(x)+H(y).

Решение:

По определению,

$H(x,y) = \sum_{i=1}^{n} \sum_{j=1}^{m} p(x_i,y_j) \log p(x_i,y_j)$

Для статистически зависимых событий

$p(x_i, y_i) = p(x_i)\, p(x_i/y_i)$.

$H(x,y) = \sum_{i=1}^{n} \sum_{j=1}^{m} p(x_i,y_j)[\log p(x_i) + \log p(x_i/y_i)] =$

$= -\sum_{i=1}^{n} \sum_{j=1}^{m} p(x_i,y_j)[\log p(x_i) + \log p(x_i/y_i)] =$

$= -\sum_{i=1}^{n} p(x_i) \log p(x_i) \sum_{j=1}^{m} p(y_i/x_i)] + \sum_{i=1}^{n} p(x_i)(-\sum_{j=1}^{m} p(y_i/x_i) \log p(y_i/x_i))$,

Тогда

$\sum_{j=1}^{m} p(yi/xi)] = 1, \quad -\sum_{j=1}^{m} p(yi/xi) \log p(yi/xi) = H(x,y)$

$H(x,y) = -\sum_{i=1}^{n} p(x_i) \log p(x_i) + \sum_{j=1}^{m} p(y_i/x_i) H(x,y/x_i) = H(x) + H(y/x_i))$

$H(y/x_i)$- это частная энтропия Y при условии, что известно состояние $X=x_i$.

Наличие информации о состоянии X не может увеличить неопределенность состояния Y, но может уменьшить его в случае зависимости Y от X. Значит, условная энтропия $H(y/x_i)$ не больше безусловной энтропии $H(y)$, то есть $H(y/x_i) \leq H(y)$. Тогда средняя условная энтропия

$H(x,y) = -\sum_{i=1}^{n} p(x_i) \quad H(y/x_i) \leq \sum_{i=1}^{n} p(x_i) \quad H(y) = H(y) \sum_{i=1}^{n} p(x_i) = H(y)$, *так как* $\sum_{i=1}^{n} p(x_i) = 1$,

то есть $H(y/x_i) \leq H(y)$.

Значит, $H(x,y) = H(x) + H(x/y) \leq H(x) + H(y)$.

Задача №2

Показать, что для регулярной марковской цепи энтропия $H(x)^{(r)}$ за r шагов равняется энтропии за один шаг, умноженной на число шагов r.

Решение:

Регулярная цепь Маркова полностью характеризуется матрицей переходных вероятностей $p=\begin{pmatrix} p_{11} & \cdots & p_{1m} \\ \vdots & \ddots & \vdots \\ p_{m1} & \cdots & p_{mm} \end{pmatrix}$ и предельным стационарным распределением вероятностей состояний $(p_1, p_2, \dots, p_m)$

В стационарном режиме энтропия за один шаг не зависит от номера шага и равна $H(y)= \sum_{k=1}^{m} p_k H_k(x)$,

p_k - стационарная вероятность k-го состояния,

$H_k(y)=-\sum_{k=1}^{m} p_{ki} \log p_{ki}$ - энтропия в k-м состоянии.

Энтропия за r шагов равна сумме энтропий за каждый шаг. Так как энтропия за каждый шаг одинакова, то сумма энтропий равна $H(X)^{(r)}=r* H(X)^{(1)}$.

Задача 3

В результате полной дезорганизации управления m самолетов летят произвольными курсами. Управление восстановлено, и все самолеты взяли общий курс со среднеквадратической ошибкой отклонения от курса $\sigma=3^0$. Найти изменение энтропии, считая, что в первом случае имело место равномерное распределение вероятностей углов, а во втором случае – нормальное.

Ответ: 4.86 бита

Решение.

Начальное распределение вероятностей углов курсов самолетов равномерное в интервале от $a=0$ до $b=360°=2\pi\ рад$ с плотностью вероятности $p_{1X}(x)=\frac{1}{b-a},\ a\le x\le b$.

Дифференциальная энтропия этого распределения

$$H_{1Д}(x)=-\int_a^b \frac{1}{b-a}\log_2\frac{1}{b-a}dx=-\log_2\frac{1}{b-a}=\log_2(b-a)=\log_2 2\pi=2,65 \text{ бит.}$$

Конечное распределение вероятностей углов курсов самолетов нормальное с параметрами $a=0,\ \sigma=3°=\frac{\pi}{60}\ рад$ и плотностью вероятности $p_{2X}(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{-x^2/(2\sigma^2)}$.

Дифференциальная энтропия этого распределения

$$H_{2Д}(x) = -\int_{-\infty}^{\infty} p_{2X}(x)\log_2\left[\frac{1}{\sigma\sqrt{2\pi}}e^{-x^2/(2\sigma^2)}\right]dx =$$

$$= -\log_2\frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{\infty} p_{2X}(x)dx + \frac{\log_2 e}{2\sigma^2}\int_{-\infty}^{\infty} x^2 p_{2X}(x)dx =$$

$$= -\log_2\frac{1}{\sigma\sqrt{2\pi}} + \frac{\log_2 e}{2\sigma^2}\cdot\sigma^2 = \log_2\sigma\sqrt{2\pi} + \frac{\log_2 e}{2} = \log_2\frac{\pi\sqrt{2\pi e}}{60} = -2,21 \text{ } бит.$$

Изменение энтропии $\Delta H(x) = H_{2Д}(x) - H_{1Д}(x) = -2,21 - 2,65 = -4,86$ бит.

Энтропия уменьшилась на 4,86 бит.

Задача 4

Измерительное устройство вырабатывает временные интервалы, распределенные случайным образом в пределах от 100 до 500 мс. Как изменится энтропия случайной величины при изменении точности измерения с 1 мс до 1 мкс?

Ответ: Энтропия увеличивается примерно на 10 бит

Решение.

При точности 1мс дискретная случайная величина X – результат измерения – может равновероятно принимать одно из $n = \frac{500-100}{1} = 400$ значений. Энтропия равна $H_1(x) = \log_2 n$.

При точности 1мкс дискретная случайная величина X – результат измерения – может равновероятно принимать одно из $m = \frac{500-100}{10^{-3}} = 400\cdot 10^3 = 1000n$ значений. Энтропия равна $H_2(x) = \log_2 m$.

Изменение энтропии

$$\Delta H(x) = H_2(x) - H_1(x) = \log_2 m - \log_2 n = \log_2 1000n - \log_2 n = \log_2 1000 \approx \log_2 1024 = 10$$

бит.

Энтропия увеличилась примерно на 10 бит.

Задача 6

Записать отношения между энтропиями:

H(x), H(y), H(x|y), H(y|x), H(x,y), $H(x|y_j)$, $H(y|x_i)$

Решение.

Связь между энтропией совместного распределения и полными энтропиями и средними условными энтропиями

$H(x,y)=H(x)+H(y/x)=H(y)+H(x/y)$.

Если *x* и *y* независимы, то $H(y/x)=H(y),\ H(x/y)=H(x)$ и энтропия совместного распределения $H(x,y)=H(x)+H(y)$ максимальна, так как $H(y/x)\le H(y),\ H(x/y)\le H(x)$.

Связь между частными и средними условными энтропиями

$$H(y/x)=\sum_{i=1}^{n}p(x_i)H(y/x_i),\ \ H(x/y)=\sum_{j=1}^{m}p(y_j)H(x/y_j).$$

Задача 7

На рисунке представлена диаграмма канала со слабым разрешением. Определить количество информации, передаваемое по каналу.

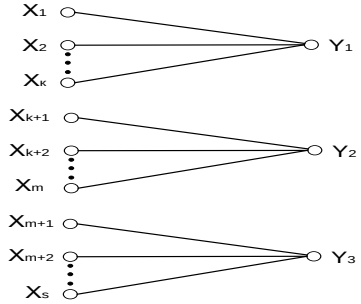


Рисунок 4.1 Диаграмма канала1.

Ответ: I (x,y)=H(x)

Решение:

Количество переданной по каналу информации равно $I=H-H_{апост.}$.

$H=H(y)$ – энтропия полученных элементов *y* до отправления элемента *x*; $H_{апост.}=H(y/x)$ - энтропия полученных элементов *y* после отправления элемента *x*. Для данного канала со слабым разрешением $H(y/x)=0$, так как полученный элемент *y* однозначно определяется по отправленному элементу *x*.

Получаем $I=H(y)$.

Задача 2

На рисунке представлена диаграмма канала с неоднозначностью. Определить количество информации, передаваемое по каналу.

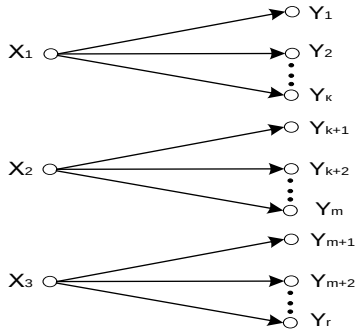


Рисунок 4.2 Диаграмма канала 2.

ответ: I (x,y)=H(x)

Решение:

Количество переданной по каналу информации равно $I = H - H_{апост.}$.

$H = H(x)$ – энтропия отправленных элементов *x* до получения элемента *y*; $H_{апост.} = H(x/y)$ - энтропия отправленных элементов *x* после получения элемента *y*. Для данного канала с неоднозначностью $H(x/y) = 0$, так как полученный элемент *y* однозначно определяет отправленный элемент *x*.

Получаем $I = H(x)$.

Задачи по вычислению энтропии

1. Найдите энтропию для числа белых шаров при извлечении двух шаров из урны, содержащей два белых и один черный шар.
2. Найдите энтропию для числа козырных карт при извлечении двух карт из колоды в 36 карт.
3. Какую степень неопределенности содержит опыт угадывания суммы очков на извлеченной кости из полного набора домино?
4. Найдите энтропию для числа тузов при извлечении трех карт из карт с картинками.
5. Найдите дифференциальную энтропию для равномерного распределения.
6. Найдите дифференциальную энтропию для показательного закона распределения, если известно, что случайная величина *x* принимает значение меньше единицы с вероятностью 0,5.

Отчет

- Отчет должен быть оформлен в текстовом редакторе и содержать:
- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;
- вывод о проделанной работе.

В нижний колонтитул поместить фамилию, инициалы и номер группы обучаемого (8 пт., Arial, выравнивание по правому краю).

Контрольные вопросы

1. Как определяется энтропия дискретных случайных величин?
2. Приведите примеры энтропий для классических законов распределения.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 5

ПРИМЕНЕНИЕ ТЕОРЕМЫ ОТЧЕТОВ.

Цель: Изучение возможности синтезирования сигналов по дискретным отсчетам в соответствии с теоремой Котельникова.

Оборудование: ПК.

Программное обеспечение: операционная система, калькулятор, текстовый редактор.

Теоретические основы

Теорема Котельникова

В 1933 году В.А. Котельниковым доказана теорема отсчетов [6, 32], имеющая важное значение в теории связи: непрерывный сигнал $s(t)$ с ограниченным спектром можно точно восстановить (интерполировать) по его отсчетам $s(k\Delta t)$, взятым через интервалы $\Delta t = \frac{1}{(2F)}$, где F – верхняя частота спектра сигнала.

В соответствии с этой теоремой сигнал $s(t)$ можно представить рядом Котельникова

$$s(t) = \sum_{k=-\infty}^{\infty} s\left(\frac{k}{2F}\right) \frac{\sin 2\pi F\left[t - \frac{k}{2F}\right]}{2\pi F\left[t - \frac{k}{2F}\right]} \qquad (1.21)$$

Таким образом, сигнал $s(t)$, можно абсолютно точно представить с помощью последовательности отсчетов $s\left(\frac{k}{2F}\right)$, заданных в дискретных точках $\frac{k}{2F}$ (рис.1.16).

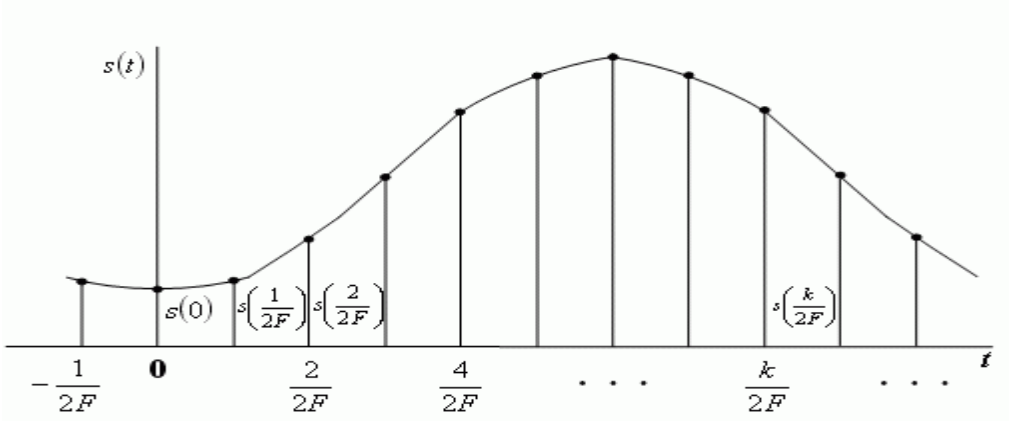


Рис. 1.16. Сигнал и его отсчеты

Функции

$$\varphi(t)=\frac{\sin 2\pi F\left[t-\frac{k}{2F}\right]}{2\pi F\left[t-\frac{k}{2F}\right]} \qquad (1.22)$$

образуют ортогональный базис в пространстве сигналов, характеризующихся ограниченным спектром:

$\Phi(f)=0$ при $|f|>F$. (1.23)

Обычно для реальных сигналов можно указать диапазон частот, в пределах которого сосредоточена основная часть его энергии и которым определяется ширина спектра сигнала. В ряде случаев спектр сознательно сокращают. Это обусловлено тем, что аппаратура и линия связи должны иметь минимальную полосу частот. Сокращение спектра выполняют, исходя из допустимых искажений сигнала. Например, при телефонной связи хорошая разборчивость речи и узнаваемость абонента обеспечиваются при передаче сигналов в полосе частот $\Delta F=0{,}3\ldots3{,}4\ [\text{кГц}]$. Увеличение ΔF приводит к неоправданному усложнению аппаратуры и повышению затрат. Для передачи телевизионного изображения при стандарте в 625 строк полоса частот, занимаемая сигналом, составляет около 6 МГц.

Из вышесказанного следует, что процессы с ограниченными спектрами могут служить адекватными математическими моделями многих реальных сигналов.

Функция вида $\frac{\sin 2\pi F\left[t-\frac{k}{2F}\right]}{2\pi F\left[t-\frac{k}{2F}\right]}$ называется функцией отсчетов (рис.1.17).

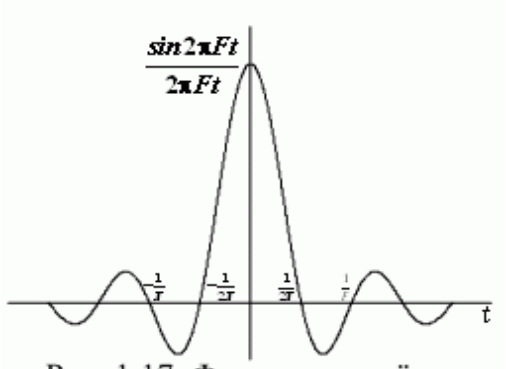


Рис. 1.17. Функция отсчётов

Она характеризуется следующими свойствами. Если $k = 0$, функция отсчетов имеет максимальное значение при $t = 0$, а в моменты времени $t = \frac{i}{2F}$ ($i = 1,2,\ldots$) она обращается в нуль; ширина главного лепестка функции отсчетов на нулевом уровне равна $\frac{1}{F}$, поэтому минимальная длительность импульса, который может существовать на выходе линейной системы с полосой пропускания F, равна $\frac{1}{F}$; функции отсчетов ортогональны на бесконечном интервале времени.

На основании теоремы Котельникова может быть предложен следующий способ дискретной передачи непрерывных сигналов:

Для передачи непрерывного сигнала $s(t)$ по каналу связи с полосой пропускания F определим мгновенные значения сигнала $s(t)$ в дискретные моменты времени $t_k = \frac{1}{2F}$, ($k = 0,1,2,\ldots$). После этого передадим эти значения по каналу связи каким - либо из возможных способов и восстановим на приемной стороне переданные отсчеты. Для преобразования потока импульсных отсчетов в непрерывную функцию пропустим их через идеальный ФНЧ с граничной частотой F.

Можно показать, что энергия сигнала находится по формуле [6, 32]:

$$E = \int_{-\infty}^{\infty} s^2(t)dt = \frac{1}{2F}\sum_{k=-\infty}^{\infty} s^2\left(\frac{k}{2F}\right). \qquad (1.24)$$

Для сигнала, ограниченного во времени, выражение (1.24) преобразуется к виду:

$$E = \int_{1}^{2FT} s^2(t)dt = \frac{1}{2F}\sum_{k=1}^{2FT} s^2\left(\frac{k}{2F}\right). \qquad (1.25)$$

Выражение (1.25) широко применяется в теории помехоустойчивого приема сигналов, но является приближенным, т.к. сигналы не могут быть одновременно ограничены по частоте и времени.

Практическое задание

1. Изобразить сигналы, синтезируемые в лабораторной работе:

а) синусоидальный сигнал частотой 5кГц;

б) видеоимпульсы прямоугольной формы длительностью 0,25; 0,5; 1,0 мс;

в) видеоимпульсы пилообразной формы длительностью 0,5 мс; 1,0 мс.

2. Рассчитать и построить идеальные выборочные сигналы для сигналов, указанных в п. 1а, 1б, 1в, при $f_{выб}$=5, 10, 20, 40 кГц.

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;
- вывод о проделанной работе.
- В нижний колонтитул поместить фамилию, инициалы и номер группы обучаемого (8 пт., Arial, выравнивание по правому краю).

Контрольные вопросы

1. Сформулируйте теорему Котельникова для сигналов с ограниченным спектром.

2. Объясните погрешности синтезирования реальных сигналов по дискретным отсчетам.

Практическое занятие № 6

Смысл энтропии Шеннона. Расчет вероятностей.

Цель: научиться вычислять вероятности событий (появление символов в сообщении) и рассчитывать энтропию.

Оборудование: ПК.

Программное обеспечение: операционная система, калькулятор, текстовый редактор.

Теоретические основы

Количество информации по Хартли и Шеннону

Понятие количество информации отождествляется с понятием информация. Эти два понятия являются синонимами. Мера информации должна монотонно возрастать с увеличением длительности сообщения (сигнала), которую естественно измерять числом символов в дискретном сообщении и временем передачи в непрерывном случае. Кроме того, на содержание количества информации должны влиять и статистические характеристики, так как сигнал должен рассматриваться как случайный процесс.

При этом наложено ряд ограничений:

1. Рассматриваются только дискретные сообщения.

2. Множество различных сообщений конечно.

3. Символы, составляющие сообщения равновероятны и независимы.

Хартли впервые предложил в качестве меры количества информации принять логарифм числа возможных последовательностей символов.

$$I=\log m^k=\log N \qquad (1)$$

К.Шеннон попытался снять те ограничения, которые наложил Хартли. На самом деле в рассмотренном выше случае равной вероятности и независимости символов при любом k все возможные сообщения оказываются также равновероятными, вероятность каждого из таких сообщений равна P=1/N. Тогда количество информации можно выразить через вероятности появления сообщений $I=-\log P$.

В силу статистической независимости символов, вероятность сообщения длиной в k символов равна

$$P = \prod_{i=1}^{k} p_i$$

Если i–й символ повторяется в данном сообщении k_i раз, то $P = \prod_{i=1}^{m} p_i^{k_i}$

так как при повторении i символа k_i раз k уменьшается до m. Из теории вероятностей известно, что, при достаточно длинных сообщениях (большое число символов k) $k_i \approx k \cdot p_i$ и тогда вероятность сообщений будет равняться $P = \prod_{i=1}^{m} p_i^{kp_i}$

Тогда окончательно получим

$$I = -\log P = -k\sum_{i=1}^{m} p_i \log p_i \qquad (2)$$

Данное выражение называется формулой Шеннона для определения количества информации.

Формула Шеннона для количества информации на отдельный символ сообщения совпадает с энтропией. Тогда количество информации сообщения состоящего из k символов будет равняться $I = k \cdot H$

Количество информации, как мера снятой неопределенности

При передаче сообщений, о какой либо системе происходит уменьшение неопределенности. Если о системе все известно, то нет смысла посылать сообщение. Количество информации измеряют уменьшением энтропии.

Количество информации, приобретаемое при полном выяснении состояния некоторой физической системы, равно энтропии этой системы:

$$I = -\sum_{i=1}^{n} p_i \log p_i$$

Количество информации I – есть осредненное значение логарифма вероятности состояния. Тогда каждое отдельное слагаемое $-\log p_i$ необходимо рассматривать как частную информацию, получаемую от отдельного сообщения, то есть

$$I_i = -\log p_i$$

Избыточность информации

Если бы сообщения передавались с помощью равновероятных букв алфавита и между собой статистически независимых, то энтропия таких сообщений была бы максимальной. На самом деле реальные сообщения строятся из не равновероятных букв алфавита с

наличием статистических связей между буквами. Поэтому энтропия реальных сообщений –H_p, оказывается много меньше оптимальных сообщений – H_o. Допустим, нужно передать сообщение, содержащее количество информации, равное I. Источнику, обладающему энтропией на букву, равной H_p, придется затратить некоторое число n_p, то есть

$$I = n_p H_p$$

Если энтропия источника была бы H_0, то пришлось бы затратить меньше букв на передачу этого же количества информации

$$n_o = \frac{I}{H_o} < n_p$$

I= n_0H_0

Таким образом, часть букв $n_p – n_o$ являются как бы лишними, избыточными. Мера удлинения реальных сообщений по сравнению с оптимально закодированными и представляет собой избыточность D.

$$D = 1 - \frac{H_p}{H_o} = 1 - \frac{n_o}{n_p} = \frac{n_p - n_o}{n_p} \qquad (3)$$

Но наличие избыточности нельзя рассматривать как признак несовершенства источника сообщений. Наличие избыточности способствует повышению помехоустойчивости сообщений. Высокая избыточность естественных языков обеспечивает надежное общение между людьми.

Частотные характеристики текстовых сообщений

Важными характеристиками текста являются повторяемость букв, пар букв (биграмм) и вообще m-ок (m-грамм), сочетаемость букв друг с другом, чередование гласных и согласных и некоторые другие. Замечательно, что эти характеристики являются достаточно устойчивыми.

Идея состоит в подсчете чисел вхождений каждой n^m возможных m-грамм в достаточно длинных открытых текстах $T=t_1t_2…t_l$, составленных из букв алфавита $\{a_1, a_2, ..., a_n\}$. При этом просматриваются подряд идущие m-граммы текста

$t_1t_2...t_m$, $t_2t_3... t_{m+1}$, ..., $t_{i-m+1}t_{l-m+2}...t_l$.

Если $\vartheta(a_{i1}a_{i2}...a_{im})$ – число появлений m-граммы $a_{i1}a_{i2}...a_{im}$ в тексте T, а L общее число подсчитанных m-грамм, то опыт показывает, что при достаточно больших L частоты

$\frac{\vartheta(a_{i1}a_{i2}...a_{im})}{L}$ для данной m-граммы мало отличаются друг от друга.

В силу этого, относительную частоту считают приближением вероятности P ($a_{i1}a_{i2}...a_{im}$) появления данной m-граммы в случайно выбранном месте текста (такой подход принят при статистическом определении вероятности).

Для русского языка частоты (в порядке убывания) знаков алфавита, в котором отождествлены Е с Ё, Ь с Ъ, а также имеется знак пробела (-) между словами, приведены в таблице 1.

Таблица 7.1

- 0.175	О 0.090	Е, Ё 0.072	А 0.062
И 0.062	Т 0.053	Н 0.053	С 0.045
Р 0.040	В 0.038	Л 0.035	К 0.028
М 0.026	Д 0.025	П 0.023	У 0.021
Я 0.018	Ы 0.016	З 0.016	Ь, Ъ 0.014
Б 0.014	Г 0.013	Ч 0.012	Й 0.010
Х 0.009	Ж 0.007	Ю 0.006	Ш 0.006
Ц 0.004	Щ 0.003	Э 0.003	Ф 0.002

Некоторая разница значений частот в приводимых в различных источниках таблицах объясняется тем, что частоты существенно зависят не только от длины текста, но и от его характера.

Устойчивыми являются также частотные характеристики биграмм, триграмм и четырехграмм осмысленных текстов.

Порядок выполнения работы

Определить количество информации (по Хартли), содержащееся в заданном сообщении, при условии, что значениями являются буквы кириллицы.

«Фамилия Имя Отчество» завершил ежегодный съезд эрудированных школьников, мечтающих глубоко проникнуть в тайны физических явлений и химических реакций

Построить таблицу распределения частот символов, характерные для заданного сообщения. Производится так называемая частотная селекция, текст сообщения анализируется как поток символов и

высчитывается частота встречаемости каждого символа. Сравнить с имеющимися данными в таблице 7.1.

На основании полученных данных определить среднее и полное количество информации, содержащееся в заданном сообщении. Оценить избыточность сообщения.

Построить таблицу распределения частот символов, характерных для заданного сообщения путём деления количества определённого символа в данном сообщении на общее число символов

По формуле

$$\sum_{i=1}^{m} p_i \log p_i$$

H= вычислил энтропию сообщения

Далее по формуле Шеннона для определения кол-ва информации

$$I = -\log P = -k\sum_{i=1}^{m} p_i \log p_i$$ вычислил кол-во информации в передаваемом сообщении

Вычислил избыточность D по формуле

$$D = 1 - \frac{H_p}{H_o} = 1 - \frac{n_o}{n_p} = \frac{n_p - n_o}{n_p}$$

Отчет должен быть оформлен в текстовом редакторе и содержать:

- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;
- вывод о проделанной работе.

В нижний колонтитул поместить фамилию, инициалы и номер группы обучаемого (8 пт., Arial, выравнивание по правому краю).

Контрольные вопросы

1. Дать определение понятие энтропия?
2. Что означает вероятностный способ измерения информации?
3. Что означает статическое определение вероятности?
4. Запишите уравнение Хартли? Какие основные разработки внес в основу теории информации Шеннон?

Практическое занятие № 7

Сжатие информации.

Системные требования алгоритмов сжатия. Алгоритмы сжатия данных неизвестного формата.

Цель: научиться сжимать информацию с помощью метода Хаффмана и метода RLE.

Оборудование: ПК.

Программное обеспечение: операционная система, калькулятор, текстовый редактор.

Теоретические основы

Сжатие данных (англ. *data compression*) — алгоритмическое преобразование данных, производимое с целью уменьшения их объёма. Применяется для более рационального использования устройств хранения и передачи данных. Синонимы — упаковка данных, компрессия, сжимающее кодирование, кодирование источника. Обратная процедура называется восстановлением данных (распаковкой, декомпрессией). Сжатие основано на устранении избыточности, содержащейся в исходных данных. Простейшим примером избыточности является повторение в тексте фрагментов (например, слов естественного или машинного языка). Подобная избыточность обычно устраняется заменой повторяющейся последовательности ссылкой на уже закодированный фрагмент с указанием его длины. Другой вид избыточности связан с тем, что некоторые значения в сжимаемых данных встречаются чаще других. Сокращение объёма данных достигается за счёт замены часто встречающихся данных короткими кодовыми словами, а редких — длинными (энтропийное кодирование). Сжатие данных, не обладающих свойством избыточности (например, случайный сигнал или белый шум, зашифрованные сообщения), принципиально невозможно без потерь. В основе любого способа сжатия лежит модель источника данных, или, точнее, модель избыточности. Иными словами, для сжатия данных используются некоторые априорные сведения о том, какого рода данные сжимаются. Не обладая такими сведениями об источнике, невозможно сделать никаких предположений о преобразовании, которое позволило бы уменьшить объём сообщения. Модель избыточности может быть статической, неизменной для всего сжимаемого сообщения, либо строиться или параметризоваться на этапе сжатия (и

восстановления). Методы, позволяющие на основе входных данных изменять модель избыточности информации, называются адаптивными. Неадаптивными являются обычно узкоспециализированные алгоритмы, применяемые для работы с данными, обладающими хорошо определёнными и неизменными характеристиками. Подавляющая часть достаточно универсальных алгоритмов являются в той или иной мере адаптивными.

Все методы сжатия данных делятся на два основных класса:

Сжатие без потерь

Сжатие с потерями

При использовании сжатия без потерь возможно полное восстановление исходных данных, сжатие с потерями позволяет восстановить данные с искажениями, обычно несущественными с точки зрения дальнейшего использования восстановленных данных. Сжатие без потерь обычно используется для передачи и хранения текстовых данных, компьютерных программ, реже — для сокращения объёма аудио- и видеоданных, цифровых фотографий и т. п., в случаях, когда искажения недопустимы или нежелательны. Сжатие с потерями, обладающее значительно большей, чем сжатие без потерь, эффективностью, обычно применяется для сокращения объёма аудио- и видеоданных и цифровых фотографий в тех случаях, когда такое сокращение является приоритетным, а полное соответствие исходных и восстановленных данных не требуется.

Системные требования алгоритмов

Различные алгоритмы могут требовать различного количества ресурсов вычислительной системы, на которых они реализованы:

оперативной памяти (под промежуточные данные);

постоянной памяти (под код программы и константы);

процессорного времени.

В целом, эти требования зависят от сложности и «интеллектуальности» алгоритма. Общая тенденция такова: чем эффективнее и универсальнее алгоритм, тем большие требования к вычислительным ресурсам он предъявляет. Тем не менее, в специфических случаях простые и компактные алгоритмы могут работать не хуже сложных и универсальных. Системные требования определяют их потребительские качества: чем менее требователен алгоритм, тем на более простой, а следовательно, компактной, надёжной и дешёвой системе он может быть реализован.

Так как алгоритмы сжатия и восстановления работают в паре, имеет значение соотношение системных требований к ним. Нередко можно усложнив один алгоритм значительно упростить другой. Таким образом, возможны три варианта:

Алгоритм сжатия требует больших вычислительных ресурсов, нежели алгоритм восстановления.

Это наиболее распространённое соотношение, характерное для случаев, когда однократно сжатые данные будут использоваться многократно. В качестве примера можно привести цифровые аудио- и видеопроигрыватели.

Алгоритмы сжатия и восстановления требуют приблизительно равных вычислительных ресурсов.

Наиболее приемлемый вариант для линий связи, когда сжатие и восстановление происходит однократно на двух её концах (например, в цифровой телефонии).

Алгоритм сжатия существенно менее требователен, чем алгоритм восстановления.

Такая ситуация характерна для случаев, когда процедура сжатия реализуется простым, часто портативным устройством, для которого объём доступных ресурсов весьма критичен, например, космический аппарат или большая распределённая сеть датчиков. Это могут быть также данные, распаковка которых требуется в очень малом проценте случаев, например запись камер видеонаблюдения.

Алгоритмы сжатия данных неизвестного формата

Имеется два основных подхода к сжатию данных неизвестного формата.

На каждом шаге алгоритма сжатия очередной сжимаемый символ либо помещается в выходной буфер сжимающего кодера как есть (со специальным флагом, помечающим, что он не был сжат), либо группа из нескольких сжимаемых символов заменяется ссылкой на совпадающую с ней группу из уже закодированных символов. Поскольку восстановление сжатых таким образом данных выполняется очень быстро, такой подход часто используется для создания самораспаковывающихся программ.

Для каждой сжимаемой последовательности символов однократно либо в каждый момент времени собирается статистика её встречаемости в кодируемых данных. На основе этой статистики вычисляется вероятность значения очередного кодируемого символа (либо последовательности символов). После этого применяется та

или иная разновидность энтропийного кодирования, например, арифметическое кодирование или кодирование Хаффмана, для представления часто встречающихся последовательностей короткими кодовыми словами, а редко встречающихся — более длинными.

Код Хаффмана

Определение 1: Пусть $A=\{a_1,a_2,...,a_n\}$ - алфавит из n различных символов, $W=\{w_1,w_2,...,w_n\}$ - соответствующий ему набор положительных целых весов. Тогда набор бинарных кодов $C=\{c_1,c_2,...,c_n\}$, такой что:

(1) c_i не является префиксом для c_j, при i!=j

(2) $\sum_{i=1}^{n} w_i|c_i|$ минимальна ($|c_i|$ длина кода c_i)

называется *минимально-избыточным префиксным кодом* или иначе *кодом Хаффмана*.

Замечания:

Свойство (1) называется *свойством префиксности*. Оно позволяет однозначно декодировать коды переменной длины.

Сумму в свойстве (2) можно трактовать как размер закодированных данных в битах. На практике это очень удобно, т.к. позволяет оценить степень сжатия не прибегая непосредственно к кодированию.

В дальнейшем, чтобы избежать недоразумений, под кодом будем понимать битовую строку определенной длины, а под минимально-избыточным кодом или кодом Хаффмана - множество кодов (битовых строк), соответствующих определенным символам и обладающих определенными свойствами.

Известно, что любому бинарному префиксному коду соответствует определенное бинарное дерево.

Определение 2: Бинарное дерево, соответствующее коду Хаффмана, будем называть *деревом Хаффмана*.

Задача построения кода Хаффмана равносильна задаче построения соответствующего ему дерева. Приведем общую схему построения дерева Хаффмана:

Составим список кодируемых символов (при этом будем рассматривать каждый символ как одноэлементное бинарное дерево, вес которого равен весу символа).

Из списка выберем 2 узла с наименьшим весом.

Сформируем новый узел и присоединим к нему, в качестве дочерних, два узла выбранных из списка. При этом вес сформированного узла положим равным сумме весов дочерних узлов.

Добавим сформированный узел к списку.

Если в списке больше одного узла, то повторить 2-5.

Приведем пример: построим дерево Хаффмана для сообщения S="A H F B H C E H E H C E A H D C E E H H H C H H H D E G H G G E H C H H".

Для начала введем несколько обозначений:

Символы кодируемого алфавита будем выделять жирным шрифтом: **A**, **B**, **C**.

Веса узлов будем обозначать нижними индексами: A_5, B_3, C_7.

Составные узлы будем заключать в скобки: $((A_5+B_3)_8+C_7)_{15}$.

Итак, в нашем случае A={**A**, **B**, **C**, **D**, **E**, **F**, **G**, **H**}, W={2, 1, 5, 2, 7, 1, 3, 15}.

A_2 B_1 C_5 D_2 E_7 F_1 G_3 H_{15}

A_2 C_5 D_2 E_7 G_3 H_{15} $(F_1+B_1)_2$

C_5 E_7 G_3 H_{15} $(F_1+B_1)_2$ $(A_2+D_2)_4$

C_5 E_7 H_{15} $(A_2+D_2)_4$ $((F_1+B_1)_2+G_3)_5$

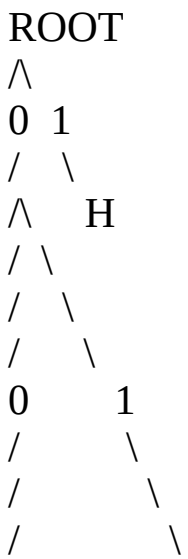
E_7 H_{15} $((F_1+B_1)_2+G_3)_5$ $(C_5+(A_2+D_2)_4)_9$

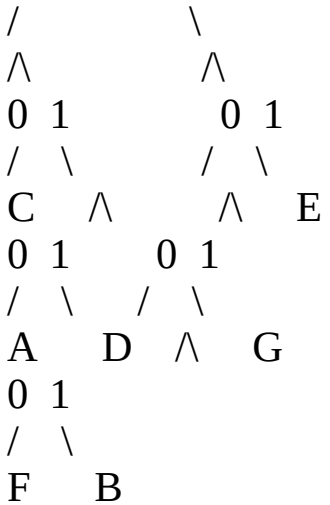
H_{15} $(C_5+(A_2+D_2)_4)_9$ $(((F_1+B_1)_2+G_3)\ _5+E_7)_{12}$

H_{15} $((C_5+(A_2+D_2)_4)\ _9+(((F_1+B_1)_2+G_3)\ _5+E_7)_{12})_{21}$

$(((C_5+(A_2+D_2)_4)\ _9+(((F_1+B_1)_2+G_3)\ _5+E_7)_{12})_{21}+H_{15})_{36}$

В списке, как и требовалось, остался всего один узел. Дерево Хаффмана построено. Теперь запишем его в более привычном для нас виде.





Листовые узлы дерева Хаффмана соответствуют символам кодируемого алфавита. Глубина листовых узлов равна длине кода соответствующих символов.

Путь от корня дерева к листовому узлу можно представить в виде битовой строки, в которой "0" соответствует выбору левого поддерева, а "1" - правого. Используя этот механизм, мы без труда можем присвоить коды всем символам кодируемого алфавита. Выпишем, к примеру, коды для всех символов в нашем примере:

$A=0010_{bin}$	$C=000_{bin}$	$E=011_{bin}$	$G=0101_{bin}$
$B=01001_{bin}$	$D=0011_{bin}$	$F=01000_{bin}$	$H=1_{bin}$

Теперь у нас есть все необходимое для того чтобы закодировать сообщение S. Достаточно просто заменить каждый символ соответствующим ему кодом:

$S^{/}$="0010 1 01000 01001 1 000 011 1 011 1 000 011 0010 1 0011 000 011 011 1 1 1 000 1 1 1 0011 011 0101 1 0101 0101 011 1 000 1 1".

Оценим теперь степень сжатия. В исходном сообщении S было 36 символов, на каждый из которых отводилось по $[\log_2|A|]=3$ бита (здесь и далее будем понимать квадратные скобки [] как целую часть, округленную в положительную сторону, т.е. [3,018]=4). Таким образом, размер S равен 36*3=108 бит

Размер закодированного сообщения $S^{/}$ можно получить воспользовавшись замечанием 2 к определению 1, или непосредственно, подсчитав количество бит в $S^{/}$. И в том и другом случае мы получим 89 бит.

Итак, нам удалось сжать 108 в 89 бит.

Теперь декодируем сообщение $S^{/}$. Начиная с корня дерева будем двигаться вниз, выбирая левое поддерево, если очередной бит в

потоке равен "0", и правое - если "1". Дойдя до листового узла мы декодируем соответствующий ему символ.

Ясно, что следуя этому алгоритму мы в точности получим исходное сообщение S.

Метод RLE.

Наиболее известный простой подход и алгоритм сжатия информации обратимым путем - это кодирование серий последовательностей (Run Length Encoding - RLE). Суть методов данного подхода состоит в замене цепочек или серий повторяющихся байтов или их последовательностей на один кодирующий байт и счетчик числа их повторений. Проблема всех аналогичных методов заключается лишь в определении способа, при помощи которого распаковывающий алгоритм мог бы отличить в результирующем потоке байтов кодированную серию от других - некодированных последовательностей байтов. Решение проблемы достигается обычно простановкой меток в начале кодированных цепочек. Такими метками могут быть, например, характерные значения битов в первом байте кодированной серии, значения первого байта кодированной серии и т.п. Данные методы, как правило, достаточно эффективны для сжатия растровых графических изображений (BMP, PCX, TIF, GIF), т.к. последние содержат достаточно много длинных серий повторяющихся последовательностей байтов. Недостатком метода RLE является достаточно низкая степень сжатия или стоимость кодирования файлов с малым числом серий и, что еще хуже - с малым числом повторяющихся байтов в сериях.

Практическое задание

1. Сжатие методом Хаффмана

«КАКАЯ ЗИМА ЗОЛОТАЯ!
КАК БУДТО ИЗ ДЕТСКИХ ВРЕМЕН...
НЕ НАДО НИ СОЛНЦА, НИ МАЯ –
ПУСТЬ ДЛИТСЯ ТОРЖЕСТВЕНИЫЙ СОН.

ПУСТЬ Я В ЭТОМ СНЕ ПОЗАБУДУ
КОГДА-ТО МАНИВШИЙ ОГОНЬ,
И ЛЕТО ПРЕДАМ, КАК ИУДА,
ЗА ТРИДЦАТЬ СНЕЖИНОК В ЛАДОНЬ.

ЗАТЕМ, ЧТО И Я ХОЛОДЕЮ,
ТЕПЛО УЖЕ СТРАШНО ПРИНЯТЬ:

Я СЛИШКОМ ДАВНО НЕ УМЕЮ
НИ ТЛЕТЬ, НИ ГОРЕТЬ, НИ СЖИГАТЬ…

ВСЕ ЧАЩЕ, ВСЕ ДОЛЬШЕ НЕМЕЮ:
К ЗИМЕ УЖЕ ДЕЛО, К ЗИМЕ...
И ТОЛЬКО ТОГО ОТОГРЕЮ,
КОМУ ХОЛОДНЕЕ, ЧЕМ МНЕ»

2. С помощью сжатия по методу RLE.

1 последовательность:

SSSSOOOEEERROOOAAYYYYYDDDDOEUUUUUWWWWJJJORRUUUUUUUUUUXXXKHHHHHHMMMMMMMGGGLLLLLLLJJJJ

2 последовательность:

FFFFFFFFKKKKKSSSSUURERRRRRRRRRPPPPPPPPDDDDKKKKKKGLDDDDDDDDDKKKKKKKKGGGGMGMMMM

Отчет

- Отчет должен содержать:
- наименование работы;
- цель работы;
- задание;
- последовательность выполнения работы;
- ответы на контрольные вопросы;
- вывод о проделанной работе.

В нижний колонтитул поместить фамилию, инициалы и номер группы обучаемого (8 пт., Arial, выравнивание по правому краю).

Контрольные вопросы

1. Запишите какие Вы знаете методы сжатия информации?
2. Перечислите основные программы применяемые для сжатия информации?

Практическое занятие № 8

Сжатие информации. Практическое применение различных алгоритмов сжатия

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

Оборудование: ПК.

Программное обеспечение: операционная система, программы архиваторы: RAR, ARJ и ZIP

Теоретические основы

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: BBBBBLLLLLAAAAA

В шестнадцатиричной системе :

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

Архиватор может представить этот файл в следующем шестнадцатиричном

виде : 01 05 42 06 05 4 C OA 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 > [- <спецификация 1> [- <спецификация 2>]…] < 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 файла 1> < имя файла 2>

Для извлечения файлов из архива: arj e < имя архива> <имя файла 1> < имя

файла 2>

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 файла 1> < имя файла 2>

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

файла 1> < имя файла 2>

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.

Практическое 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.

Содержание отчета

Отчет должен содержать следующие разделы:

- Ответы на контрольные вопросы.
- Результаты сжатия файлов в виде таблицы.
- Выводы о проделанной работе.
- В нижний колонтитул поместить фамилию, инициалы и номер группы обучаемого (8 пт., Arial, выравнивание по правому краю).

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.

Практическая работа №9

Сжатие информации. Сравнение и анализ архиваторов

Цель: формирование практических навыков и умений архивирования и сжатия файлов.

Оборудование: ПК.

Программное обеспечение: операционная система, программы архиваторы.

Теоретические основы

Характерной особенностью большинства типов данных, с которыми традиционно работают пользователи, является определенная избыточность. Степень избыточности зависит от типа данных.

При обработке информации избыточность также играет важную роль. Так, например, при преобразовании или селекции информации избыточность используют для повышения ее качества (репрезентативности, актуальности, адекватности и т.п.). Однако, когда речь заходит не об обработке, а о хранении готовых документов или их передаче, то избыточность можно уменьшить, что дает эффект сжатия данных.

Если методы сжатия информации применяют к готовым документам, то нередко термин *сжатие данных* подменяют термином *архивация данных,* а программные средства, выполняющие эти операции, называют *архиваторами.*

Степень сжатия файлов характеризуется коэффициентом K_c, определяемым как отношение объема сжатого файла V_c к объему исходного файла V, выраженное в процентах: $K_o=(V_c / V*100)$. Степень сжатия зависит от используемой программы, метода сжатия и типа исходного файла. Наиболее хорошо сжимаются файлы графических образов, текстовые файлы и файлы данных, для которых степень сжатия может достигать 5-40%, меньше сжимаются файлы исполняемых программ и загрузочных модулей – 60-90%. Почти не сжимаются архивные файлы.

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, например на гибком диске.

Уплотнение папок используют как средство архивации данных перед длительным хранением, в частности, при резервном копировании.

Уплотнение дисков служит целям повышения эффективности использования их рабочего пространства и, как правило, применяется к дискам, имеющим недостаточную емкость.

Несмотря на изобилие алгоритмов сжатия данных, теоретически есть только три способа уменьшения их избыточности. Это либо изменение содержания данных, либо 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 также *методами сжатия с регулируемой потерей информации.* Они применимы только для тех типов данных, для которых формальная утрата части содержания не приводит к значительному снижению потребительских свойств. В первую очередь, это относится к мультимедийным данным: видеорядам, музыкальным записям, звукозаписям и рисункам. Методы сжатия с потерей информации обычно обеспечивают гораздо более высокую степень сжатия, чем обратимые методы, но их нельзя применять к текстовым документам, базам данных и, тем более, 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.

Архиваторы

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

К базовым функциям, которые выполняют современные диспетчеры архивов, относятся: 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.

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;

«прозрачное» уплотнение исполнимых файлов .EXE и .DLL;

создание самораспаковывающихся многотомных архивов;

выбор или настройку коэффициента сжатия информации.

Структура окон WinRAR и WinZip типична для приложений Windows. Вид панели инструментов WinRAR приведен на рис. 1.

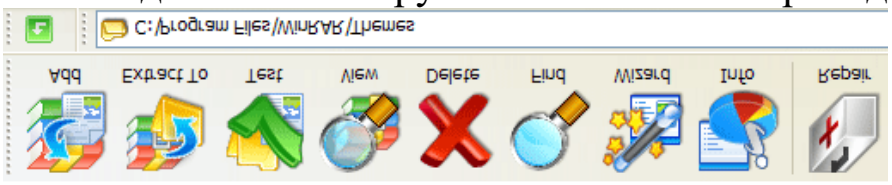


Рисунок 10.1. Панель инструментов WinRAR

Самораспаковывающиеся архивы

В тех случаях, когда архивация производится для передачи документа потребителю, следует предусмотреть наличие у него программного средства, необходимого для извлечения исходных данных из уплотненного архива. Если таких средств у потребителя нет – создают самораспаковывающиеся архивы. Самораспаковывающийся архив готовится на базе обычного архива путем присоединения к нему небольшого программного модуля. Сам архив получает расширение .EXE, характерное для исполняемых файлов (рис. 2). 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. Некоторые диспетчеры (например, WinZip) выполняют разбиение сразу на гибкие диски, а некоторые (например, WinRAR) позволяют выполнить предварительное разбиение архива на фрагменты заданного размера на жестком диске. Впоследствии их можно перенести на внешние носители путем копирования.

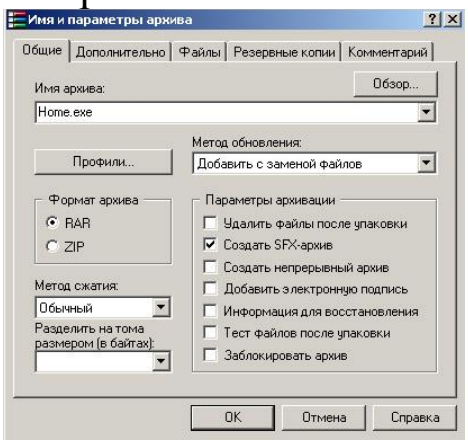


Рисунок 10.2. Определение параметров архива

При создании распределенных архивов диспетчер WinZip обладает особенностью: каждый том несет файлы с одинаковыми именами. В результате этого нет возможности установить номера томов, хранящихся на каждом из гибких дисков, по названию файла. Поэтому каждый диск следует маркировать пометками на наклейке, а при создании распределенного архива следует быть внимательнее, чтобы не перепутать последовательность немаркированных томов.

В случае необходимости узнать номер тома можно не по названию файла, а по метке на диске, хотя эта операция не слишком удобна. Для этого следует открыть окно «Мой компьютер», выбрать значок дисковода, щелкнуть на нем правой кнопкой мыши и выбрать в контекстном меню пункт «Свойства». В диалоговом окне «Свойства: Диск …» на вкладке «Общие» можно узнать номер тома распределенного архива в поле «Метка тома».

Консольная версия WinRAR

Консольная версия WinRAR поддерживает архивы только в формате RAR, у которых обычно расширение ".rar". ZIP и прочие форматы не поддерживаются. Пользователи Windows могут установить GUI-версию RAR – WinRAR, которая обрабатывает и архивы других типов.

Некоторые отличительные особенности RAR:

оригинальный высокоэффективный алгоритм сжатия данных;

специальные алгоритмы сжатия, оптимизированные для текстовых, аудио- и графических данных, а также для 32- и 64-битовых исполняемых файлов архитектуры Intel;

лучшая, чем у аналогичных продуктов, степень сжатия при использовании режима "непрерывного" (solid) архивирования;

электронная подпись (только в зарегистрированной версии);

самораспаковывающиеся (SFX) архивы и тома;

восстановление физически поврежденных архивов;

блокировка, шифрование, задание порядка архивирования файлов;

сохранение прав доступа к файлам, меток тома и др.

Следует отметить, что при создании томов RAR в FAT или FAT32 WinRAR автоматически ограничивает максимальный объем тома до 4 ГБ минус 1 байт, так как эти файловые системы не поддерживают файлы объемом больше 4 ГБ.

Работа с WinRAR из консоли

Синтаксис командной строки WinRAR

Формат вызова:

WinRAR <команда> [-<ключи>...] <архив> [<@файлы-списки...>] [<файлы...>] [<путь_для_извлечения\>]

Для создания и управления архивами служат параметры командной строки (команды и ключи). *Команда* – это строка (или одна буква), указывающая, что WinRAR должен выполнить соответству-

ющее действие. *Ключи* модифицируют действие команды. Остальные параметры – это имена архива и файлов, которые будут добавлены или извлечены из архива.

Файлы-списки – это обычные текстовые файлы, содержащие имена файлов для обработки. Каждое имя файла должно быть указано на отдельной строке и начинаться с первой позиции строки. В файл-список допускается помещать комментарии, признак начала комментария – символы //. Например, для архивирования файлов *.txt из каталога c:\1kurs\doc, файлов *.bmp из каталога c:\1kurs\image и всех файлов из каталога c:\evm\misc можно создать backup.lst, содержащий следующие строки:

c:\1kurs\doc.txt //резервная копия текстов*

c:\1kurs\image.bmp //резервная копия рисунков*

c:\1kurs\misc

После этого для архивирования достаточно будет выполнить команду:

winrar a backup @backup.lst

Если требуется прочитать имена файлов с устройства stdin (стандартный ввод), то после символа "@" не указывайте имя файла (просто @).

В одной командной строке разрешается указывать как обычные имена или группы файлов для обработки, так и файлы-списки. Если не указаны ни файлы, ни файлы-списки, то подразумевается шаблон *.* (т.е. WinRAR обработает все файлы).

Команды:

a – добавляет указанные файлы к архиву;

m – переносит указанные файлы и подкаталоги в архив;

d – удаляет указанные файлы из архива;

x – извлекает указанные файлы из архива с восстановлением структуры подкаталогов;

e – извлекает указанные файлы из архива в текущий подкаталог;

v – просмотр содержимого архива;

u – добавляет те файлы к архиву, которых в нем нет;

c – добавляет комментарии к архиву;

k – защита данных от модификации.

Ключи:

-? – выводит экран помощи;

-r – сохраняет структуру подкаталогов;

-o+ – при распаковке разрешает перезаписывать существующие файлы;

-o- – при распаковке не разрешает перезаписывать существующие файлы;

-x<name> – все файлы, с соответствующими name именами, будут исключены из обработки (можно использовать шаблоны);

-x@<list> – задает файл, в котором содержатся имена файлов, исключаемых из обработки;

-v<size> – создание архивных томов;

-p<password> – назначить пароль.

Примеры команд

1). Добавить комментарий к архиву:

rar c distrib.rar

Комментарии отображаются во время обработки архива. Длина комментария не должна превышать 62000 байт.

2). Добавить комментарий из файла: *rar c -zinfo.txt dummy*

3). Записать комментарий архива в указанный файл:

rar cw oldarch comment.txt

4). Выполнить регистрозависимый поиск строки "first level" в файлах *.txt, находящихся в архивах *.rar на диске c:.

rar "ic=first level" -r c:.rar *.txt*

Поддерживаются следующие необязательные параметры:

i – не различать прописные и строчные буквы (по умолчанию);

c – различать прописные и строчные буквы;

h – поиск в шестнадцатеричном режиме;

t – использовать таблицы символов ANSI, Unicode и др.

Если ни один параметр не указан, вместо синтаксиса i=<строка> можно использовать более простую команду i<строка>. Модификатор 't' допускается применять вместе с другими параметрами.

5). Найти шестнадцатеричную строку f0e0aeaeab2d83e3a9 в архивах RAR, расположенных в каталоге e:\texts

rar ih=f0e0aeaeab2d83e3a9 -r e:\texts

6). Добавить к пути назначения имя архива

*rar x -ad *.rar data*

Эта опция может пригодиться при распаковке группы архивов. По умолчанию RAR извлекает файлы из всех архивов в одну и ту же папку, если же указать этот ключ, то файлы из каждого архива

будут распакованы в отдельные папки (в данном случае в папке 'data').

Работа с архиватором WinRAR

Получение справки о программе

Для получения справочной информации выберите команду ? Содержание. В окне *Справка* выберите на вкладке *Содержание* раздел *WinRAR Interface*, подраздел *WinRAR menus*, как показано на рис. 3.

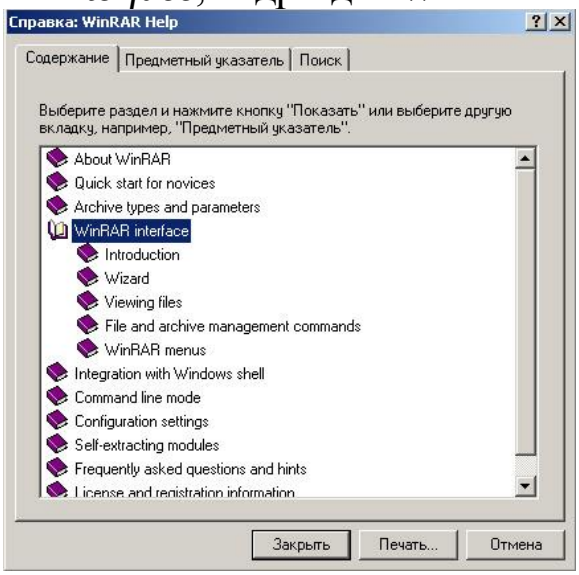


Рисунок 10.3. Окно справки WinRAR

После запуска архиватора WinRAR на экране будет раскрыто окно, приведенное на рис. 4.

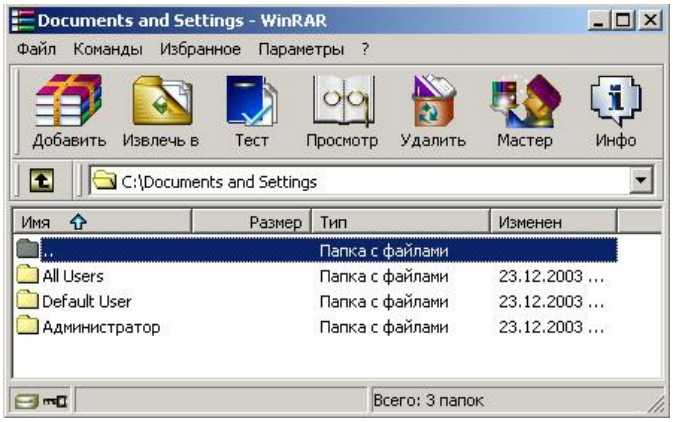


Рисунок 10.4. Окно *WinRAR* в режиме операций с файлами

Окно архиватора WinRAR, в отличие от окна, WinZip, имеет средства навигации по дискам и папкам компьютера: поле списка для выбора дисков и папок, кнопку для перехода на верхний уровень в иерархии папок.

Для выбора нужного диска используйте окно списка дисков. Для выхода в родительский каталог щелкните ярлык папки с именем «..». Для открытия нужной папки щелкните ярлык с названием папки.

При проведении процессов архивации (разархивации) с группой файлов, имена которых задаются шаблонами, применяются следующие действия. Для выделения группы файлов выберите в меню *File* команду *Select group* или щелкните *кнопку Серый плюс* и задайте в окне выбора маску «*0*.**», как показано на рис. 5. Щелкнув кнопку «ОК», завершите создание маски для выбора группы файлов.

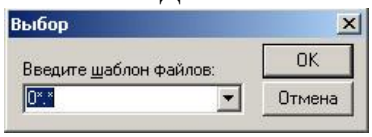


Рисунок 10.5. Выделение группы файлов в архиве

Для создания архива из нескольких файлов, выделите нужные файлы и щелкните кнопку «Добавить» (Add) на панели инструментов (рис. 6).

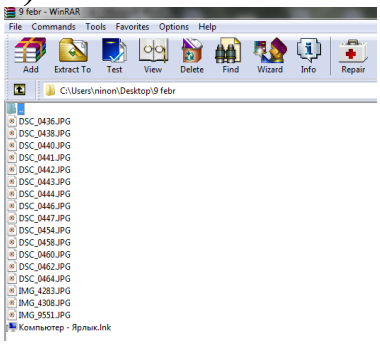


Рисунок 10.6. Добавление выбранных файлов в архив

Для удаления из архива файла необходимо открыть архив в окне архиватора WinRAR, указать удаляемый файл и щелкнуть кнопку «Удалить» на панели инструментов или выбрать последовательность команд: *Команды-Удалить файлы.* Подтвердить удаление можно, нажав кнопку «Да» в окне подтверждения *Удаление* (рис. 7).

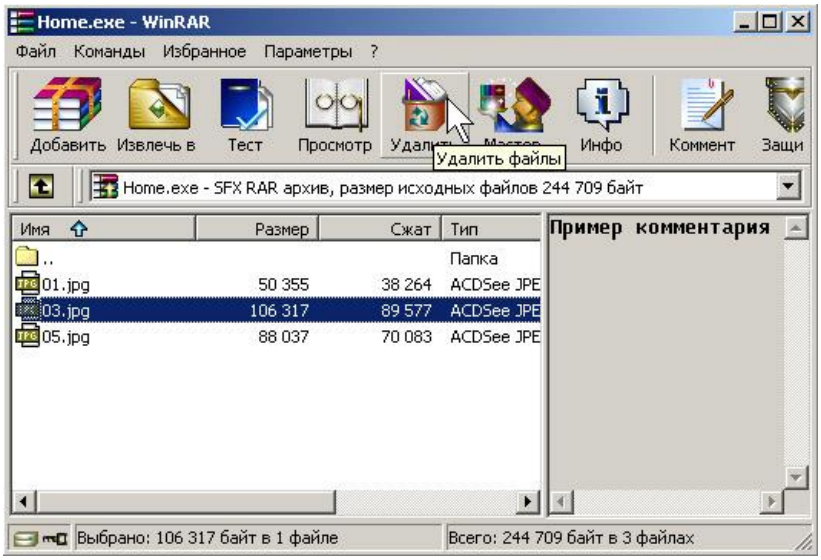


Рисунок 10.7. Окно WinRAR в режиме «удаление файла из архива»

Изменение настроек программы WinRAR

Для изменения настроек выберите команду *Параметры-Установки,* после чего на экране развернется окно настройки параметров WinRAR. Выбирая различные вкладки окна *Параметры* для получения подсказки по параметрам настройки, используйте всплывающую подсказку. Задайте следующие параметры настройки WinRAR:

- на вкладке *Архивация* щелкните кнопку «Создать по умолчанию» для создания опций архивирования по умолчанию, в открывшемся после этого окне *Установить параметры* сжатия по умолчанию включите опции Создать SFX-архив, в списке Размер тома выберите стандартный размер тома сменного носителя. Щелкнув кнопку «ОК», закройте окно *Установить параметры сжатия* по умолчанию (рис. 8). Можно отредактировать значение размера тома в списке Размер тома, задав его величину вручную;

- на вкладке Интеграция включите все флажки в поле *Связать WinRAR* с и щелкните кнопку «ОК» для применения внесенных изменений. Проверьте действие измененных параметров, выделив несколько файлов и щелкнув кнопку *«Добавить»* на панели инструментов. После этого откроется окно *Имя и параметры архива,* в поле *Имя архива* которого выводится имя с расширением .exe (как было установлено, по умолчанию создается SFX-архив), в поле *Размер тома* отображается значение заданного по умолчанию размер тома. Щелкнув клавишу *Esc*, отмените архивацию.

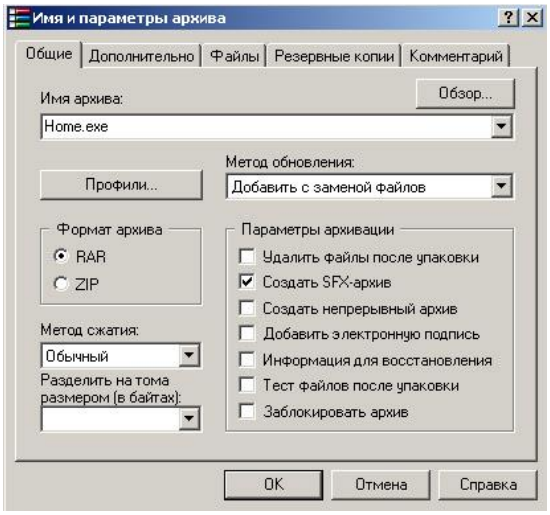


Рисунок 10.8. Определение параметров архива

Создание многотомных архивов

Для создания многотомного архива файлов необходимо открыть окно архиватора, выбрать в поле списка дисков и папок папку, подлежащую архивации, выделите все файлы и щелкните кнопку *«Добавить»* на панели инструментов.

В окне *Имя и параметры архива* выбрать вкладку *Общие.* Далее в поле *Имя архива* задайте имя архива (например, Archive2.rar), выберите вариант формата архива RAR, в поле *Volume size* (Размер тома) задайте размер тома архива (например, 1.44).

Внимание: при выполнении лабораторной работы размер тома определите в несколько раз меньше суммарного объема файлов, включаемых в архив, чтобы в процессе архивации было создано нескольких томов.

Щелкнув кнопку «ОК», запустите операцию упаковки файлов в архив. По окончании архивации в текущем каталоге появится несколько файлов с именем созданного архива, с расширениями, отличающимися нумерацией, например: Archive2.rar, Archive2.r00, Archive2.r01, Archive2.r02 и т.п., где файл с расширением .rar – первый том архива, файлы с расширением .r00, .r01, .r02 и т.п. – файлы следующих томов архива.

Создание защищенных архивов

Для создания архивов, доступ к которым защищен паролем, выберите в меню *Файл* команду *Пароль*, в окне *Ввод пароля* по умолчанию в поле *Введите пароль* введите значение пароля и повторите ввод пароля в поле *Повторите пароль для проверки.* Щелкнув кнопку «ОК», завершите определение пароля. После этого в данном сеансе работы

архиватора доступ ко всем создающимся архивам будет закрываться заданным паролем (рис. 9).

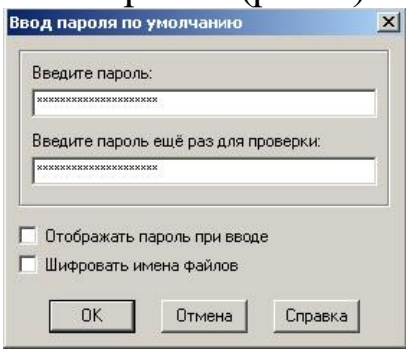


Рисунок 10.9. Задание пароля архива

Внимание: при вводе пароля обратите внимание на включенный регистр символов.

Создайте архив из нескольких файлов в рабочем каталоге.

При извлечении файлов из защищенного паролем архива откроется окно *Ввод пароля.* Введите в поле *Введите пароль для зашифрованного файла* любое сочетание символов – неправильный пароль и щелкните кнопку «ОК». Если пароль неправильный, то раскроется окно сообщений, в котором будет выведено сообщение: *Ошибка CRC* в зашифрованном файле (неправильный пароль). Щелкнув кнопку «Закрыть», закройте окно сообщения. Повторно щелкнув кнопку «Извлечь в» на панели инструментов, в окне *Ввод пароля* введите правильный пароль и щелкните кнопку «ОК». Если пароль был введен правильно, то файл будет распакован из архива.

Создание самораспаковывающегося ZIP-архива

1). Запустите программу WinZip.

2). Выполните команду File/Open Archive (Файл /Открыть архив). Откройте ранее созданный архив .zip.

3). Выполните команду Actions /Make .Exe File (Действия/Создать исполнимый файл) – откроется диалоговое окно WinZip Self-Extractor (Генератор самораспаковывающегося архива).

4). В поле Create Self-Extracting Zip files from (Создать самораспаковывающийся архив из ...) необходимо записать адрес исходного ZIP-файла. Можно воспользоваться кнопкой Browse (Обзор) для поиска нужного файла.

5). В группе Self Extractor Type (Тип самораспаковывающегося архива) включите переключатель, соответствующий операционной системе компьютера, для которого готовится архив.

6). В группе Spanning Support (Поддержка распределенного архива) включите переключатель No spanning (Без распределения) и нажмите кнопку ОК.

Создание самораспаковывающегося распределенного архива

1). Запустите программу WinZip.

2). Выполните команду File/Open Archive (Файл /Открыть архив). Откройте ранее созданный архив .zip.

3). Выполните команду Actions /Make .Exe File (Действия /Создать исполнимый файл) – откроется диалоговое окно WinZip Self-Extractor (Генератор самораспаковывающегося архива).

4). В группе элементов управления Spanning Support (Поддержка распределенного архива) включите переключатель Safe Spanning Method (Защищенный метод распределения) или Old Spanning Method (Обычный метод распределения).

Защищенный метод создает на первом гибком диске два файла: исполнимый файл, выполняющий автоматическую распаковку, и первый том распределенного архива. На последующих дисках создается продолжение распределенного архива. Такой подход повышает уровень безопасности, поскольку даже в том случае, когда исполнимый файл поврежден, например компьютерным вирусом, информация остается в архивном файле. Этот метод применяют для передачи архивных материалов на гибких дисках.

Обычный метод не создает отдельного исполнимого файла и весь архив хранится в одном исполнимом файле, распределенном по нескольким носителям. Данный метод используют для самораспаковывающихся архивов, передаваемых по каналам компьютерных сетей.

5). Откройте диалоговое окно WinZip Self-Extractor (Генератор самораспаковывающегося архива) и установите флажок Erase any existing files on the new disk before continuing (Предварительно стереть все существующие файлы на гибких дисках).

6). Далее нажмите кнопку ОК – начнется процесс создания первого тома распределенного архива. По окончании процесса по указанию программы извлеките записанный гибкий диск и вставьте новый.

7). Создав последний том, программа предложит извлечь последний диск и вставить первый для внесения правок в заголовок архива.

Альтернативные архиваторы

Среди альтернативных архиваторов можно выделить 5 программ: *Universal Extractor* – программа, служащая для извлечения данных из архивов практически любых типов; *7-Zip* – бесплатный файловый архиватор для Windows с высокой степенью сжатия; *PeaZip* – свободный бесплатный архиватор и графическая оболочка для других архиваторов; *IZArc* – бесплатный архиватор для Windows, поддерживающий большое количество форматов; *TUGZip* – простой в использовании архиватор, поддерживающий большое количество форматов. Среди перечисленных архиваторов лидирующие позиции занимает 7-Zip. По степени сжатия он является лучшим не только среди бесплатных программ, но и подавляющего большинства коммерческих продуктов. 7-Zip работает со всеми популярными форматами архивов, поддерживает шифрование, умеет создавать самораспаковывающиеся архивы и обладает многими другими удобными функциями. К недостаткам 7-Zip можно отнести сравнительно малое количество поддерживаемых форматов. Программа IZArc умеет распаковывать около 50 типов архивов, включая многие редкие. Также он может архивировать и сохранять файлы в 12 различных форматах и обрабатывать многотомные ZIP-архивы. Мультиформатный архиватор TUGZip имеет некоторые специальные возможности, например, восстановление поврежденных архивов ZIP и SQX. PeaZip – небольшой, бесплатный архиватор с открытыми кодами, как и IZArc поддерживает множество форматов архивов, включая ACE, ARJ, CAB, DMG, ISO, LHA, RAR, и UDF, работает как с 32, так и с 64-битнымы версиями Windows. Universal Extractor нельзя назвать настоящим архиватором, так как сжимать файлы он не умеет, но является наилучшим распаковщиком редких форматов. Огромное количество поддерживаемых форматов делает его лучшим в этом секторе

Интеграция служебных и прикладных программ с ОС

Под интеграцией программного обеспечения понимают возможность совместной работы нескольких различных программ в рамках единой системы управления. Так, например, известным системным средством интеграции является концепция внедрения и связывания объектов и основанный на ней буфер обмена Windows. Другим приемом интеграции, в основе которого лежит изменение свойств программы Проводник и связанного с ней контекстного меню объектов. Для эпизодических работ по архивации и извлече-

нию файлов и папок удобнее использовать систему, хорошо интегрированную в Windows, например, WinZip. Для регулярных работ по созданию резервных копий папок и дисков удобнее использовать автономные средства, поскольку для них проще организуется взаимодействие с прочими программами (в частности, со средствами автоматизации). В этих случаях можно рекомендовать, например, программу WinRAR.

1). Запустите программу «Проводник» (Пуск / Программы / Проводник).

2). Скопируйте в созданную папку несколько произвольных файлов.

3). Выделите один из файлов и откройте контекстное меню. Обратите внимание на то, что в нем имеются два пункта для создания архива (создание архива с произвольным именем и с именем, соответствующим текущему файлу). Появление этих пунктов связано с наличием в компьютерной системе диспетчера архивов и интеграции WinZip с Проводником Windows.

4). Выполните команду Add to Zip (Добавить в архив). Далее произойдет автоматический запуск диспетчера архивов WinZip и откроется диалоговое окно Add (Добавление в архив).

5). В поле Add to archive (Добавить в архив) ввести название файла создаваемого архива, адрес текущей папки заносится автоматически. Проверив настройку прочих элементов управления, запустите процесс архивации щелчком на командной кнопке Add (Добавить).

6). Перейдите в окно программы Проводник и убедитесь в том, что в папке появился архивный файл test.zip. Щелкните на значке архивного файла правой кнопкой мыши и изучите новые команды контекстного меню, позволяющие выполнить операции с архивным файлом.

7). Выполните команду Create Self-Extractor (Создать самораспаковывающийся архив). В открывшемся диалоговом окне щелкните на командной кнопке «Да» и в последующих диалоговых окнах откажитесь от проверки созданного архива.

8). Закройте открытые окна программы WinZip и в программе Проводник убедитесь в том, что в рабочей папке появился исполняемый файл (.exe).

9). В программе Проводник выполните перетаскивание значка любого файла (или группы файлов) на значок созданного ZIP-архива. При отпускании кнопки мыши в конце перетаскивания происходит

автоматическое добавление новых файлов в архив. Если содержимое правой панели Проводника открыто в режиме Таблица, после каждого перетаскивания можно наблюдать увеличение размера файла архива.

Исследование свойств форматов сжатия графических данных

1). Откройте графический редактор Paint (Пуск/Программы/Стандартные/ Paint).

2). Загрузите в него заранее подготовленный многоцветный рисунок.

3). Определите размер рисунка в пикселях (Рисунок/Атрибуты).

4). Оцените теоретический размер рисунка в 24-разрядной палитре (3 байта на точку) по формуле:

$S=M-N-3$, где S – размер файла с рисунком (байт);

M – ширина рисунка (точек);

N – высота рисунка (точек).

5). Сохраните рисунок в папку C:\Temp\Pictures, выбрав имя файла test и назначив тип файла: 24-разрядный рисунок (.BMP).

6). Повторно сохраните рисунок, выбрав то же имя test, но назначив тип файла .GIF. При сохранении произойдет потеря определенной части графической информации.

7). Восстановите рисунок, загрузив его из ранее сохраненного файла Test.bmp.

8). Вновь сохраните его под тем же именем, но выбрав в качестве типа файла формата *.JPEG*.

9). Запустите программу Проводник.

10). Откройте папку C:\Temp\Pictures в режиме Таблица.

11). Определите размеры файлов Test.bmp, Test.gif и Test.jpg.

12). Определите коэффициент сжатия файлов *(R)*, взяв отношения размеров файлов к теоретической величине, полученной расчетным путем.

Порядок выполнения работы

1) Создать или скопировать на рабочем диске в рабочей директории 5-7 файлов (текстовых, исполняемых, командных, программных).

2) Создать архивы для этих файлов с помощью различных архиваторов, например, WinRar, WinZip и др.

3) Сравнить объемы получившихся файлов, результаты занести в таблицу и сделать выводы:

Таблица10.1.

Название архиватора	Тип файла	Размер файла	Размер файла после сжатия	Степень сжатия(%)

4) С помощью архиватора (в соответствии с заданием преподавателя) выполнить следующие команды:

а) добавить в архив заданный файл;

б) поместить в архив все файлы из текущего каталога, за исключением файлов с заданным расширением;

в) создать защищенный архив;

г) создать архивный файл, позволяющий сохранить структуру каталогов;

д) добавить комментарии к архивам;

е) извлечь заданный файл из архива.

ж) создать многотомный архив, указав размер тома – 80 К;

з) выполнить поиск заданной строки в архивах по различным поисковым признакам.

5) Используя программу архивации, создать на диске, заданном в параметрах, многотомный архив с паролем, заданным в параметрах, поместив в них все файлы из каталога LAB рабочего диска, исключив файлы с расширением EXE.

6) Просмотреть списки созданных архивов.

7) Создать командный файл, который с помощью архиватора позволяет расположить файлы в архиве в заданном порядке, просмотреть архив, извлечь файлы из архива в заранее созданный каталог.

8) Создать самораспаковывающиеся RAR- и ZIP-архивы, не поддерживающие распределенные архивы (включить переключатель «Без распределения» в группе Spanning Support – Поддержка распределенного архива).

9) Создать самораспаковывающиеся распределенные архивы RAR- и ZIP-архивы.

10) Используя диспетчер архивов WinZip, выполнить интеграцию служебных и прикладных программ с операционной системой Windows.

11) Исследуйте свойства форматов сжатия графических данных (файлы .bmp, .gif, .jpg). Результаты занесите в таблицу:

Таблица10.2.

Формат файла	Размер файла (Кбайт)	Степень сжатия (%)
24 разрядный .bmp		
.gif		
.jpg		

12) Используя программу, например, Excel, построить диаграммы по результатам, приведенным в таблицах, и сделать выводы.

Содержание отчета

Отчет должен содержать следующие разделы:

Ответы на контрольные вопросы.

- Результаты сжатия файлов в виде таблицы.
- Выводы о проделанной работе.
- В нижний колонтитул поместить фамилию, инициалы и номер группы обучаемого (8 пт., Arial, выравнивание по правому краю).

Контрольные вопросы

1. Для чего необходимо создавать архив?
2. Поясните основные алгоритмы архивации.
3. Как можно упаковать информацию при хранении на диске?
4. Приведите команды упаковки данных в архив и распаковки данных из архива для архиватора Winrar.exe в консольном режиме.
5. Как создать защищенный архив?
6. Приведите команды упаковки данных в архив Winzip.exe и распаковки данных из архива.
7. Как создать многотомный архив?
8. Укажите расширение имен файлов продолжения архива.
9. Как получить полную справку по всем возможным режимам работы программы-архиватора?

10. Как создать самораспаковывающий архив?
11. Приведите примеры альтернативных программ архивации.
12. В чем особенность альтернативных программ архивации.
13. Что понимается под интеграцией служебных и прикладных программ с ОС?

Сжатие информации. Итоговое практическое задание

Цель: Закрепление теоретических знаний о методе сжатия сообщений с использованием динамических словарей (алгоритм LZ)

Оборудование: ПК.

Программное обеспечение: операционная система, программы архиваторы.

Теоретические основы

Сообщения, включающие текст, числовые данные код программ и т.п. не допускают потери информации. Наиболее распространенным средством их сжатия является метод динамических словарей. В частности, в современных программах архиваторах широко используется алгоритм Лемпеля-Зива (LZ), основанный на этом методе.

Применение динамических словарей позволяет эффективно сжимать повторяющиеся цепочки знаков, независимо от того, являются ли они однородными. Описание алгоритма LZ приведено в электронном конспекте лекций (тема 8).

Для разных типов сообщений – и, соответственно, форматов файлов, - характерны различные вероятности и длины повторяющихся цепочек знаков. В связи с этим эффективность их сжатия может существенно различаться. Например, текстовые данные обычно сжимаются в 2-3 раза, сжатие табличных данных может достигать 8-10 раз, в то же время, изображения в формате jpeg почти не сжимаются LZ-архиватором.

Эта эффективность зависит также от некоторых настраиваемых параметров алгоритма, в частности, используемой длины словаря. В современных архиваторах такого рода настройка как правила выполняется автоматически – с учетом типа обрабатываемых файлов.

Содержание работы

В настоящей работе исследуется сжатие файлов с помощью популярного архиватора WinRAR, в котором реализован алгоритм LZ.

Исследуется степень сжатия файлов различных типов, а также влияние настраиваемых параметров алгоритма сжатия.

Для этого используются следующие типы файлов:
- текст в формате doc и txt;
- табличные данные в формате htm;
- изображения в форматах bmp и jpeg;
- звуковые файлы формата wav.
Оценка сжатия звуковых файлов и изображений в последствие будет использована для сравнения методов сжатия этих типов данных без потерь информации и с потерями (со снижением качества).

Порядок выполнения работы

Используя программу WinRAR, выполнить исследование степени сжатия файлов разных типов при автоматически выбираемых настройках архиватора (режим сжатия “обычный”). Результаты представить в виде табл.1.

Объяснить различия в степени сжатия различных типов файлов исходя из того, как организованы данные в них. Сформулировать и записать соответствующие выводы.

Таблица 11.1
Исследование степени сжатия файлов разных типов

Имя и тип файла	Исходный объем (байт)	Объем после сжатия (байт)	Степень сжатия (%)
.txt			
.doc			
.htm			
.wav			
.bmp			
.jpg			

С помощью WinRAR, выполнить исследование степени сжатия файлов в различных режимах сжатия. В частности, использовать следующие параметры:
- режим (“метод”) сжатия: обычный, скоростной, максимальный (для файла, указанного преподавателем);
- объем используемой памяти для сжатия doc-файла (режим Дополнительно-Параметры сжатия-Сжатие текста-Принудительно) – опребовать три различных значения;
- длина словаря для сжатия bmp-файла (режим Дополнительно-Параметры сжатия-Сжатие полноцветной графики-Принудительно) – опробовать три различных значения.

Результаты испытаний поместить в табл..2
Таблица 2
Исследование режимов сжатия файлов

Имя и тип файла	Режим сжатия	Исходный объем (байт)	Степень сжатия (%)

Объяснить зафиксированное влияние параметров сжатия. Сделать вывод об эффективности автоматического выбора параметров сжатия в WinRAR.

Содержание отчета

Отчет должен содержать следующие разделы:

- Ответы на контрольные вопросы.
- Результаты сжатия файлов в виде таблицы.
- Выводы о проделанной работе
- В нижний колонтитул поместить фамилию, инициалы и номер группы обучаемого (8 пт., Arial, выравнивание по правому краю).

Контрольные вопросы

1. Какого формата лучше сжимается файл?
2. Какие программы применяются для сжатия файлов?
3. Какие методы применяются для сжатия графических файлов?

Библиографический список

1. Информатика. Базовый курс / под ред. С.В. Симоновича. 3-е изд. СПб.: Питер, 2012. 640 с.
2. Справка Windows 7.
3. Лисицин, Л.А. Теоретические основы и методы исследования информационных процессов и систем :[Текст] : учебное пособие /Халин Ю.А., Катыхин, Ю.А. Курск:ЮЗГУ, 2017.-126 с