

**Автономная некоммерческая организация
«Центр поддержки программ развития
Казанского федерального университета»**

**Набережночелнинский институт (филиал) «Казанского
(Приволжского) федерального университета»
Кафедра Бизнес-информатики и математических методов в
экономике**



**Информатика и информационные
технологии. Экономическая
информатика.
Лабораторный практикум
Часть 1
(серия «Бакалавриат»)**

для студентов вузов

Рекомендовано УМО РАЕ по классическому университетскому и техническому образованию в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлениям подготовки:

09.03.03 Прикладная информатика, профиль –
Прикладная информатика в экономике,
38.03.05 Бизнес-Информатика профиль
Реинжиниринг бизнес-процессов предприятий
(организаций)

Протокол № _____ от «___» _____ 2019 г.

Казань, 2019

УДК 004.451
ББК 32.937-018.2
Е 70

**Автономная некоммерческая организация
«Центр поддержки программ развития
Казанского федерального университета»**

Рецензенты:

Лысанов Д.М. Канд. тех. наук, доцент кафедры Бизнес-информатики и математических методов в экономике Набережночелнинского института (филиала) ФГАОУ ВО "Казанский (Приволжский) федеральный университет"

Яковлев Е.В. доктор педагогических наук, профессор, кафедра физики, Нижнекамский химикотехнологический институт (филиал) Казанского государственного технологического университета.

Еремина И.И., Савицкий С.К., Умаров М.Ф., Хаустов С.Л. Информатика и информационные технологии. Экономическая информатика. Лабораторный практикум. Часть 1. (серия «Бакалавриат») /Учебное пособие – Казань: Автономная некоммерческая организация «Центр поддержки программ развития Казанского федерального университета», 2019. – 128 с.: ил.

В учебном пособии приводятся основные понятия по информатике и информационным технологиям, описаны принципы работы с современными прикладными программными средствами. Особое внимание уделено теоретическим основам информатики. Приводятся подробные пояснения, советы и рекомендации по практической работе с описываемыми средствами и технологиями. В конце каждой лабораторной работы содержится перечень контрольных вопросов и заданий, позволяющих самостоятельно проверить качество усвоения изложенного материала.

Рекомендовано УМО РАЕ по классическому университетскому и техническому образованию в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлениям подготовки:

09.03.03 Прикладная информатика, профиль – Прикладная информатика в экономике,

38.03.05 Бизнес-Информатика профиль Реинжиниринг бизнес-процессов предприятий (организаций),

Протокол № _____ от «___» _____ 2019 г.

© И.И. Еремина, С.К. Савицкий, М.Ф. Умаров, С.Л. Хаустов 2019

© Издательство Автономной некоммерческой организация «Центр поддержки программ развития Казанского федерального университета», 2019

Содержание

Раздел 1. Информация, информатика, информационные технологии	4
Тема 1. Измерение и представление информации	4
Лабораторная работа №1. Измерение и представление информации	4
Лабораторная работа №2. Единицы измерения информации.....	6
Тема 2. Естественные и формальные языки. Язык как знаковая система.	11
Лабораторная работа №3. Системы счисления. Перевод чисел из одной позиционной системы счисления в другую	11
Раздел 2. Математические и логические основы информатики	34
Тема 3. Математические и логические основы информатики	34
Лабораторная работа №4. Основные понятия математической логики.	34
Лабораторная работа №5. Базовые логические элементы	52
Раздел 3. Технические средства реализации информационных процессов ..	61
Тема 4. Конструкция ПК	61
Лабораторная работа №6. Основы организации ЭВМ	61
Раздел 4. Программные средства реализации информационных процессов	85
Тема 5. Классификация программного обеспечения ЭВМ	85
Лабораторная работа №7. Программное обеспечение ЭВМ (software).....	85
СПИСОК ЛИТЕРАТУРЫ	124

Раздел 1. Информация, информатика, информационные технологии

Тема 1. Измерение и представление информации **Лабораторная работа №1. Измерение и представление информации**

1. Понятие информации

Слово “информация” происходит от латинского слова *informatio*, что в переводе означает сведение, разъяснение, ознакомление.

Понятие “информация” в курсе информатики является базовым (основным), его нельзя дать через другие, более простые понятия. В геометрии, например, базовыми являются понятия: “точка”, “луч”, “плоскость”. Содержание базовых понятий в любой науке поясняется на примерах или выявляется путем сопоставления с содержанием других понятий.

Информация – это сведения об окружающем мире, которые повышают уровень осведомленности человека.

2. Свойства информации

1. Понятность (на понятном языке, понятными терминами);
2. Полезность (информация должна иметь практическую ценность);
3. Достоверность (информация должна быть правдивой);
4. Актуальность (своевременность);
5. Полнота (информация полна, если ее достаточно для принятия решений);
6. Точность (определяется степенью близости к реальному состоянию объекта, процесса, явления).

Задание 1. Просмотрите презентацию «Понятие информации» и законспектируйте в тетради её содержание.



Презентация 1. Понятие информации

Задание 2. *Каким свойством не обладает информация в следующих случаях 1-6. Таблицу с ответами занесите в тетрадь.*

№	Вопрос	Ответ
1	Афиша о концерте, который отменили	
2	Афиша, часть которой оторвана	
3	Афиша о концерте, который уже состоялся	
4	Афиша на непонятном языке	
5	Афиша о концерте неприятного Вам исполнителя	
6	Афиша, в которой вместо времени указано время суток (например, “Утром состоится концерт...”)	

3. Понятие информационного процесса

Действия, выполняемые с информацией, называются **информационными процессами**. Выделяют следующие информационные процессы:

1. *Процесс передачи информации*, который включает в себя:

- ввод (сбор, получение) информации;
- вывод информации;
- передачу информации.

2. *Процесс обработки (преобразования) информации*

3. *Процесс хранения информации* (в собственной памяти, или на внешних носителях).

Задание 3. *Посмотрите цифровые образовательные ресурсы «Обработка информации», «Хранение информации» и перепишите в тетрадь.*



Рисунок 1. Обработка информации



Рисунок 2. Хранение информации

Задание 4. Заполните примеры информационных процессов в **Таблице 1.** в тетради:

Таблица 1. Примеры информационных процессов.

Название информац. процесса	Примеры информационных процессов
Передача информации	<p><u>Сбор информации</u> об объекте с помощью органов чувств:</p> <p><u>зрения</u> – по цвету клубники (красная, зеленая) можно определить, спелая ягода или нет;</p> <p><u>слуха</u> –</p> <p><u>вкуса</u> –</p> <p><u>обоняния</u> –</p> <p><u>осязания</u> –</p> <p><u>Вывод информации</u> –</p> <p><u>Передача информации</u> –</p>
Обработка информации	<p>1.</p> <p>2.</p> <p>3.</p>
Хранение информации	<p>1.</p> <p>2.</p> <p>3.</p>

Лабораторная работа №2. Единицы измерения информации

Содержательный подход к измерению информации

За *единицу измерения информации* принимается *1 бит* - такое количество информации, которое содержит сообщение, уменьшающее неопределенность знаний в два раза.

Что такое неопределенность знаний? Поясним на примере.

Допустим, вы бросаете монету, загадывая: орел или решка? Любой из вариантов ответа уменьшает неопределенность в 2 раза и, следовательно, количество информации равно 1 биту.

Количество информации (i), содержащееся в сообщении о том, что произошло одно из N равновесных событий, определяется из решения уравнения:

$$2^i = N \quad (1)$$

Прологарифмировав равенство (1) по основанию 2, получим:

$$i \cdot \log_2 2 = \log_2 N,$$

следовательно,

$$i = \log_2 N \quad (2)$$

ЕДИНИЦЫ ИЗМЕРЕНИЯ ИНФОРМАЦИИ

СИМВОЛЬНЫЙ АЛФАВИТ КОМПЬЮТЕРА

- русские (РУССКИЕ) буквы
- латинские (LAT) буквы
- цифры (1, 2, 3, 4, 5, 6, 7, 8, 9, 0)
- математические знаки (+, -, *, /, ^, =)
- прочие символы («», №, %, <, >, ;, #, &)

$N = 2^i$ → $N = 256 = 2^8$ → $i = 8 \text{ бит} = 1 \text{ байт}$

1 байт - это информационный вес одного символа компьютерного алфавита

1 килобайт	=	1 Кб	=	2^{10} байт	=	1024 байта
1 мегабайт	=	1 Мб	=	2^{10} Кб	=	1024 Кб
1 гигабайт	=	1 Гб	=	2^{10} Мб	=	1024 Мб

Рисунок 3. Единицы измерения информации

Алфавитный подход к измерению информации



Рисунок 4. Алфавитный подход к измерению информации

Алфавитный подход к измерению информации не связывает количество информации с содержательным сообщением. Рассмотрим этот подход на примере текста, написанного на каком-нибудь языке, например, на русском. Все множество используемых в языке символов будем называть *алфавитом*. Полное количество символов алфавита будем называть *мощность алфавита*.



Рисунок 5. Количество информации в сообщении

Задание 1. В алфавит мощностью $N=256$ символов можно поместить все необходимые символы: латинские и русские буквы, цифры, знаки арифметических операций, знаки препинания и т.д. Представим себе, что текст, состоящий из 256 символов, поступает последовательно, и в каждый момент времени может появиться любой из них. Тогда по формуле (1):

$$2^i = 256, \rightarrow i=8 \text{ (бит)}$$

Таким образом, один символ алфавита мощностью 256 символов, “весит” 8 бит. Поскольку 8 бит – часто встречающаяся величина, ей присвоили свое название 1 байт:

$$\boxed{1 \text{ байт} = 8 \text{ бит}} \quad (3)$$

Чтобы подсчитать количество информации на одной странице текста, необходимо: количество символов в строке умножить на количество строк на листе. Так, например, если взять страницу текста, содержащую 40 строк по 60 символов в каждой строке, то одна страница такого текста будет содержать

$$60 \cdot 40 = 2400 \text{ (байт информации)}$$

Если требуется подсчитать количество информации, содержащееся в книге из 160 страниц, нужно

$$2400 \cdot 160 = 384000 \text{ (байт)}$$

Уже на этом примере видно, что байт – достаточно мелкая единица. *Для измерения больших объемов информации используются следующие производные от байта единицы:*

$$\boxed{\begin{array}{l} 1 \text{ килобайт} = 1 \text{ Кб} = 2^{10} \text{ байт} = 1024 \text{ байта} \\ 1 \text{ мегабайт} = 1 \text{ Мб} = 2^{10} \text{ Кб} = 1024 \text{ Кб} \\ 1 \text{ гигабайт} = 1 \text{ Гб} = 2^{10} \text{ Мб} = 1024 \text{ Мб} \end{array}} \quad (4)$$

<i>ИМЕНОВАНИЕ</i>	<i>ОБОЗНАЧЕНИЕ</i>	<i>ЗНАЧЕНИЕ В БАЙТАХ</i>	
Килобайт	1 Кб	2^{10} б	1 024 б
Мегабайт	1 Мб	$2^{10} \text{ Кб} = 2^{20} \text{ б}$	1 048 576 б
Гигабайт	1 Гб	$2^{10} \text{ Мб} = 2^{30} \text{ б}$	1 073 741 824 б
Терабайт	1 Тб	$2^{10} \text{ Гб} = 2^{40} \text{ б}$	1 099 511 627 776 б
Петабайт	1 PiB	$2^{10} \text{ Тб} = 2^{50} \text{ б}$	примерно 10^{15} байт
Эксабайт	1 EiB	$2^{10} \text{ PiB} = 2^{60} \text{ б}$	примерно 10^{18} байт
Зеттабайт	1 ZiB	$2^{10} \text{ EiB} = 2^{70} \text{ б}$	примерно 10^{21} байт
Йоттабайт	1 YiB	$2^{10} \text{ ZiB} = 2^{80} \text{ б}$	примерно 10^{24} байт

Задание 2. В алфавите формального (искусственного) языка всего два знака-буквы («+» и «-»). Каждое слово этого языка состоит из двух букв.

Максимальное число слов этого языка:

1) 4	2) 2	3) 8	4) 6
------	------	------	------

Решение.

Решение задачи сводится к поиску количества (N) комбинаций строк длиной (i) 2 символа, составленных из 2 знаков. Следовательно, используя формулу $2^i = N$, получаем $2^2 = 4$.

Ответ: 1.

Задание 3. Алфавит племени содержит всего 8 букв. Какое количество информации несет одна буква этого алфавита?

1) 8 бит	2) 1 байт	3) 3 бита	4) 2 бита
----------	-----------	-----------	-----------

Задание 4. Если вариант теста в среднем имеет объем 20 килобайт (на каждой странице теста 40 строк по 64 символа в строке, 1 символ занимает 8 бит), то количество страниц в тесте равно:

1) 10	2) 16	3) 4	4) 8
-------	-------	------	------

Задание 5. В пяти килобайтах:

1) 5000 байт	2) 5120 байт	3) 500 байт	4) 5000 бит
--------------	--------------	-------------	-------------

Задание 6 Сколько байт в 32 Гбайт?

1) 2^{35}	2) $16 \cdot 2^{20}$	3) 2^{24}	4) 2^{22}
-------------	----------------------	-------------	-------------

Задание 7. Считая, что один символ кодируется одним байтом, подсчитать в байтах количество информации, содержащееся в фразе: “Терпение и труд все перетрут.”

Задание 8. (Задание А3 демоверсии 2004 г.) Шахматная доска состоит из 64 полей: 8 столбцов и 8 строк. Какое минимальное количество бит потребуется для кодирования координат одного шахматного поля?

1) 4	2) 5	3) 6	4) 7
------	------	------	------

Задание 9. (Задание А4 демоверсии 2004 г.) Получено сообщение, информационный объём которого равен 32 битам. Чему равен этот объём в байтах?

1) 5	2) 2	3) 3	4) 4
------	------	------	------

Задание 10. Каждое показание счётчика, фиксируемое в памяти компьютера, занимает 10 бит. Записано 100 показаний этого датчика. Каков информационный объём снятых значений в байтах?

1) 10	2) 100	3) 125	4) 1000
-------	--------	--------	---------

Контрольные вопросы

1. Что такое Информация?
2. Какими свойствами должна обладать информация?
3. Выделите основные информационные процессы.
4. Единицы измерения информации.

Тема 2. Естественные и формальные языки. Язык как знаковая система.

Лабораторная работа №3. Системы счисления. Перевод чисел из одной позиционной системы счисления в другую

Часть 1

Для обмена информацией с другими людьми человек использует

естественные языки (русский, английский, китайский и др.), т.е. информация представляется с помощью естественных языков.

В основе естественного языка лежит алфавит – набор знаков, которые различаются человеком по их начертанию. В основе русского языка лежит кириллица, содержащая 33 знака, английский язык использует латинский алфавит (26 знаков), китайский язык использует алфавит из десятков тысяч знаков (иероглифов).

В соответствии с правилами грамматики из последовательности символов строят основные объекты языка – слова. Из слов в соответствии с правилами синтаксиса строят предложения.

В естественных языках грамматика и синтаксис языка формулируются с помощью большого числа правил, из которых есть исключения, т.к. правила складывались исторически.

Наряду с естественными языками были разработаны формальные (иногда говорят, искусственные) языки (системы счисления, язык алгебры, языки программирования и т.д.). Основное отличие формальных от естественных языков состоит в наличии строгих правил грамматики и синтаксиса.

Например, системы счисления можно рассматривать как формальные языки, имеющие алфавит (цифры) и позволяющие не только именовать и записывать объекты (числа), но и выполнять над ними арифметические операции по строго определенным правилам.

Существуют формальные языки, которые в качестве знаков используют химические формулы, изображения элементов электрических схем, ноты, дорожные знаки, точки и тире (код азбуки Морзе) и т.д.

Представление информации может осуществляться с помощью формальных языков, которые являются знаковыми системами. Каждая знаковая система строится на основе определенного алфавита и правил выполнения операций над знаками (см. фильм «История цифр 1-9».)

Кодирование информации

Кодирование информации – это операция преобразования информации из

одной знаковой системы в другую.

Средством кодирования служит таблица соответствия знаковых систем, которая устанавливает взаимно однозначное соответствие между знаками или группами знаков двух различных знаковых систем.

В процессе обмена информацией часто приходится производить операции кодирования и декодирования информации. Например, при вводе знака алфавита в компьютер путем нажатия соответствующей клавиши на компьютере, происходит кодирование знака, т.е. преобразование его в компьютерный код. При выводе знака на экран монитора или принтер происходит обратный процесс – декодирование, когда из компьютерного кода знак преобразуется в его графическое изображение.

Двоичное кодирование информации.

Для представления информации в компьютере используется двоичное кодирование, т.к. технические устройства компьютера могут сохранять и распознавать не более двух различных состояний (цифр): намагничен / размагничен (участок поверхности магнитного носителя информации), отражает/не отражает (участок поверхности лазерного диска); и т.д.

Информация на компьютере представлена в машинном коде, алфавит которого состоит из цифр (0 и 1). Каждая цифра машинного кода несет информацию в 1 бит.

Системы счисления

Система счисления – это знаковая система, в которой числа записываются по определенным правилам с помощью символов некоторого алфавита, называемыми цифрами.

Системы счисления делятся на непозиционные и позиционные.

Непозиционная система счисления – система счисления, в которой значение цифры не зависит от ее позиции в записи числа.

Таблица 2. Позиционная и непозиционная СС

<i>Позиционная СС</i>	<i>Непозиционная СС</i>
005 = 5*1 (пять)	IX = 10-1 = 9

050 = 5*10 (пятьдесят)	XI = 10+1 = 11
500 = 5*100 (пятьсот)	XX = 10+10 = 20

В позиционных системах счисления величина, обозначаемая цифрой в записи числа, зависит от её положения в числе (позиции, разряда). Количество используемых цифр называется *основанием системы счисления*.

Так, в десятичной системе счисления, основание которой равно 10, различают 10 арабских цифр - 0, 1, 2, ..., 9.

В вычислительной технике широко применяют двоичную, восьмеричную и шестнадцатеричную систему счисления.

Двоичная система счисления имеет основание 2, и, следовательно, ее алфавит состоит из двух цифр - 0 и 1; алфавит восьмеричной системы счисления составляют цифры 0, 1, 2, 3, 4, 5, 6, 7; шестнадцатеричной - десять арабских цифр от 0 до 9 и еще шесть символов - А (10), В (11), С (12), D (13), Е (14), F (15).

Для любой позиционной системы счисления справедливо следующее правило формирования числа на основании входящих в эту систему цифр:

$$y = \sum_{i=0} x_i * k^i$$

или, если расписать сумму в этом выражении,

$$y = x_0 * k^0 + x_1 * k^1 + x_2 * k^2 + \dots,$$

где

у – число;

k – основание системы счисления;

x_i – цифры числа;

i – номер позиции (разряда) числа, начиная с 0.

Так, на основании формулы $y = \sum_{i=0} x_i * k^i$ десятичное число 638₍₁₀₎ представляется следующим образом:

$$y = 8 * 10^0 + 3 * 10^1 + 6 * 10^2 = 8 * 1 + 3 * 10 + 6 * 100 = 8 + 30 + 600 = 638$$

Мы говорим в таком случае, что в этом числе 6 сотен, 3 десятка и 8 единиц.

Исторически, использование для счета десяти цифр связано с тем, что человечество училось считать на пальцах. На самом деле для представления любого числа достаточно алфавита, состоящего только из двух символов, что и реализуется, при хранении информации в памяти электронных устройств. Ячейка памяти в этом случае может находиться в одном из двух состояний, которые кодируются как 0 и 1. Информационная емкость такой ячейки равна 1 биту.

Перевод целых чисел из системы счисления с основанием k в десятичную систему счисления

Число, записанное в позиционной системе счисления с любым основанием,

переводится в десятичную систему счисления по правилу
$$y = \sum_{i=0} x_i * k^i$$
.

Если, например, $45_{(8)}$ – число, записанное в восьмеричной системе счисления, то

$$45_{(8)} = 4 * 8^1 + 5 * 8^0 = 4 * 8 + 5 * 1 = 32 + 5 = 37_{(10)}$$

Число $203_{(5)}$ записано в пятеричной системе счисления, тогда

$$203_{(5)} = 2 * 5^2 + 0 * 5^1 + 3 * 5^0 = 2 * 25 + 0 * 5 + 3 * 1 = 50 + 0 + 3 = 53_{(10)}$$

Меняется только основание системы счисления, алгоритм остается неизменным.

Основание позиционной системы счисления в ней самой всегда записывается как 10; например, в двоичной системе счисления $10_{(2)}$ означает число $2_{(10)}$, а в восьмеричной $10_{(8)}$ означает число $8_{(10)}$.

Чтобы легче осуществлять перевод из системы счисления по любому основанию в десятичную, следует для начала явно пронумеровать разряды исходного числа справа налево, начиная с 0.

Примеры непозиционных систем счисления: унарная (единичная) система счисления, римская система счисления, алфавитная система счисления.

Унарная (единичная) система счисления характеризуется тем, что в ней для записи чисел применяется только один вид знаков – палочка. Каждое число в этой системе счисления обозначалось с помощью строки, составленной из палочек, количество которых равнялось обозначаемому числу. Неудобства такой системы счисления очевидны: это громоздкость записи больших чисел, значение числа сразу не видно, чтобы его получить, нужно сосчитать палочки.

В римской системе счисления для обозначения чисел используются заглавные латинские буквы, являющиеся «цифрами» этой системы счисления:

1	5	10	50	100	500	1000
I	V	X	L	C	D	M

Число в римской системе счисления обозначается набором стоящих подряд «цифр». Значение числа равно:

- 1) сумме значений идущих подряд нескольких одинаковых «цифр» (назовем их группой первого вида);
- 2) разности значений большей и меньшей «цифр», если слева от большей «цифры» стоит меньшая (группа второго вида);
- 3) сумме значений групп и «цифр», не вошедших в группы первого и второго видов.

Примеры.

1. Число 32 в римской системе счисления имеет вид:

$$XXXII = (X+X+X)+(I+I) = 30+2 \text{ (две группы первого вида)}$$

2. Число 444 в римской системе счисления имеет вид:

$$CDXLIV = (D-C)+(L-X)+(V-I) (= 400 + 40 + 4 - \text{три группы второго вида})$$

3. Число 1974:

$$MCMLXXIV = M+(M-C)+L+(X++X)+(V-I) = 1000+900+50+20+4 \text{ (наряду с группами обоих видов в формировании числа участвуют отдельные «цифры»)}$$

4. Число 2005:

$$MMV = (M+M) + V = 1000+1000+5 \text{ (две группы первого вида)}$$

Позиционные системы счисления характеризуется тем, что

количественное значение цифры зависит от ее позиции в числе. Каждая позиционная система счисления имеет определенный алфавит цифр и основание, равное количеству цифр (знаков в ее алфавите).

Наиболее распространенными позиционными системами счисления являются десятичная, двоичная, восьмеричная и шестнадцатеричная.

Десятичная система счисления имеет алфавит из десяти цифр: 0, 1, ..., 9.

Двоичная система счисления имеет алфавит из двух цифр: 0, 1.

Например, в числе 1987_{10} цифра «1» обозначает одну тысячу ($1 \cdot 10^3$),

цифра «9» обозначает девять сотен ($9 \cdot 10^2$),

цифра «8» обозначает восемь десятков ($8 \cdot 10^1$),

цифра «7» обозначает семь единиц ($7 \cdot 10^0$).

В общем виде, если запись числа в системе счисления с основанием $n > 1$ выглядит как $abcd$, то само число равно значению выражения $an^3 + bn^2 + cn^1 + dn^0$.

1. Двоичная система счисления

См. презентацию `sistemy_schislenija.ppt`

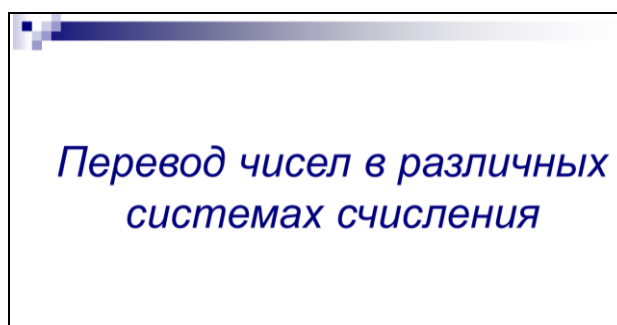


Рисунок 6

Двоичная (бинарная) система счисления имеет основание 2. Ее алфавит – цифры 0 и 1. Для перевода числа из двоичной системы счисления в десятичную

также справедливо правило $y = \sum_{i=0} x_i \cdot k^i$. Представим в десятичном виде число $1101_{(2)}$, или, что то же самое, $\&1101$ (& - амперсant, - этим символом принято указывать то, что следующая за ним запись двоичная).

$$1101_{(2)} = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 13_{(10)}$$

Но двоичная система имеет некоторые приятные особенности, т.к. коэффициентами при степенях двойки в ней могут быть только либо нули (и тогда можно просто игнорировать разряд числа, имеющий значение “0”), либо единицы (умножение на “1” также можно опустить).

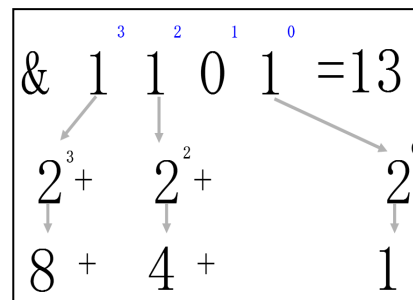


Рисунок 7. Перевод числа из двоичной СС в десятичную

Т.е. достаточно просуммировать “два в соответствующей степени” только в тех позициях двоичного числа, в которых находятся единицы. Степень же, в которую нужно возводить число 2, равна номеру позиции.

Арифметические операции в любой позиционной системе счисления также имеют общую логику.

Таблица 3

1		“Круглые” числа в двоичной СС	
&101	= 5 ₍₁₀₎	&1	= 2 ⁰ = 1
+ 1		&10	= 2 ¹ = 2
<hr/>	&110 = 6 ₍₁₀₎	&100	= 2 ² = 4
+ 1		&1000	= 2 ³ = 8
<hr/>	&111 = 7 ₍₁₀₎	&10000	= 2 ⁴ = 16

Каждый разряд двоичного числа имеет информационную емкость 1 бит. На основании одного двоичного разряда можно закодировать только два десятичных числа - &0=0₍₁₀₎, &1=1₍₁₀₎, на основании двух двоичных разрядов можно закодировать уже четыре десятичных числа – &00=0₍₁₀₎, &01=1₍₁₀₎, &10=2₍₁₀₎, &11=3₍₁₀₎, тремя двоичными разрядами можно представить восемь десятичных чисел и т.д. в соответствии с формулой Хартли.

Таблица 4

2 ⁰	десятичное
1	1

2 ²	2 ¹	2 ⁰	десятичное
1	1	1	7

0	0
---	---

1	1	0	6
1	0	1	5
1	0	0	4
0	1	1	3
0	1	0	2
0	0	1	1
0	0	0	0

2^1	2^0	десятичное
1	1	3
1	0	2
0	1	1
0	0	0

Мы видим, что добавление каждого следующего разряда вдвое увеличивает количество двоичных комбинаций. Графически это может быть представлено так:

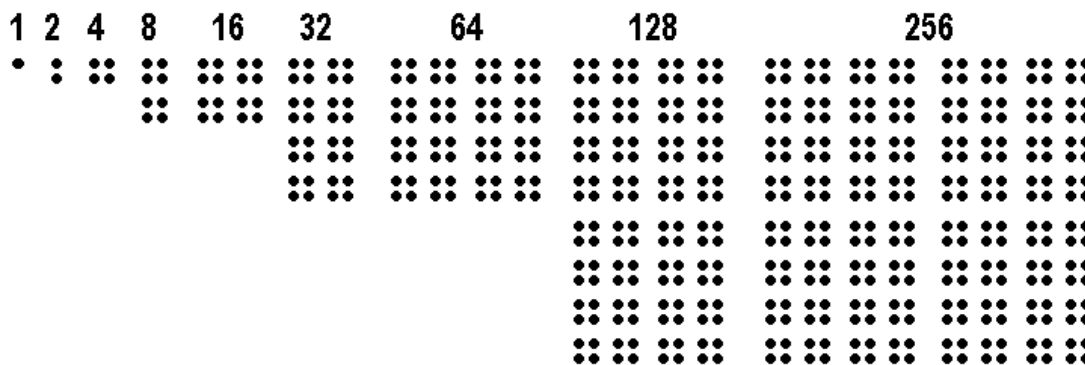


Рисунок 8. Каждый следующий разряд двоичного числа удваивает количество возможных комбинаций из нулей и единиц

Таблицу степеней числа 2 от 2^0 до 2^{10} следует знать наизусть.

Таблица 5

N	0	1	2	3	4	5	6	7	8	9	10
2^N	1	2	4	8	16	32	64	128	256	512	1024

Открытие двоичного способа представления чисел приписывают китайскому императору Фо Ги, жизнь которого относится к 4-му тысячелетию до новой эры. Известный немецкий математик *Лейбниц* (1646-1716) в 1697 г. разработал

правила двоичной арифметики. Он подчеркивал, что "вычисление с помощью двоек, то есть 0 и 1, в вознаграждение его длиннот, является для науки основным и порождает новые открытия, которые оказываются полезными впоследствии, даже в практике чисел, а особенно в геометрии: причиной чего служит то обстоятельство, что при сведении чисел к простейшим началам, каковы 0 и 1, всюду выявляется чудесный порядок".

Блестящие предсказания Лейбница сбылись только через 2,5 столетия, когда именно двоичная система счисления нашла применение в качестве универсального способа кодирования информации в компьютерах.

Перевод целого числа из двоичной системы счисления в десятичную.

Пример.

$$101_2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 4 + 0 + 1 = 5_{10}$$

Задание 1.

Переведите число 101101_2 в десятичную систему счисления.

Решение.

$$101101_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 32 + 8 + 4 + 1 = 45_{10}$$

Ответ: $101101_2 = 45_{10}$

Задание 2. Переведите числа из заданных систем счисления в десятичную систему.

Методические указания.

Перевод чисел в десятичную систему осуществляется путем составления суммы степенного ряда с основанием той системы, из которой число переводится. Затем подсчитывается значение этой суммы.

Примеры:

а) Перевести $10101101,101_2 \rightarrow 10$ с.с.

$$10101101,101_2 =$$

$$1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 173,625_{10}$$

Ответ: $10101101,101_2 = 173,625_{10}$.

б) Перевести $703,04_8 \rightarrow 10$ с.с.

$$703,04_8 = 7 \cdot 8^2 + 0 \cdot 8^1 + 3 \cdot 8^0 + 0 \cdot 8^{-1} + 4 \cdot 8^{-2} = 451,0625_{10}$$

Ответ: $703,04_8 = 451,0625_{10}$.

в) Перевести $B2E,4_{16} \rightarrow 10$ с.с.

$$B2E,4_{16} = 11 \cdot 16^2 + 2 \cdot 16^1 + 14 \cdot 16^0 + 4 \cdot 16^{-1} = 2862,25_{10}$$

Ответ: $B2E,4_{16} = 2862,25_{10}$.

Перевод целого числа из десятичной системы счисления в двоичную

Алгоритм

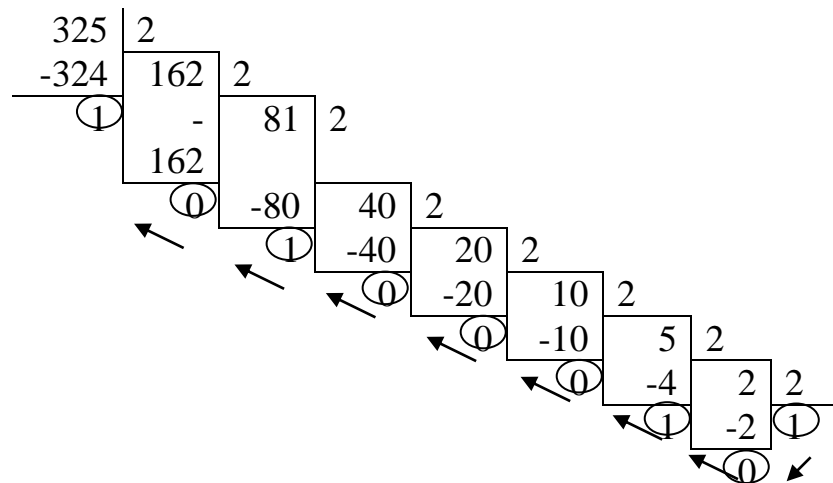
1. Последовательно выполнить деление исходного целого десятичного числа и получаемых целых частных на основание системы (на 2) до тех пор, пока не получится частное, меньшее делителя (т.е. меньшее 2).

2. Записать полученные остатки в обратной последовательности.

Пример.

Решение.

$$325_{10} = 101000101_2$$

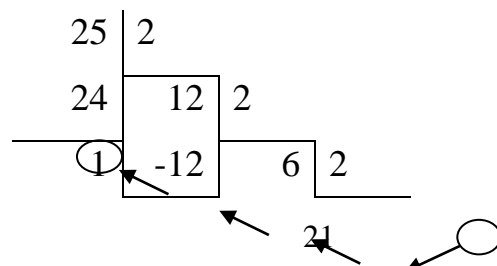


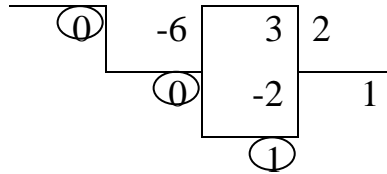
Задание 3.

Как представляется число 25_{10} в двоичной системе счисления?

1) 1001_2	2) 11001_2	3) 10011_2	4) 11010_2
-------------	--------------	--------------	--------------

Решение.





$25_{10} = 10011_2$, что соответствует ответу №2.

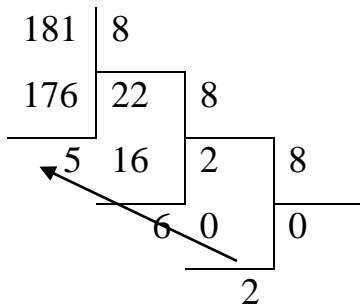
Ответ: 2.

Задание 4. Переведите целые числа из десятичной системы в восьмеричную, шестнадцатеричную и двоичную системы.

Методические указания.

Перевод целых десятичных чисел в восьмеричную, шестнадцатеричную и двоичную системы осуществляется последовательным делением десятичного числа на основание той системы, в которую оно переводится, до тех пор, пока не получится частное, равное нулю. Число в новой системе записывается в виде остатков от деления, начиная с последнего.

Примеры: а) Перевести $181_{10} \rightarrow 8$ с.с.



В таблице представлено деление:

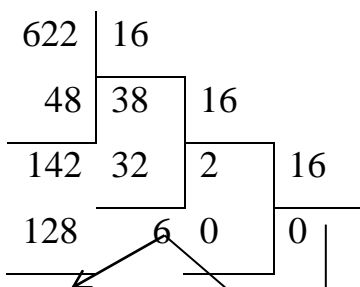
$$181 : 8 = 22 \text{ (остаток 5)}$$

$$22 : 8 = 2 \text{ (остаток 6)}$$

$$2 : 8 = 0 \text{ (остаток 2)}$$

Ответ: $181_{10} = 265_8$.

б) Перевести $622_{10} \rightarrow 16$ с.с.



14

2

В таблице представлено деление:

$$622:16=38 \text{ (остаток } 14_{10} = E_{16})$$

$$38:16=2 \text{ (остаток 6)}$$

$$2:16=0 \text{ (остаток 2)}$$

Ответ: $622_{10} = 26E_{16}$.

В аппаратной основе ЭВМ лежат двухпозиционные элементы, которые могут находиться только в двух состояниях, одно из которых обозначается 0, а другое — 1. Поэтому арифметико-логической основой ЭВМ является двоичная система счисления.

Двоичная система счисления. Используется две цифры: 0 и 1. В двоичной системе любое число может быть представлено в виде $X = b_M b_{M-1} \dots b_j \dots b_1 b_0 . b_{-1} b_{-2} \dots$, где b_j либо 0, либо 1.

Эта запись соответствует сумме степеней числа 2, взятых с указанными коэффициентами:

$$X = b_M \cdot 2^M + b_{M-1} \cdot 2^{M-1} + \dots + b_1 \cdot 2^1 + b_0 \cdot 2^0 + b_{-1} \cdot 2^{-1} + b_{-2} \cdot 2^{-2} + \dots$$

Восьмеричная система счисления. Используется восемь цифр: 0, 1, 2, 3, 4, 5, 6, 7. Употребляется в ЭВМ как вспомогательная система для записи информации в сокращенном виде. Для представления одной цифры восьмеричной системы используется три двоичных разряда (триада) (табл. 1).

Шестнадцатеричная система счисления. Для изображения чисел используется 16 цифр. Первые десять цифр этой системы обозначаются цифрами от 0 до 9, а старшие шесть цифр — латинскими буквами: А (10), В (11), С (12), D(13), E(14), F(15). Шестнадцатеричная система, также как и восьмеричная, используется для записи информации в сокращенном виде. Для представления одной цифры шестнадцатеричной системы счисления используется четыре двоичных разряда (тетрада) (табл. 5).

Таблица 5

Алфавиты позиционных систем счисления (с.с.)

Двоичная с.с. (Основание 2)	Восьмеричная с.с. (основание 8)	Десятичная с.с. (Основание 10)	Шестнадцатеричная с.с. (Основание 16)
	Двоичные триады		Двоичные тетрады
0	00	0	0000
1	0	1	0001
	00	2	0010
	1	3	0011
	01	4	0100
	0	5	0101
	01	6	0110
	1	7	0111
	10	8	1000
	0	9	1001
	10		A
	1		B
	11		C
	0		D
	11		E
	1		F

Задание 5. Переведите числа из восьмеричной и шестнадцатеричной систем счисления в двоичную систему.

Методические указания.

Для перевода восьмеричного или шестнадцатеричного числа в двоичную форму достаточно заменить каждую цифру этого числа соответствующим трехразрядным двоичным числом (триадой) или четырехразрядным двоичным числом (тетрадой), при этом отбрасывают незначащие нули в старших и младших (после запятой) разрядах.

Примеры:

$$а) \begin{array}{cccc} \underline{2} & \underline{0} & \underline{4} & \underline{4}_8 \\ \underline{010000100} & & \underline{100} & \\ \hline & & & 10000100,1_2; \end{array}$$

$$б) \begin{array}{ccc} \underline{6} & \underline{C} & \underline{3} \\ \underline{0110} & \underline{1100} & \underline{0011} \\ \hline & & \underline{1010} \end{array}, \underline{A}_{16} = 11011000011,101_2.$$

Задание 6. Переведите числа из восьмеричной и шестнадцатеричной систем счисления в двоичную систему.

Методические указания.

Для перехода от двоичной к восьмеричной (шестнадцатеричной) системе поступают следующим образом: двигаясь от точки влево и вправо, разбивают двоичное число на группы по три (четыре) разряда, дополняя при необходимости нулями крайние левую и правую группы. Затем триаду (тетраду) заменяют соответствующей восьмеричной (шестнадцатеричной) цифрой.

Примеры:

а) Перевести $10011001111,0101_2 \rightarrow 8 \text{ с. с.}$

$$\begin{array}{cccc} \underline{010} & \underline{011} & \underline{001} & \underline{111} \\ \underline{2} & \underline{3} & \underline{1} & \underline{7} \\ \hline & & & \underline{010} & \underline{100} \\ & & & \underline{2} & \underline{4} \\ \hline & & & & 2317,24_8. \end{array}$$

б) Перевести $10111111011,100011_2 \rightarrow 16 \text{ с. с.}$

$$\begin{array}{ccc} \underline{0101} & \underline{1111} & \underline{1011} \\ \underline{5} & \underline{F} & \underline{B} \\ \hline & & \underline{1000} & \underline{1100} \\ & & \underline{8} & \underline{C} \\ \hline & & & 5FB,8C_{16}. \end{array}$$

Задание 7. Переведите числа из восьмеричной в шестнадцатеричную систему счисления и из шестнадцатеричной в восьмеричную систему счисления.

Методические указания.

Перевод из восьмеричной в шестнадцатеричную систему и обратно осуществляется через двоичную систему с помощью триад и тетрад.

Пример.

Перевести $135,14_8 \rightarrow 16 \text{ с.с.}$

$$\underbrace{1}_{001} \underbrace{3}_{011} \underbrace{5}_{101}, \underbrace{1}_{001} \underbrace{4}_{100}_8 = 1011101,0011_2 = \underbrace{0101}_5 \underbrace{1101}_D, \underbrace{0011}_3_2 = 5D,3_{16}$$

Ответ: $135,14_8 = 5D,3_{16}$.

Перевод дробного числа из двоичной системы счисления в десятичную.

Пример.

$$111,01_2 = 1*2^2 + 1*2^1 + 1*2^0 + 1*2^{-1} + 1*2^{-2} = 1*4 + 1*2 + 1 + 0*\frac{1}{2} + 1*\frac{1}{4} =$$

$$= 4+2+1+0,5+0,25 = 7,75_{10}$$

Часть 2

1. Шестнадцатеричная система счисления

Система счисления с основанием 16 интересна тем, что она включает в себя больше разрядов, чем десятичная, и соответственно десяти арабских цифр недостаточно для алфавита этой системы счисления, поэтому в качестве недостающих цифр в ней используются буквы латинского алфавита.

Для обозначения того, что запись является шестнадцатеричным числом, принято использовать также символ #.

Таблица 6

Основание СС (k)	Цифры, составляющие алфавит СС	Пример записи
2	0, 1	&101011111
10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	351
16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F 10 11 12 13 14 15	#15f

Для шестнадцатеричной системы счисления действуют те же правила перевода, что и для всякой позиционной системы счисления.

$$\# 4^1 b^0 = 75$$

$$4^1 \times 16^1 + b^0 \times 16^0$$

$$4 \times 16 + 11 \times 1$$

$$4b_{(16)} = 75_{(10)}$$

Рисунок 9. Перевод из СС с основанием 10 в СС с основанием 16.

$$180 \mid 16$$

$$176 \quad 11 = b$$

$$4$$

$$180_{(10)} = b4_{(16)}$$

Рисунок 10. Перевод из СС с основанием 16 в СС с основанием 10.

Вавилонская (шестидесятеричная) система счисления

Исторический интерес представляет так называемая «вавилонская», или шестидесятеричная система счисления, весьма сложная, существовавшая в Древнем Вавилоне, за две тысячи лет до н.э.

Это первая известная нам система счисления, основанная на позиционном принципе. Система вавилонян сыграла большую роль в развитии математики и астрономии, ее следы сохранились до наших дней. Так, мы до сих пор делим час на 60 минут, а минуту на 60 секунд. Точно так же, следуя примеру вавилонян, окружность мы делим на 360 частей (градусов).

Кодирование двоичным кодом

Информация любого типа: символьная, графическая, звуковая, командная для представления на электронных носителях кодируется на основании алфавита, состоящего только из двух символов (0, 1). Информация представленная в аналоговом виде, для того, чтобы быть сохраненной в электронной памяти, *оцифровывается* и приводится к двоичному коду.

Каждая ячейка электронной памяти обладает информационной ёмкостью 1 бит. Физически, в зависимости от способа регистрации информации, это может быть конденсатор, находящийся в одном из двух состояний: разряжен (0), заряжен (1); элемент магнитного носителя: размагничен (0), намагничен (1); элемент поверхности оптического диска: нет лунки (0), есть лунка (1). Одним из

первых носителей информации, представленной в двоичном коде, была бумажная перфокарта, пробитое отверстие на которой означало 1, а цельная поверхность 0.

Перевод дробного числа из десятичной системы счисления в двоичную.

Алгоритм.

Последовательно умножать (в исходной системе счисления) данное число и получаемые дробные части произведений на основание новой системы (на 2) до тех пор, пока дробная часть произведения не станет равной нулю или будет достигнута требуемая точность представления данного числа.

Полученные целые части произведений, являющиеся цифрами в числа в новой системе счисления, привести в соответствие с алфавитом новой системе счисления.

Составить дробную часть числа в новой системе счисления, начиная с целой части первого произведения.

Пример.

$$0,5625_{10} = 0,1001_2.$$

Решение.

0,	5625
	x 2
1	1250
	x 2
0	2500
	x 2
0	5000
	x 2
1	0000

Пример.

$$0,7_{10} \approx x_2$$

Решение.

0,	7
	x 2
1	4
	x 2
0	8
	x 2
1	6
	x 2
1	2
	x 2
0	4

...

Очевидно, что этот процесс может продолжаться до бесконечности. Обрывают процесс на шаге, когда получена требуемая точность вычисления (количество знаков после запятой).

$$0,7_{10} \approx 0,10110_2$$

Задание 8. Переведите правильные десятичные дроби из десятичной системы в восьмеричную, шестнадцатеричную и двоичную системы.

Методические указания.

Для перевода правильной десятичной дроби в другую систему эту дробь последовательно умножают на основание той системы, в которую она переводится. При этом умножаются только дробные части полученных произведений. Если в результате умножения на некотором шаге дробная часть становится равной нулю, это означает, что получили конечную дробь в новой системе счисления. В новой системе дробь записывается в виде целых частей полученных произведений, начиная с первого. Не все конечные дроби в результате перевода станут конечными, зачастую в новой системе счисления получается бесконечная дробь.

Примеры

а) Перевести $0,3125_{10} \rightarrow 8$ с.с.

0,	3125		0,	3125
	x 8			x 8
2,	5000		2,	5000
	x 8			x 8
4	0000	↓	4	0000

Условно отделим вертикальной чертой целую и дробную части полученных произведений.

Результат перевода есть последовательность цифр, состоящих из целых частей произведений, записанная сверху вниз.

Ответ: $0,3125_{10} = 0,24_8$.

б) Перевести с точностью до 6 знаков после запятой $0,65_{10} \rightarrow 2$ с.с.

0,	65 ×2	0,65×2=1,3
1	3 ×2	0,3×2=0,6
0	6 ×2	0,6×2=1,2
1	2 ×2	0,2×2=0,4
0	4 ×2	0,4×2=0,8
0	8 ×2	0,8×2=1,6
1	6 ×2	0,6×2=1,2

Далее умножаем дробную часть полученного произведения. Каждый раз умножаем только дробную часть произведения.

Обратите внимание, в результате перевода получилась бесконечная периодическая дробь

Ответ: $0,65_{10} \approx 0,10(1001)_2$.

Задание 9. Переведите неправильные десятичные дроби из десятичной системы в восьмеричную, шестнадцатеричную и двоичную системы.

Методические указания.

Для перевода неправильной десятичной дроби в систему счисления с недесятичным основанием необходимо отдельно перевести целую часть и отдельно дробную.

Пример.

Перевести $23,125_{10} \rightarrow 2$ с.с.

Переведем целую часть:

$$\begin{array}{r}
 23 \quad | \quad 2 \\
 22 \quad | \quad 11 \quad 2 \\
 \hline
 1 \quad | \quad 10 \quad 5 \quad 2 \\
 \quad | \quad 1 \quad 4 \quad 2 \quad 2 \\
 \quad | \quad \quad 1 \quad 2 \quad 1 \quad 2 \\
 \quad | \quad \quad \quad 0 \quad 0 \quad 0 \\
 \quad | \quad \quad \quad \quad 1
 \end{array}$$

2) Переведем дробную часть:

0,	125 x 2
0	25 x 2
0	5 x 2
1	0

Получили $23_{10} = 10111_2$; $0,125_{10} = 0,001_2$. Результат перевода
 $23,125_{10} = 10111,001_2$.

Ответ: $23,125_{10} = 10111,001_2$.

Задания для самостоятельной работы

Задание 1.

<i>Вариант</i>	<i>Переведите числа в 10-ю с.с.</i>	<i>Переведите десятичные числа в 2-ю, 8-ю, и 16-ю с.с.</i>	<i>Восьмеричное число переведите в 16-ю с.с., а шестнадцатеричное – в 8-ю с.с.</i>
1.	10010011111,101 ₂ 1372,12 ₈ 3CA,7D ₁₆	1802 286,06	1263,71 ₈ 2BA,2C ₁₆
2.	11100101010,011 ₂ 2136,31 ₈ 1C3,A2 ₁₆	1731 476,91	3472,62 ₈ 4CA,27 ₁₆
3.	11001100111,011 ₂ 1742,36 ₈ 123E,4D ₁₆	1660 438,76	1724,31 ₈ 2AF,3C ₁₆
4.	11101011101,1001 ₂ 1467,63 ₈ 1AF,73 ₁₆	1589 362,87	1273,56 ₈ 30A,E0F ₁₆
5.	101011010110,001 ₂ 1523,24 ₈ 2A7,3E ₁₆	1518 305,37	1623,72 ₈ 5C2,C7 ₁₆
6.	11001100011,1001 ₂ 1273,56 ₈ 30A,E0F ₁₆	1682 324,93	12372,41 ₈ 1D2,7D ₁₆
7.	10011010111,011 ₂ 1623,72 ₈ 5C2,C7 ₁₆	1846 457,21	1735,12 ₈ 5AD,4D ₁₆
8.	11000001111,011 ₂ 1735,66 ₈ 23A,EF ₁₆	2010 343,43	2451,23 ₈ 2BA,D3 ₁₆
9.	10000111111,1001 ₂ 1327,46 ₈ 3CD,BA ₁₆	1933 381,93	1372,12 ₈ 3CA,7D ₁₆
10.	11100001101,011 ₂	1856	2136,31 ₈

<i>Вариант</i>	<i>Переведите числа в 10-ю с.с.</i>	<i>Переведите десятичные числа в 2-ю, 8-ю, и 16-ю с.с.</i>	<i>Восьмеричное число переведите в 16-ю с.с., а шестнадцатеричное – в 8-ю с.с.</i>
	1523,74 ₈ 4BA, 2F ₁₆	419,96	1C3, A2 ₁₆
11.	11011110110,101 ₂ 4123,17 ₈ 1C3, A5 ₁₆	1779 400,01	1742,36 ₈ 123E, 4D ₁₆
12.	110010010111,1001 ₂ 1272,12 ₈ 3AD, 7D ₁₆	1702 153,63	5123,14 ₈ 1B3, 4D ₁₆
13.	11100110101,1011 ₂ 1071,21 ₈ 5DC, F2 ₁₆	1625 172,04	1263,71 ₈ 2BA, 2C ₁₆
14.	10011010111,011 ₂ 2372,12 ₈ 1F2, 7B ₁₆	1548 191,11	3472,62 ₈ 4CA, 27 ₁₆
15.	11110010101,1001 ₂ 1574,61 ₈ 35C, F1 ₁₆	1702 210,96	1724,31 ₈ 2AF, 3C ₁₆
16.	11000011010,1001 ₂ 6123,51 ₈ 13A, C2 ₁₆	1856 229,74	1272,12 ₈ 3AD, 7D ₁₆
17.	10011000111,1111 ₂ 5412,63 ₈ 52A, 17 ₁₆	1794 248,2	1071,21 ₈ 5DC, F2 ₁₆
18.	11101101101,1001 ₂ 5123,14 ₈ 1B3, 4D ₁₆	1732 267,72	2372,12 ₈ 1F2, 7B ₁₆
19.	11101011001,0101 ₂ 1263,71 ₈ 2BA, 2C ₁₆	1670 571,58	1742,36 ₈ 123E, 4D ₁₆
20.	10101110111,0101 ₂ 3472,62 ₈ 4CA, 27 ₁₆	1608 590,72	1467,63 ₁₈ 1AF, 73 ₁₆
21.	10101101111,011 ₂ 1724,31 ₈ 2AF, 3C ₁₆	1732 495,32	1523,24 ₈ 2A7, 3E ₁₆
22.	11100101101,1011 ₂ 1275,46 ₈ 23A, E7 ₁₆	1856 552,5	1735,66 ₈ 23A, EF ₁₆

<i>Вариант</i>	<i>Переведите числа в 10-ю с.с.</i>	<i>Переведите десятичные числа в 2-ю, 8-ю, и 16-ю с.с.</i>	<i>Восьмеричное число переведите в 16-ю с.с., а шестнадцатеричное – в 8-ю с.с.</i>
23.	10011010111,011 ₂ 12372,41 ₈ 1D2,7D ₁₆	1980 533,51	1327,46 ₈ 3CD, BA ₁₆
24.	11011100010,1101 ₂ 1735,12 ₈ 5AD, 4D ₁₆	1805 514,58	1523,74 ₈ 4BA, 2F ₁₆
25.	10100101111,101 ₂ 2451,23 ₈ 2BA, D3 ₁₆	1630 609,11	4123,17 ₈ 1C3, A5 ₁₆

Задание 2.

Внимательно изучить теоретический материал и придумать «нелепую» историю, в которой числовые данные приведены в системе счисления, отличной от десятичной.

Например:

история «Бабушка»

У моей бабушки, пенсионерки в возрасте 144 лет, недалеко от Елабуги есть маленький сад площадью всего 20 га. Этот год был очень урожайным, и именно поэтому на своей плантации бабушка вырастила 122 кг огурцов, 22 кг помидоров и 320 кг яблок. Каждое лето на эту дачу приезжает множество бабушкиных знакомых. Вот и в этом году к ней приехала на выходные её подруга, которой 152 года; моя бабушка подарила ей $1/23$ всех яблок, а остальные 304 кг оставила себе. Её подруга была очень рада и пообещала нам приехать в гости ещё раз 50 октября в мои осенние каникулы.

Комментарий

В этой истории была использована 6-я СС.

В 10-й СС бабушке – 64 года, площадь сада 12 га, а бабушка вырастила 50 кг огурцов, 14 кг помидоров и 120 кг яблок. Бабушкиной подруге 68 лет, бабушка подарила ей $1/15$ всех яблок, а остальные 112 кг оставила себе, подруга пообещала приехать 30 октября.1

Задание 3.

В таблице представлены числа, записанные в различных СС. Среди них встречаются и недопустимые записи, которые нужно вычеркнуть, оставшиеся после вычёркивания «правильные» числа необходимо перевести в 10-ю систему, и в ней выполнить остальные задания. Ответы приводятся в 10-й СС.

По данной таблице требуется посчитать:

1. Сумму чисел в каждой строке.
2. Произведение чисел в столбцах.
3. Сумму чисел на главной диагонали.
4. Произведение чисел на побочной диагонали.

157	14 3	103	196
23 7	24 5	314	356
25 6	1 F16	177	168
33 3	112	189	235

Контрольные вопросы

Как осуществляется перевод чисел из r -й с.с. в десятичную?

Как перевести целое десятичное число в r -ю с.с.?

Как перевести правильную десятичную дробь в r -ю с.с.?

Как перевести неправильную десятичную дробь в r -ю с.с.?

Раздел 2. Математические и логические основы информатики

Тема 3. Математические и логические основы информатики

Лабораторная работа №4. Основные понятия математической логики.

1. Основные понятия математической логики

Алгебра логики – это раздел математики, изучающий высказывания,

рассматриваемые со стороны их логических значений (истинности и ложности) и логических операций над ними [4].

Логическое высказывание – это любое повествовательное предложение, в отношении которого можно однозначно сказать, истинно оно или ложно [4].

Для обозначения истины (истинного высказывания) используется символ 1, а для обозначения лжи (ложного высказывания) используется символ 0.

Рассмотрим примеры логических высказываний (см. **Таблицу 6**):

Таблица 6. Примеры логических выражений

Предложение	Характеристика с точки зрения алгебры логики
Иваново – Родина Первого Совета	Истинное логическое высказывание
За зимой наступит весна	Истинное логическое высказывание
В городе Иваново проживают только граждане России	Ложное логическое высказывание
После дождя всегда тепло	Ложное логическое высказывание
После вторника будет выходной	<u>Не является логическим высказыванием</u> , т.к. не известно, о каком человеке, каком месяце и дне идет речь (если у человека текущий график работы, возможно, что у него в среду будет выходной, в противном случае среда – рабочий день; если в среду будет праздничный день, например, 8 марта, то этот день также будет выходным)

Употребляемые в обычной речи слова и словосочетания «не», «и», «или», «если...то», «тогда и только тогда» и др. позволяют из уже заданных высказываний строить более сложные высказывания. Такие слова и словосочетания называют логическими связками. Высказывания, образованные с помощью логических связок – называют составными высказываниями. Высказывания, не являющиеся составными, называют элементарными.

Просмотрите презентацию и разберите понятия, представленные в ней.



Рисунок 11

Для обозначения логических высказываний, им назначают имена. Например, если **A** – высказывание «В четверг был дождь», **B** – высказывание «В пятницу было солнечно», то составное высказывание «В четверг был дождь, а в пятницу было солнечно», можно записать в виде:

A и B.

Здесь **A**, **B** – логические высказывания (могут быть либо истинными, либо ложными), **и** – логическая связка.

Каждая логическая связка рассматривается как операция над логическими высказываниями и имеет свое название и обозначение (см. **Таблицу 7**):

Таблица 7. Логические связки

№	Логическая связка	Название	Обозначение	Высказывание	Математическая запись
1	и	конъюнкция логическое умножение	\wedge , & *, And	A и B	$A \wedge B$, A & B $A * B$, A And B
2	или	дизъюнкция логическое сложение	\vee +, Or	A или B	$A \vee B$ $A + B$, A Or B
3	не	инверсия, логическое отрицание	\neg , \bar{A} , Not	не A	$\neg A$, \bar{A} , Not A
4	Если...то	импликация, логическое следование	\rightarrow , \Rightarrow	Если A, то B	$A \rightarrow B$ $A \Rightarrow B$
5	тогда и только тогда	эквивалентность, равносильность, логическое тождество	\leftrightarrow , \equiv \Leftrightarrow , \sim	A тогда и только тогда, когда B	$A \leftrightarrow B$, $A \equiv B$ $A \Leftrightarrow B$, $A \sim B$

Логические связки удобно определять с помощью таблиц истинности

составляющих высказываний и указывается соответствующее значение истинности высказывания, полученного в результате действия логической операции:

отрицание

A	\neg A
1	0
0	1

дизъюнкция

A	B	$A \vee B$
1	1	1
1	0	1
0	1	1
0	0	0

конъюнкция

A	B	$A \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0

импликация

A	B	$A \Rightarrow B$
1	1	1
1	0	0
0	1	1
0	0	1

эквиваленция

A	B	$A \Leftrightarrow B$
1	1	1
1	0	0
0	1	0
0	0	1

Импликацию можно выразить через дизъюнкцию и отрицание:

$$A \rightarrow B = \neg A \vee B \quad (1)$$

Эквивалентность можно выразить через отрицание, дизъюнкцию и конъюнкцию:

$$A \leftrightarrow B = (\neg A \vee B) \wedge (\neg B \vee A) \quad (2)$$

Вычисление значения логического выражения производится слева направо в соответствии с таблицей истинности (см. Таблицу 3) и приоритетом выполнения логических операций (см. Таблицу 4). Порядок выполнения операций можно менять, используя круглые скобки.

Таблица 8. Таблица истинности

A	B	$A \vee B$	$A \wedge B$	$\neg A$
0	0	0	0	1
0	1	1	0	1
1	0	1	0	0
1	1	1	1	0

Таблица 9. Приоритет выполнения логических операций

Приоритет операции	Логическая операция
Первый (высший)	Логическое отрицание
Второй	Конъюнкция (логическое умножение)
Третий	Дизъюнкция (логическое сложение)
Четвертый	Импликация (следование)
Пятый (низший)	Эквивалентность (равносильность)

Изучите основные логические операции в презентации

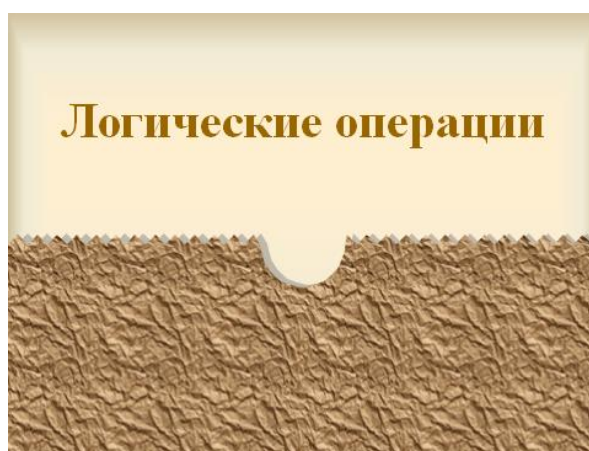


Рисунок 12

2. Основные законы алгебры логики

В алгебре логики выполняются следующие основные законы, позволяющие производить тождественные преобразования логических выражений (см. **Таблицу 10.**)

Таблица 10. Основные законы алгебры логики

Закон	Для ИЛИ	Для И	
Переместительный	$x \vee y = y \vee x$	$x \wedge y = y \wedge x$	(3)
Сочетательный	$x \vee (y \vee z) = (x \vee y) \vee z$	$x \wedge (y \wedge z) = (x \wedge y) \wedge z$	(4)
Распределительный	$x \wedge (y \vee z) = x \wedge y \vee x \wedge z$	$x \vee y \wedge z = (x \vee y) \wedge (x \vee z)$	(5)
Правила Де Моргана	$\neg(x \vee y) = \neg x \wedge \neg y$	$\neg(x \wedge y) = \neg x \vee \neg y$	(6)
Идемпотенции	$x \vee x = x$	$x \wedge x = x$	(7)

Закон	Для ИЛИ	Для И	
Поглощения	$x \vee x \wedge y = x$	$x \wedge (x \vee y) = x$	(8)
Склеивания	$x \wedge y \vee (\neg x) \wedge y = y$	$(x \vee y) \wedge (\neg x \vee y) = y$	(9)
Операция с переменной с ее инверсией	$x \vee (\neg x) = 1$	$x \wedge (\neg x) = 0$	(10)
Операция с константами	$x \vee 1 = x; x \vee 0 = x$	$x \wedge 1 = x; x \wedge 0 = 0$	(11)
Операция двойного отрицания	$\neg(\neg x) = x$		(12)

Примеры задач с таблицами истинности:

Изучите примеры задач с таблицами истинности из презентации:



Рисунок 13

Задание 1. (Задание A11 демоверсии 2004 г.)

Для какого имени истинно высказывание:

$\neg(\text{Первая буква имени гласная} \rightarrow \text{Четвертая буква имени согласная})$

- | | | | |
|----------|----------|----------|----------|
| 1) ЕЛЕНА | 2) ВАДИМ | 3) АНТОН | 4) ФЕДОР |
|----------|----------|----------|----------|

Решение.

Введем обозначения для высказываний:

$A = \text{«Первая буква имени гласная»}$ (13)

$B = \text{«Четвертая буква имени согласная»}$ (14)

тогда наше высказывание примет вид: $\neg(A \rightarrow B)$. Чтобы преобразовать высказывание, воспользуемся тождествами (1), (6), (12):

$$\begin{aligned} & \text{(1)} & \text{(6)} & \text{(12)} \\ \neg(A \rightarrow B) &= \neg((\neg A) \vee B) &= \neg(\neg A) \wedge (\neg B) &= A \wedge (\neg B) \end{aligned}$$

Используя обозначения (13), (14), получим, что исходное высказывание равносильно следующему:

Первая буква гласная \wedge \neg (Четвертая буква имени согласная), \leftrightarrow

Первая буква гласная \wedge Четвертая буква имени гласная.

Этому условию удовлетворяет только имя АНТОН (вариант ответа №3).

Ответ: 3

Задание 2. (Задание A12 демоверсии 2004 г.)

Какое логическое выражение равносильно выражению $\neg(A \vee \neg B)$

1) $A \vee B$	2) $A \wedge B$	3) $\neg A \vee \neg B$	4) $\neg A \wedge B$
---------------	-----------------	-------------------------	----------------------

Решение.

Чтобы преобразовать высказывание, воспользуемся законами (6), (12):

$$\begin{aligned} & \text{(6)} & \text{(12)} \\ \neg(A \vee \neg B) &= \neg A \wedge \neg(\neg B) &= \neg A \wedge B, \text{ что соответствует ответу №4.} \end{aligned}$$

Ответ: 4

Задание 3. (Задание A13 демоверсии 2004г., A11 демоверсий 2005, 2006г.)

Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z. Дан фрагмент таблицы истинности выражения F:

X	Y	Z	F
0	0	0	1
0	0	1	0
0	1	0	1

Какое выражение соответствует F?

1) $\neg X \wedge \neg Y \wedge Z$	2) $\neg X \vee \neg Y \vee Z$	3) $X \vee Y \vee \neg Z$	4) $X \vee Y \vee Z$
------------------------------------	--------------------------------	---------------------------	----------------------

Решение.

Способ 1. Наличие двух единиц в столбце F позволяет предположить использование дизъюнкции в логическом выражении. F принимает значение, равное 0, при $X=0, Y=0, Z=1$, что соответствует логической сумме $X \vee Y \vee \neg Z$.

При проверке этой формулы при значениях первой и третьей строки, получаем верные значения F.

Способ 2. Проверим предложенные ответы:

- 1) $F = \neg X \wedge \neg Y \wedge Z = 0$ при $X=0, Y=0, Z=0$, что не соответствует первой строке таблицы.
- 2) $F = \neg X \vee \neg Y \vee Z = 1$ при $X=0, Y=0, Z=1$, что не соответствует второй строке таблицы.
- 3) Выражение $X \vee Y \vee \neg Z$ соответствует F при всех предложенных комбинациях X, Y, Z.
- 4) $F = X \vee Y \vee Z = 1$ при $X=0, Y=0, Z=1$, что не соответствует второй строке таблицы.

Таким образом, верный вариант ответа №3.

Ответ: 3

Задание 4. (Задание A9 демоверсий 2005 г., 2006 г.)

Для какого числа X истинно высказывание

$$X > 1 \wedge ((X < 5) \rightarrow (X < 3)) \quad (15)$$

1) 1	2) 2	3) 3	4) 4
------	------	------	------

Решение.

Заменим импликацию, входящую в исходное выражение, воспользовавшись тождеством (1):

(1)

$$(X > 5) \rightarrow (X < 3) = \neg (X < 5) \vee (X < 3)$$

Подставим получившееся выражение в (15):

$$\begin{aligned} (X > 1) \wedge ((X < 5) \rightarrow (X < 3)) &= (X > 1) \wedge (\neg (X < 5) \vee (X < 3)) = \\ &= (X > 1) \wedge ((X \geq 5) \vee (X < 3)) \end{aligned} \quad (16)$$

Найдем значение выражения (16) при заданных значениях X (=1;2;3;4)

$$X=1: (1 > 1) \wedge ((1 \geq 5) \vee (1 < 3)) = 0 \wedge (1 \vee 1) = 0 \wedge 1 = 0$$

$$X=2: (2 > 1) \wedge ((2 \geq 5) \vee (2 < 3)) = 1 \wedge (0 \vee 1) = 1 \wedge 1 = 1$$

$$X=3: (3>1) \wedge ((3>=5) \vee (3<3)) = 1 \wedge (0 \vee 0) = 1 \wedge 0 = 0$$

$$X=4: (4>1) \wedge ((4>=5) \vee (4<3)) = 1 \wedge (0 \vee 0) = 1 \wedge 0 = 0$$

Верный вариант ответа №2.

Ответ: 2.

Задание 5. (Задание A10 демоверсий 2005 г., 2006 г.)

Укажите, какое логическое выражение равносильно выражению $\neg(\neg A \wedge B)$

1) $A \vee \neg B$	2) $\neg A \vee B$	3) $B \wedge \neg A$	4) $A \wedge \neg B$
--------------------	--------------------	----------------------	----------------------

Решение.

Воспользуемся равенствами (6) и (12):

$$(6) \qquad (12)$$

$$\neg(\neg A \wedge B) = \neg(\neg A) \vee \neg B = A \vee \neg B$$

Верный вариант ответа №1.

Ответ: 1.

Задание 6. (Задание A13 демоверсий 2005 г., 2006 г.)

Для 5 букв латинского алфавита заданы их двоичные коды (для некоторых букв – из двух бит, для некоторых – из трех). Эти коды представлены в таблице:

A	B	C	D	E
000	01	100	10	011

Определить, какой набор букв закодирован двоичной строкой 0110100011000

1) EBCEA	2) BDDEA	3) BDCEA	4) EBAEA
----------	----------	----------	----------

Решение.

Заметим, что строка 0110100011000 может начинаться только с двух букв: 01(B) или 011(E). При этом, если первая буква B, то для второй буквы имеется две возможности: 10(D) и 101(-) – нет соответствующей буквы (см. Схему 1) и т.д.

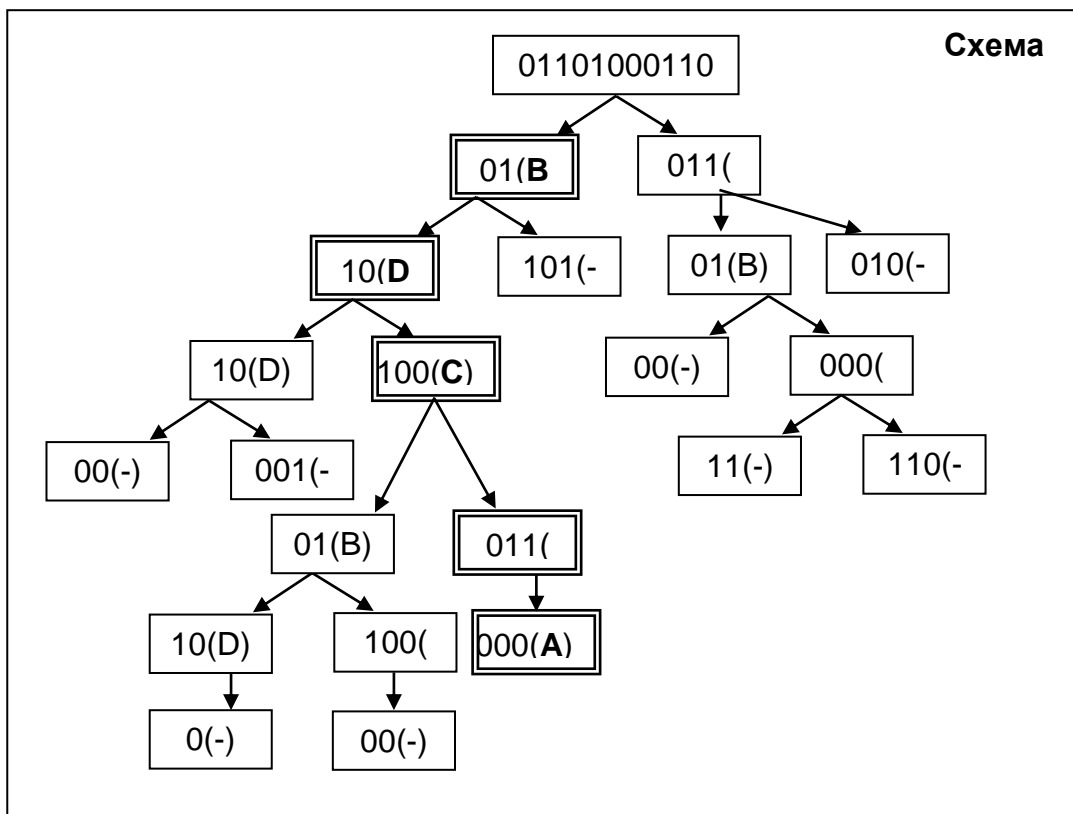
При этом результативным является только одна ветвь дерева (на Схеме 1

она выделена двойной рамкой) – BDCEA, что соответствует варианту ответа №3.

Верный вариант ответа №3.

Ответ: 3.

Задание 7. (Задание A14 демоверсий 2005 г., 2006 г.)



Для составления цепочек используются бусины, помеченные буквами: А, В, С, D, Е. На первом месте в цепочке стоит одна из бусин А, С, Е. На втором – любая гласная, если первая буква согласная, и любая согласная, если первая гласная. На третьем месте – одна из бусин С, D, Е, не стоящая в цепочке на первом месте. Какая из перечисленных цепочек создана по правилу?

- | | | | |
|--------|--------|--------|--------|
| 1) CBE | 2) ADD | 3) ECE | 4) EAD |
|--------|--------|--------|--------|

Решение.

Введем обозначения для условий:

Условие 1 = «На первом месте в цепочке стоит одна из бусин А, С, Е».

Условие 2 = «На втором – любая гласная, если первая буква согласная, и любая согласная, если первая гласная».

Условие 3 = «На третьем месте – одна из бусин C, D, E, не стоящая в цепочке на первом месте».

Рассмотрим выполнимость **Условий 1-3** для вариантов ответов 1) - 4). Поставим символ «1», если соответствующее условие выполнено, «0» - если условие не выполнено (см. **Таблицу 11**).

Таблица 11. Выполнимость условий 1-3 для вариантов ответов Задания 7.

№ ответа	Вариант ответа	Условие 1	Условие 2	Условие 3
1	CBE	1	0	1
2	ADD	1	1	1
3	ECE	1	1	0
4	EAD	1	0	1

Из **таблицы 6** видно, что все три условия выполнены только для варианта ответа №2.

Ответ: 2.

Задание 8. (Задание В2 демоверсий 2005 г., 2006 г.)

Сколько различных решений имеет уравнение

$$(K \wedge L \wedge M) \vee (\neg L \wedge \neg M \wedge N) = 1$$

где K, L, M, N – логические переменные?

В ответе **не нужно** перечислять все различные наборы K, L, M и N, при которых выполнено данное равенство. В качестве ответа вам нужно указать только количество таких наборов.

Решение.

Заметим, что поскольку исходное уравнение представляет собой объединение двух логических выражений, то оно равносильно совокупности (объединению) уравнений, состоящих из этих выражений:

$$\left[\begin{array}{l} K \wedge L \wedge M = 1 \\ \neg L \wedge \neg M \wedge N = 1 \end{array} \right. \quad \begin{array}{l} (17) \\ (18) \end{array}$$

При этом уравнение (17) представляет собой пересечение трех

логических выражений, и потому оно принимает значение, равное 1, тогда и только тогда, когда каждое из них истинно, т.е. $K=L=M=1$. На выражение N условий не накладывается, поэтому возможны два варианта решений: 1) $K=L=M=1, N=1$; 2) $K=L=M=1, N=0$.

Уравнение (18) также представляет собой пересечение трех логических выражений, и потому оно принимает значение, равное 1, тогда и только тогда, когда $\neg L = \neg M = N = 1$. Откуда: $L=M=0, N=1$. На выражение K условий не накладывается, поэтому у уравнения (18) – также два решения: 1) $K=M=0, N=1, K=1$; 2) $K=M=0, N=1, k=0$.

Таким образом, уравнение (17) имеет 2 решения и уравнение (18) имеет два решения. Поскольку исходное уравнение представляет собой объединение этих двух уравнений, то количество его решений равно сумме решений уравнений (17) и (18), т.е. равно 4.

Ответ: 4.

Задание 9. (Задание В2 демоверсии 2004 г.)

Укажите значения переменных K, L, M, N , при которых логическое выражение

$$(\neg K \vee M) \rightarrow (\neg L \vee M \vee N)$$

ложно. Ответ запишите в виде строки из четырех символов: значений переменных K, L, M и N (в указанном порядке). Так, например, строка 1101 соответствует тому, что $K=1, L=1, M=0, N=1$.

Решение.

Преобразуем данное выражение, используя равенства (1), (6), (12):

$$(1) \qquad (6), (12)$$

$$(\neg K \vee M) \rightarrow (\neg L \vee M \vee N) = \neg (\neg K \vee M) \vee (\neg L \vee M \vee N) = (K \wedge \neg M) \vee (\neg L \vee M \vee N) = 0$$

Поскольку получившееся выражение представляет собой логическое сложение двух выражений $(K \wedge \neg M)$ и $(\neg L \vee M \vee N)$, то оно равно тогда и только тогда, когда

$$K \wedge \neg M = 0 \quad (19)$$

$$\neg L \vee M \vee N = 0 \quad (20)$$

Из (20) следует, что $\neg L = M = N = 0$, значит $L=1, M=0, N=0$.

Подставим $M=0$ в уравнение (19):

$$K \wedge 1 = 0, \text{ откуда } K=0.$$

В итоге получим: $K=0, L=1, M=0, N=0$.

Ответ: 0100.

Задание 10. (Задание В4 демоверсии 2006 г.)

Мама, прибежавшая на звон разбившейся вазы, застала всех трех своих сыновей в совершенно невинных позах: Саша, Ваня и Коля делали вид, что происшедшее к ним не относится. Однако футбольный мяч среди осколков явно говорил об обратном.

- Кто это сделал? - спросила мама.

- Коля не бил по мячу, - сказал Саша. - Это сделал Ваня.

Ваня ответил: - Разбил Коля, Саша не играл в футбол дома.

- Так я и знала, что вы друг на дружку сваливать будете, - рассердилась мама. -

Ну, а ты что скажешь? - спросила она Колю.

- Не сердись, мамочка! Я знаю, что Ваня не мог этого сделать. А я сегодня еще не сделал уроки, - сказал Коля.

Оказалось, что один из мальчиков оба раза солгал, а двое в каждом из своих заявлений говорили правду.

Кто разбил вазу?

Решение.

Введем обозначения для высказываний:

$$A = \text{«Коля не бил по мячу»} = \text{«}\neg\text{Коля»} = \neg C;$$

$$B = \text{«Это сделал Ваня»} = \text{«Ваня»};$$

$$C = \text{«Разбил Коля»} = \text{«Коля»};$$

$$D = \text{«Саша не играл в футбол дома»} = \text{«}\neg\text{Саша»};$$

$$E = \text{«Ваня не мог этого сделать»} = \text{«}\neg\text{Ваня»} = \neg B;$$

$$F = \text{«Я сегодня еще не сделал уроки»} - \text{не имеет отношения к вопросу}$$

«Кто разбил вазу?».

Из условия задачи известно, что один из мальчиков оба раза солгал, а двое в каждом из своих заявлений говорили правду.

Предположим, что солгал первый мальчик, тогда:

$$A=0 \wedge B=0 \wedge C=1 \wedge D=1 \wedge E=1 \wedge F=1.$$

Поскольку $A = \neg C$ и $E = \neg B$, имеем:

$\neg C=0$, $B=0$, $C=1$, $D=1$, $\neg B=1$, $F=1$, – противоречий не получили, этот вариант является решением задачи: $\neg B=1$, $C=1$, $F=1$, осталось лишь вспомнить обозначения:

$B=$ «Ваня», значит: $\neg B =$ не «Ваня»;

$C=$ «Коля»;

$F=1$ – не имеет отношения к вопросу. Значит, вазу разбил Коля.

На всякий случай рассмотрим два других варианта (когда солгал второй или третий мальчики).

Если солгал второй мальчик, то:

$$C=0 \wedge D=0 \wedge A=1 \wedge B=1 \wedge E=1 \wedge F=1.$$

Поскольку $A = \neg C$ и $E = \neg B$, имеем:

$C=0$, $D=0$, $\neg C=1$, $B=1$, $\neg B=1$, $F=1$ – получили противоречие: $B=1$ и $\neg B=1$, значит, этот вариант нам не подойдет.

Если солгал третий мальчик, то:

$$E=0 \wedge F=0 \wedge A=1 \wedge B=1 \wedge C=1 \wedge D=1.$$

Заменим: $A = \neg C$ и $E = \neg B$, тогда:

$E=0$, $F=0$, $\neg C=1$, $B=1$, $C=1$, $D=1$ – получили противоречие: $C=1$ и $\neg C=1$, значит, этот вариант нам не подойдет.

Ответ: Коля.

Задание 11. (Задание В8 демоверсии 2006 г.)

В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке возрастания количества страниц, которые найдет поисковый сервер по каждому запросу.

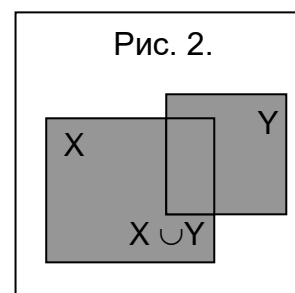
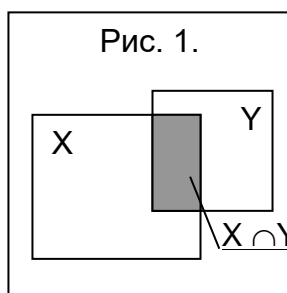
Для обозначения логической операции “ИЛИ” в запросе используется

символ $|$, а для логической операции “И” - $\&$.

А	чемпионы (бег & плавание)
Б	чемпионы & плавание
В	чемпионы бег плавание
Г	чемпионы & Европа & бег & плавание

Решение.

Решение задачи сводится к тому, чтобы расположить множества, состоящие из результатов поиска А-Г в порядке возрастания количества элементов. Воспользуемся тем, что



логическое умножение для двух множеств равносильно их пересечению, а логическое сложение – их объединению. При этом при пересечении несовпадающих множеств в результате всегда получается множество, меньшее, чем исходные множества, а при объединении – большее, чем исходные (см. Рис. 1 – Рис. 2).

Введем обозначения для множеств и запросов.

Пусть $K =$ тогда запрос $A = K \cup L \cap M;$

«чемпионы»;

$$B = K \cap M;$$

$$L = \text{«бег»};$$

$$V = K \cup L \cup M;$$

$$M = \text{«плавание»};$$

$$Г = K \cap N \cap L \cap M.$$

$$N = \text{«Европа»},$$

Из обозначений запросов видно, что самым маленьким по количеству элементов будет множество Г (состоит из пересечений четырех множеств К, L, M, N). Самым большим множеством является множество В, т.к. оно состоит из объединений трех множеств К, L и M. Значит, ответ на **Задание 11** будет выглядеть так: Г**В. Осталось определить, какие множества (из А и Б) будут стоять на 2 и 3 местах.

Заметим, что множество В состоит из пересечений двух множеств К и М,

поэтому оно является множеством, меньшим К). Множество А состоит из объединения множества К с пересечением множеств L и M, поэтому А - множество, большее К. Значит, при расположении их в порядке возрастания, получим, что на втором месте в ответе будет стоять Б, а на третьем – А.

Ответ: ГБАВ.

Контрольные вопросы

1. Какие основные логические операции вам известны?
2. Перечислите основные равносильности алгебры логики.
3. Постройте таблицы истинности для основных логических операций.

Задания для самостоятельной работы

Задание 1.

Для заданного логического выражения (высказывания):

- 1) построить таблицу истинности;
- 2) упростить высказывание, используя равносильные преобразования;
- 1) полученный результат проверить, построив для него таблицу истинности.

Вариант		Вариант	
1.	$(A \leftrightarrow B) \vee A\bar{B} \vee C$	2.	$(A \rightarrow B) \vee A\bar{C} \vee BC$
3.	$(AC \rightarrow B) \vee A\bar{C}$	4.	$A\bar{B} \vee (A \leftrightarrow C)B$
5.	$(\bar{A} \rightarrow B)(A\bar{C} \vee BC)$	6.	$(A \sim C) \vee A\bar{B} \vee AC$
7.	$(A \rightarrow C) \vee A\bar{B} \vee BC$	8.	$(C \sim B) \vee A\bar{C} \vee BC$
9.	$(BC \rightarrow A) \vee A\bar{C}$	10.	$(AB \rightarrow C) \vee A\bar{C}$
11.	$(\bar{A} \rightarrow C)(B\bar{C} \vee AB)$	12.	$(\bar{A} \leftrightarrow B)(A \rightarrow BC)$
13.	$(B \rightarrow C) \vee A\bar{B} \vee \bar{A}C$	14.	$(A \rightarrow \bar{B}C) \vee A\bar{B} \vee BC$
15.	$(AC \rightarrow \bar{B}) \vee B\bar{C}$	16.	$B \vee (A \leftrightarrow CB) \vee A\bar{C}$
17.	$(AC \rightarrow B) \vee A\bar{B}\bar{C}$	18.	$(A \sim C)(B\bar{C} \rightarrow AB)$
19.	$(B \rightarrow C) \vee (B \rightarrow AC)$	20.	$(AB \rightarrow C) \vee A \vee \bar{A}C$

21.	$(A \leftrightarrow C) \vee (A\bar{B} \rightarrow C)$	22.	$\bar{A}B \rightarrow \bar{C} \vee ABC$
23.	$(AB \rightarrow C) \vee A\bar{C}$	24.	$\bar{A} \rightarrow BC)(A \sim C)$
25.	$(\bar{A} \leftrightarrow B)(A \rightarrow BC)$	26.	$(\bar{A} \rightarrow \bar{B})(C\bar{A} \rightarrow \bar{B})$

Задание 2.

Построить таблицы истинности для формул:

- $(\bar{X} \leftrightarrow Y) \rightarrow \bar{Y}$;
- $(X \vee Y) \leftrightarrow (\bar{X} \rightarrow Y)$;
- $(\bar{X} \vee \bar{Y}) \rightarrow X \wedge Y$;
- $(X \wedge \bar{Y}) \rightarrow (\bar{X} \leftrightarrow Y)$.

Задание 3.

Для какого имени истинно высказывание:

Первая буква имени согласная \wedge (\neg Вторая буква имени согласная \rightarrow Четвертая буква имени гласная):

1) ИВАН	2) ПЕТР	3) ПАВЕЛ	4) ЕЛЕНА
---------	---------	----------	----------

Задание 4.

Какое логическое выражение равносильно $\neg(\neg X \wedge \neg Y)$?

Задание 5.

Какое логическое выражение равносильно $(\neg X \vee Y) \wedge X$?

Задание 6.

Какое логическое выражение F(A,B) соответствует таблице истинности:

A	B	F
0	0	1
0	1	1

1	0	0
1	1	1

Задание 7.

В понедельник в одном из классов должно быть проведено 4 урока – по математике, физике, информатике и биологии. Учителя высказывали свои пожелания для составления расписания. Учитель математики хотел бы проводить первый или второй урок, учитель физики – второй или третий, учитель информатики не второй и не третий, учитель биологии – третий и четвертый. Какой вариант расписания устроит всех учителей? (Обозначения: М – математика, Ф – физика, И – информатика, Б – биология)

1) ИМБФ	2) МФБИ	3) МИФБ	4) МБФИ
---------	---------	---------	---------

Задание 8.

X, Y, Z – целые числа, для которых истинно высказывание

$$(Z < X \vee Z < Y) \wedge (\neg Z + 1 < X) \wedge \neg(Z + 1 < Y)$$

Чему равно Z, если X=20, Y=10?

Задание 9.

Три свидетеля дорожного происшествия сообщили сведения о скрывшемся нарушителе. Боб утверждает, что тот был на синем «Рено». Джон сказал, что нарушитель ехал на черной «Тойоте», а Сэм сказал, что машина была точно не синяя, и, по всей видимости, это был «Форд». Когда удалось отыскать машину, выяснилось, что каждый из свидетелей точно определил только один из параметров автомобиля, а в другом ошибся. Машина какой марки и какого цвета была у нарушителя?

Задание 10.

В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке возрастания количества страниц, которые найдет

поисковый сервер по каждому запросу.

Для обозначения логической операции “ИЛИ” в запросе используется символ |, а для логической операции “И” - &.

А	законы & физика
Б	законы (физика & биология)
В	законы & физика & биология & химия
Г	законы физика биология

Лабораторная работа №5. Базовые логические элементы

Цели: сформировать представление об устройствах элементной базы компьютера и навыки построения логических схем.

Теоретический материал

Логические схемы

В основе обработки компьютером информации лежит алгебра логики, разработанная английским математиком Дж. Булем.

Базовые логические элементы реализуют основные логические операции:

- логический элемент «И» — логическое умножение;
- логический элемент «ИЛИ» — логическое сложение;
- логический элемент «НЕ» — инверсию.

Поскольку любая логическая операция может быть представлена в виде комбинации трех основных, любые устройства компьютера, производящие обработку или хранение информации, могут быть собраны из базовых логических элементов, как из «кирпичиков».

Логические элементы компьютера оперируют с сигналами, представляющими собой электрические импульсы. Есть импульс — логический смысл сигнала — 1, нет импульса — 0. На входы логического элемента поступают сигналы-значения аргументов, на выходе появляется сигнал-

значение функции.

Преобразование сигнала логическим элементом задается таблицей состояния, которая фактически является таблицей истинности, соответствующей логической функции.

Логический элемент «НЕ». На вход A логического элемента (рис. 1) подается сигнал 0 или 1. На выходе получается сигнал 0 или 1 в соответствии с таблицей истинности инверсии.

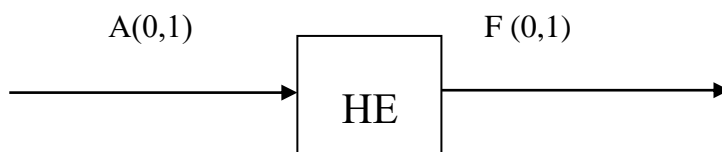


Рисунок 14. Логический элемент «НЕ»

Логический элемент «И». На входы A и B логического элемента (рис. 2) подаются два сигнала (00, 01, 10 или 11). На выходе получается сигнал 0 или 1 в соответствии с таблицей истинности операции логического умножения.

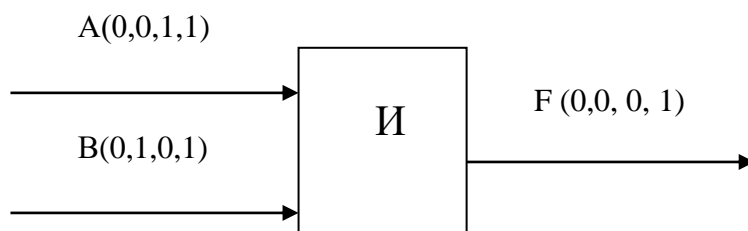


Рисунок 15. Логический элемент «И»

Логический элемент «ИЛИ». На входы A и B логического элемента (рис. 3) подаются два сигнала (00, 01, 10 или 11). На выходе получается сигнал 0 или 1 в соответствии с таблицей истинности операции логического сложения.

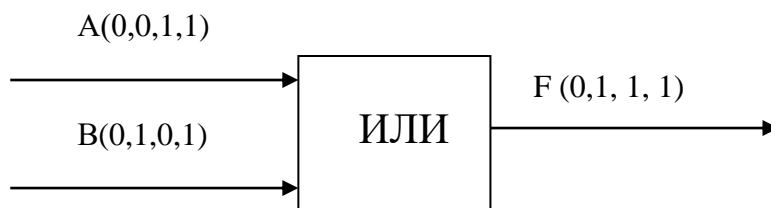


Рисунок 16. Логический элемент «ИЛИ»

Построение логических схем

Сигнал, выработанный одним логическим элементом, можно подавать на вход другого элемента, это дает возможность образовывать цепочки из отдельных логических элементов — *функциональные схемы*.

Функциональная (логическая) схема – это схема, состоящая из логических элементов, которая выполняет определённую функцию. Анализируя функциональную схему, можно понять, как работает логическое устройство, т.е. дать ответ на вопрос: какую функцию она выполняет.

Правило построения логических схем:

1. Определить число логических переменных;
2. Определить количество базовых логических операций и их порядок;
3. Изобразить для каждой логической операции соответствующий ей логический элемент;
4. Соединить логические элементы в порядке выполнения логических операций.

Упражнение 1. Пусть $X = \text{истина}$, $Y = \text{ложь}$. Составить логическую схему для следующего логического выражения: $F = X \vee Y \wedge X$.

Решение.

1. Две переменные: X и Y .
2. Две логические операции: \vee и \wedge .
3. Строим схему:

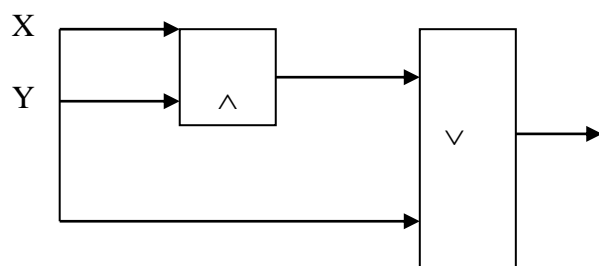


Рисунок 17

Ответ: $1 \vee 0 \wedge 1 = 1$.

Упражнение 2. Постройте логическую схему, соответствующую логическому выражению $F = X \wedge Y \vee \overline{(Y \vee X)}$. Вычислить значения выражения

для $X=1, Y=0$.

Решение.

1. Переменных две: X и Y .
2. Логических операций три: конъюнкция и две дизъюнкции.
3. Схему строим направо в соответствии с порядком логических операций:

операций:

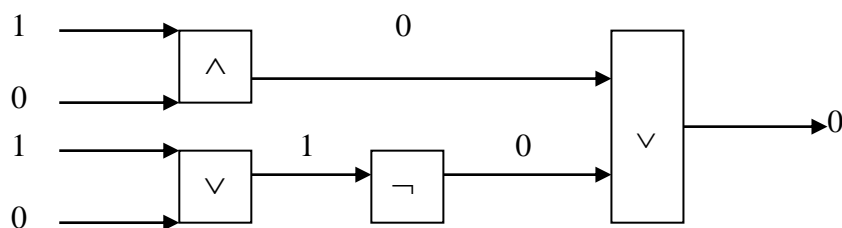


Рисунок 18

4. Вычислим значение выражения: $F = 1 \wedge 0 \vee \overline{(0 \vee 1)} = 0$.

Рассмотрим, как из логических элементов можно сконструировать устройство для сложения двух двоичных чисел — так называемый одноразрядный сумматор или полусумматор. Это устройство должно давать на выходе следующие сигналы:

- $0 + 0 = 0 0$
- $0 + 1 = 0 1$
- $1 + 0 = 0 1$
- $1 + 1 = 1 0$

Составим таблицу истинности для этого сумматора, обозначив слагаемые X и Y , а результаты P (перенос в старший разряд) и Z (сумма):

X	Y	P	Z
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Из этой таблицы сразу видно, что перенос можно реализовать с помощью операции логического умножения: $P = A \wedge B$.

Получим теперь формулу для вычисления суммы. Как видим, значение суммы есть значение отрицание эквивалентности: $S = \overline{A \leftrightarrow B}$. Упростим

последнее выражение: $\overline{A \leftrightarrow B} = \overline{(A \rightarrow B) \wedge (B \rightarrow A)} = \overline{(\overline{A} \vee B) \wedge (\overline{B} \vee A)} =$

$$= (A \wedge \overline{B}) \vee (B \wedge \overline{A}) = (A \vee B) \wedge (A \vee \overline{A}) \wedge (\overline{B} \vee B) \wedge (\overline{B} \vee \overline{A}) = (A \vee B) \wedge 1 \wedge 1 \wedge (\overline{B} \vee \overline{A}) =$$

$$= (A \vee B) \wedge (\overline{A} \vee \overline{B}) = (A \vee B) \wedge \overline{(A \wedge B)}$$

Таким образом, для определения суммы можно применить следующее логическое выражение: $S = (A \vee B) \wedge \overline{(A \wedge B)}$.

Построим таблицу истинности для данного логического выражения и убедимся в правильности наших действий:

A	B	$A \vee B$	$A \wedge B$	$\overline{A \wedge B}$	$(A \vee B) \wedge \overline{(A \wedge B)}$
0	0	0	0	1	0
0	1	1	0	1	1
1	0	1	0	1	0
1	1	1	1	0	1

Теперь на основе полученных логических выражений из базовых логических элементов можно построить схему сложения одноразрядных двоичных чисел.

По логической формуле переноса легко определить, что для получения переноса необходимо использовать логический элемент «И».

Анализ логической формулы для суммы показывает, что на выходе должен стоять элемент логического умножения «И», который имеет два входа. На один из входов надо подать результат логического сложения исходных величин A и B, то есть на него должен подаваться сигнал с элемента логического сложения «ИЛИ».

На второй вход требуется подать результат инвертированного логического умножения исходных сигналов ($\overline{A \wedge B}$), то есть на второй вход должен

подаваться сигнал с элемента «НЕ», на вход которого должен поступать сигнал с элемента логического умножения «И» (рис.19).

Рисунок 19. Полусумматор двоичных чисел

III. Выполнение практических заданий

Упражнение 3

Постройте логическую схему, соответствующую логическому выражению, и найдите значение этого выражения:

а) $F = A \vee B \wedge \bar{C}$, если $A = 1, B = 1, C = 1$ (ответ: 1)

б) $F = \overline{(A \vee B \wedge C)}$, если $A = 0, B = 1, C = 1$ (ответ: 1)

Упражнение 4.

Постройте логическое выражение по логической схеме:

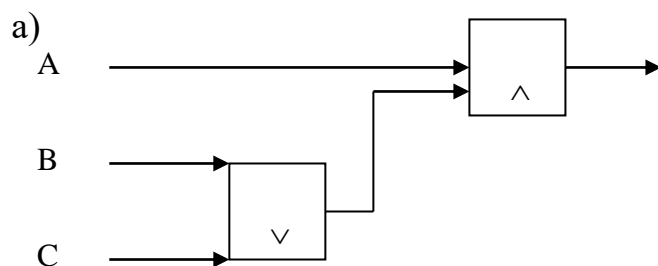


Рисунок 20

Ответ: $F = A \wedge (B \vee C)$.

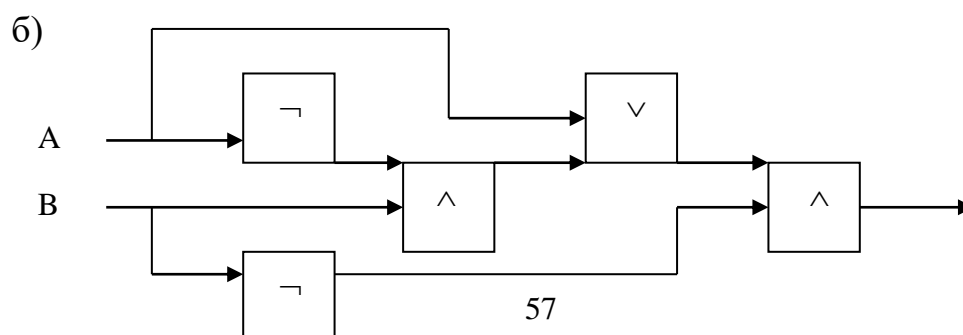


Рисунок 21

Ответ: $F = \bar{B} \wedge ((\bar{A} \wedge B) \vee A)$.

IV. Подведение итогов

Сегодня на уроке мы изучили базовые логические элементы, научились составлять логические схемы. Строили логическую схему полусумматора. Увидели взаимосвязь компьютера и алгебры логики. Кроме полусумматора существуют другие логические устройства компьютера, такие как сумматор, триггер, регистры.

V. Задания для самостоятельного выполнения

Задание 1

Постройте логическую схему, соответствующую логическому выражению, и найдите

значение этого выражения:

а) $F = \bar{A} \vee B \wedge C$, если $A = 1, B = 0, C = 1$;

б) $F = (A \vee B) \wedge (C \vee B)$, если $A = 0, B = 1, C = 0$.

Задание 2

Постройте логическое выражение по логической схеме:

а)

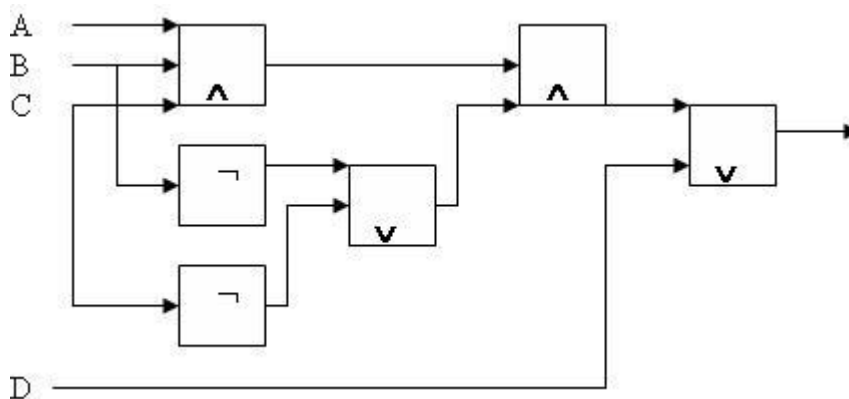


Рисунок 22

б)

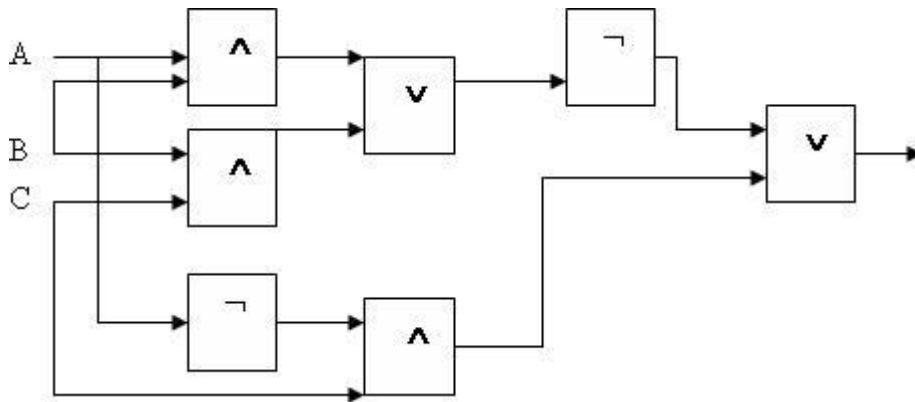
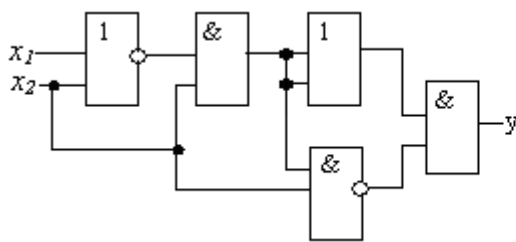


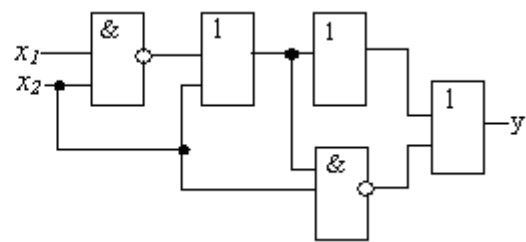
Рисунок 23

Задание 3.

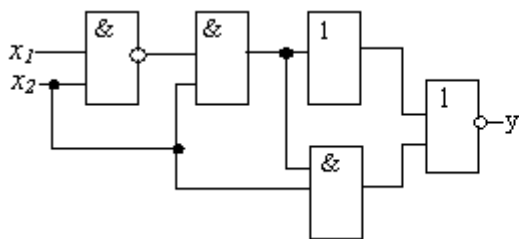
Для заданной комбинационной схемы постройте аналитическое выражение (формулу).



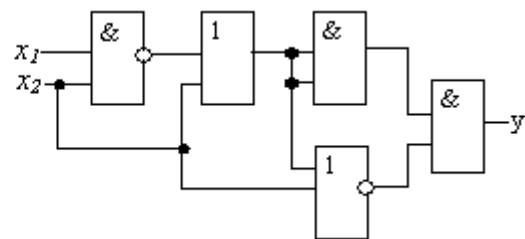
№1



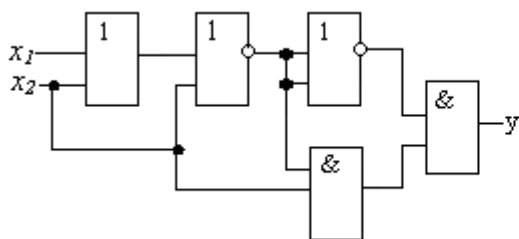
№2



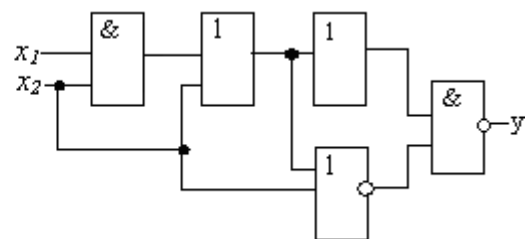
№3



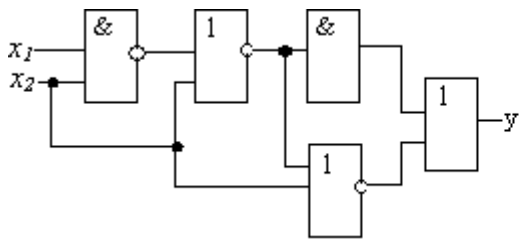
№4



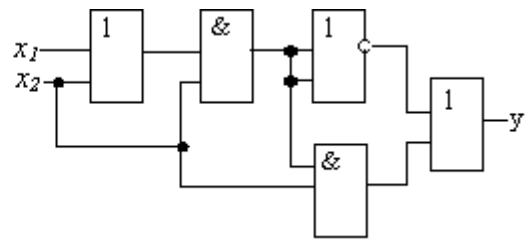
№5



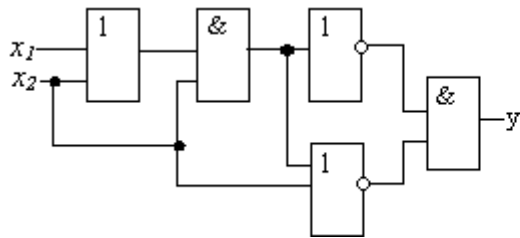
№6



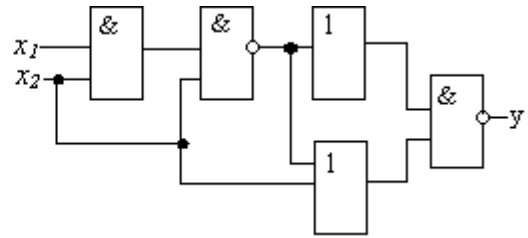
№7



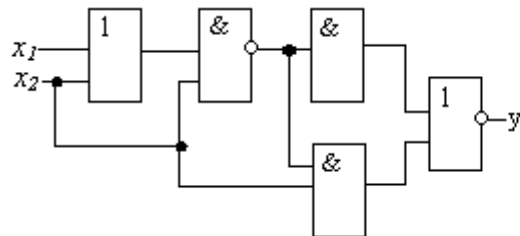
№8



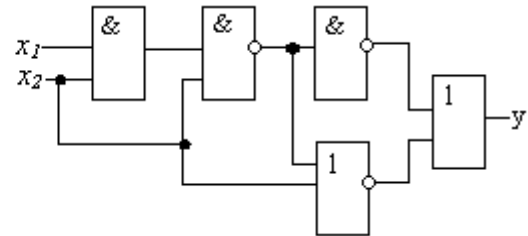
№9



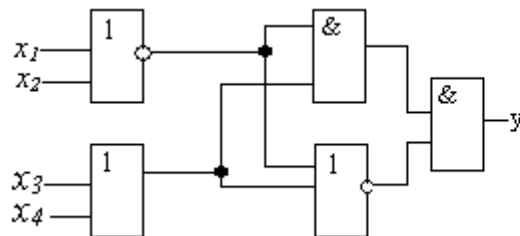
№10



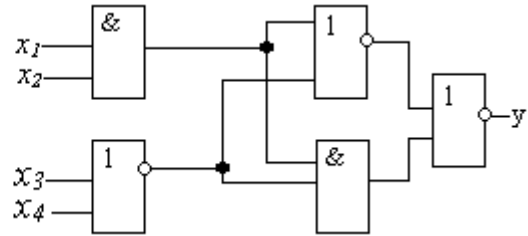
№11



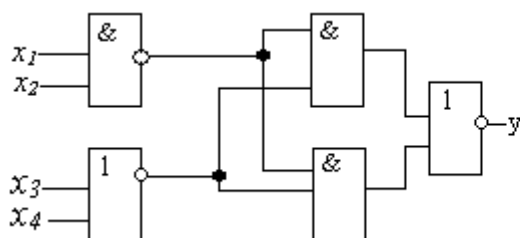
№12



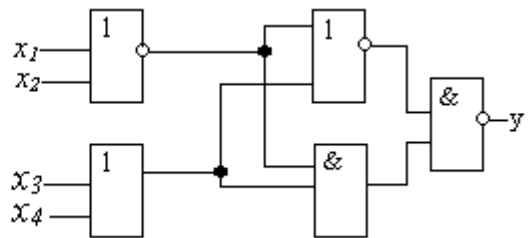
№13



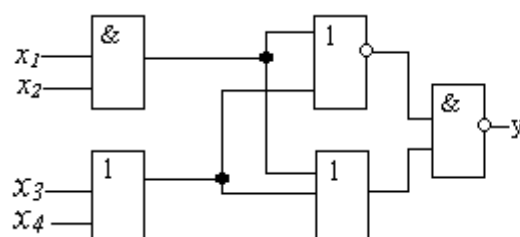
№14



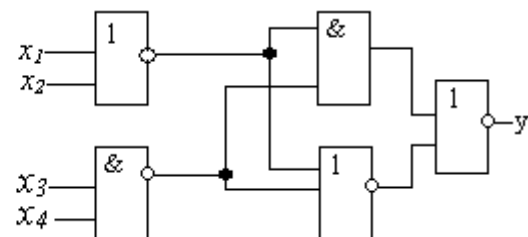
№15



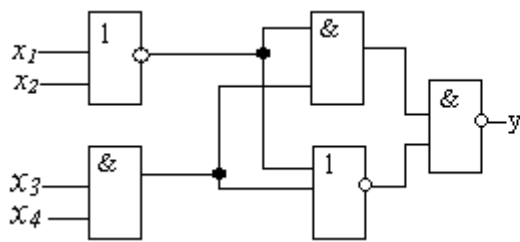
№16



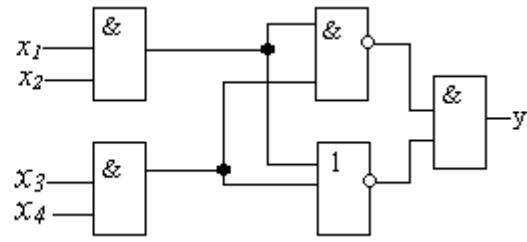
№17



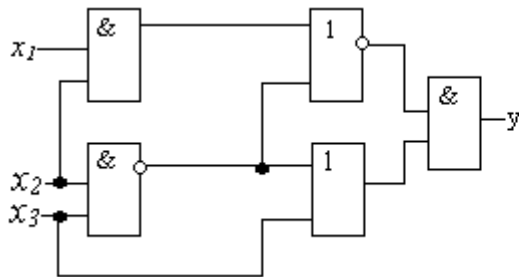
№18



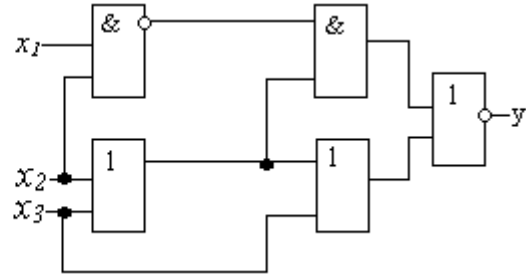
№19



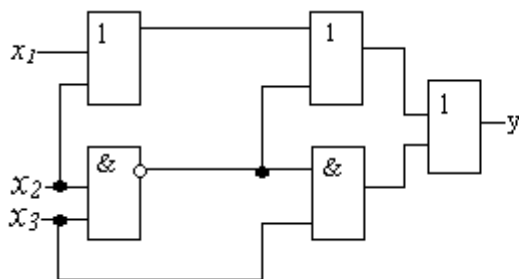
№20



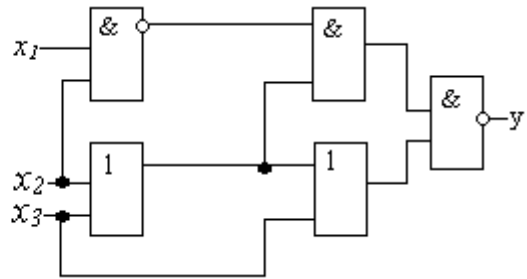
№21



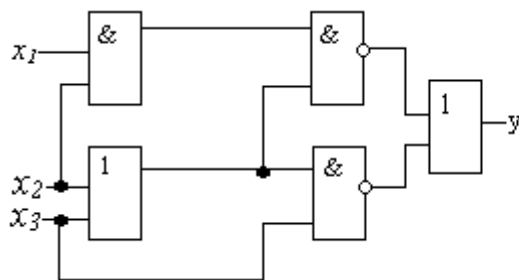
№22



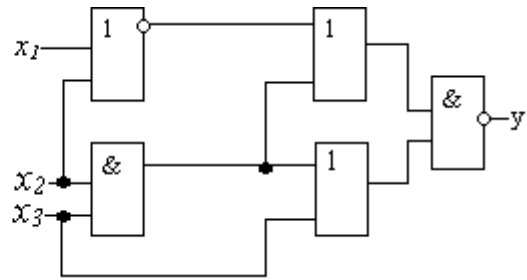
№23



№24



№25



№26

Раздел 3. Технические средства реализации информационных процессов

Тема 4. Конструкция ПК

Лабораторная работа №6. Основы организации ЭВМ

1. Архитектура компьютера.



Рисунок 24. Архитектура ЭВМ



Презентация 2. Устройства компьютера

С самых первых поколений, любая ЭВМ состоит из следующих основных устройств: **процессор**, **память** (внутренняя и внешняя) и **устройства ввода и вывода** информации. Рассмотрим более подробно назначение каждого из них.

Процессор является главным устройством компьютера, в котором собственно и происходит обработка всех видов информации. Другой важной функцией процессора является обеспечение согласованного действия всех узлов, входящих в состав компьютера. Соответственно наиболее важными частями процессора являются **арифметико-логическое устройство АЛУ** и **устройство управления УУ**.



Рисунок 25



Рисунок 26 Архитектура и структура компьютера

Каждый процессор способен выполнять вполне определенный набор универсальных инструкций, называемых чаще всего **машинными командами**. Каков именно этот набор, определяется устройством конкретного процессора, но он не очень велик и в основном аналогичен для различных процессоров. Работа ЭВМ состоит в выполнении последовательности таких команд, подготовленных в виде программы. Процессор способен организовать считывание очередной команды, ее анализ и выполнение, а также при необходимости принять данные или отправить результаты их обработки на требуемое устройство. Выбрать, какую инструкцию программы исполнять следующей, также должен сам процессор, причем результат этого выбора часто может зависеть от обрабатываемой в данный момент информации.

Хотя внутри процессора всегда имеются специальные ячейки (регистры) для оперативного хранения обрабатываемых данных и некоторой служебной информации, в нем сознательно не предусмотрено место для хранения программы. Для этой важной цели в компьютере служит другое устройство – **память**. Мы рассмотрим лишь наиболее важные виды компьютерной памяти, поскольку ее ассортимент непрерывно расширяется и пополняется все новыми и новыми типами.



Рисунок 27 Первое знакомство с компьютером

Память в целом предназначена для хранения как *данных*, так и *программ* их обработки: согласно фундаментальному принципу фон Неймана, для обоих типов информации используется единое устройство.

Начиная с самых первых ЭВМ, память сразу стали делить на **внутреннюю** и **внешнюю**. Исторически это действительно было связано с размещением внутри или вне процессорного шкафа. Однако с уменьшением размеров машин внутрь основного процессорного корпуса удавалось поместить все большее количество устройств, и первоначальный непосредственный смысл данного деления постепенно утратился. Тем не менее, терминология сохранилась.

Под **внутренней памятью** современного компьютера принято понимать быстродействующую электронную память, расположенную на его системной плате. Сейчас такая память изготавливается на базе самых современных полупроводниковых технологий (раньше использовались магнитные

устройства на основе ферритовых сердечников – лишнее свидетельство тому, что конкретная физические принципы значения не имеют). Наиболее существенная часть внутренней памяти называется ОЗУ - **оперативное запоминающее устройство**. Его главное назначение состоит в том, чтобы хранить данные и программы для решаемых в текущий момент задач. Наверное, каждому пользователю известно, что при выключении питания содержимое ОЗУ полностью теряется. В состав внутренней памяти современного компьютера помимо ОЗУ также входят и некоторые другие разновидности памяти, которые при первом знакомстве можно пропустить. Здесь упомянем только о **постоянном запоминающем устройстве (ПЗУ)**, в котором в частности хранится информация, необходимая для первоначальной загрузки компьютера в момент включения питания. Как очевидно из названия, информация в ПЗУ не зависит от состояния компьютера (для лучшего понимания можно указать на некоторую аналогию между информацией в ПЗУ и “врожденными” безусловными рефлексами у живых существ). Раньше содержимое ПЗУ раз и навсегда формировалось на заводе, теперь же современные технологии позволяют в случае необходимости обновлять его даже не извлекая из компьютерной платы.

Внешняя память реализуется в виде довольно разнообразных устройств хранения информации и обычно конструктивно оформляется в виде самостоятельных блоков. Сюда, прежде всего, следует отнести накопители на гибких и жестких магнитных дисках (последние несколько жаргонно пользователи часто именуют винчестерами), а также оптические дисководы (устройства для работы с CD ROM). В конструкции устройств внешней памяти имеются механически движущиеся части, поэтому скорость их работы существенно ниже, чем у полностью электронной внутренней памяти. Тем не менее, внешняя память позволяет сохранить огромные объемы информации с целью последующего использования. Подчеркнем, что информация во внешней памяти прежде всего предназначена для самого компьютера и поэтому хранится в удобной *ему* форме; человек без использования машины не в

состоянии, например, даже отдаленно представить содержимое немаркированной дискеты или диска CD ROM.

Современные программные системы способны объединять внутреннюю и внешнюю память *в единое целое*, причем так, чтобы наиболее редко используемая информация попадала в более медленно работающую внешнюю память. Такой метод дает возможность очень существенно расширить объем обрабатываемой с помощью компьютера информации.

Если процессор дополнить памятью, то такая система уже может быть работоспособной. Ее существенным недостатком является невозможность узнать что-либо о происходящем внутри такой системы. Для получения информации о результатах, необходимо дополнить компьютер **устройствами вывода**, которые позволяют представить их в доступной человеческому восприятию форме. Наиболее распространенным устройством вывода является дисплей, способный быстро и оперативно отображать на своем экране как текстовую, так и графическую информацию. Для того чтобы получить копию результатов на бумаге, используют печатающее устройство, или принтер.

Наконец, поскольку пользователю часто требуется вводить в компьютерную систему новую информацию, необходимы еще и **устройства ввода**. Простейшим устройством ввода является клавиатура. Широкое распространение программ с графическим интерфейсом способствовало популярности другого устройства ввода – манипулятора мышь. Наконец, очень эффективным современным устройством для автоматического ввода информации в компьютер является сканнер, позволяющий не просто преобразовать картинку с листа бумаги в графический компьютерный файл, но и с помощью специального программного обеспечения распознать в прочитанном изображении текст и сохранить его в виде, пригодном для редактирования в обычном текстовом редакторе.

Теперь, когда мы знаем основные устройства компьютера и их функции, осталось выяснить, как они взаимодействуют между собой. Для этого

обратимся к функциональной схеме современного компьютера, приведенной на рисунке.



Рисунок 28 Функциональная схема современного компьютера

Для связи основных устройств компьютера между собой используется специальная информационная магистраль, обычно называемая инженерами **шиной**. Шина состоит из трех частей:

- **шина адреса**, на которой устанавливается адрес требуемой ячейки памяти или устройства, с которым будет происходить обмен информацией;
- **шина данных**, по которой собственно и будет передана необходимая информация; и, наконец,
- **шина управления**, регулирующей этот процесс (например, один из сигналов на этой шине позволяет компьютеру различать между собой адреса памяти и устройств ввода/вывода).

2. Принципы построения компьютеров

В основу построения подавляющего большинства компьютеров положены следующие общие принципы, сформулированные в 1945 г. американским ученым **Джоном фон Нейманом**.



Рисунок 29. Джон фон Нейман, 1945 г.

Изучите и законспектируйте основные принципы фон Неймана.

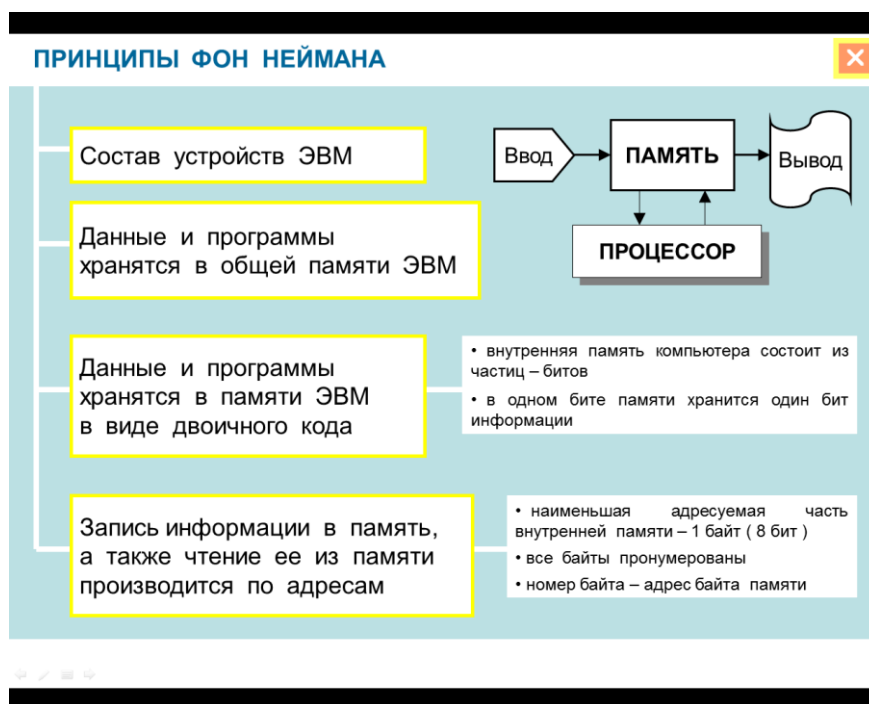


Рисунок 30 Плакат «Принципы фон Неймана»

1. Принцип программного управления. Из него следует, что программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности.

Выборка программы из памяти осуществляется с помощью счетчика команд. Этот регистр процессора последовательно увеличивает хранимый в нем адрес очередной команды на длину команды.

А так как команды программы расположены в памяти друг за другом, то тем самым организуется выборка цепочки команд из последовательно расположенных ячеек памяти.

Если же нужно после выполнения команды перейти не к следующей, а к какой-то другой, используются команды **условного** или **безусловного переходов**, которые заносят в счетчик команд номер ячейки памяти, содержащей следующую команду. Выборка команд из памяти прекращается после достижения и выполнения команды “*stop*”.

Таким образом, **процессор исполняет программу автоматически, без вмешательства человека.**

2. Принцип однородности памяти. Программы и данные хранятся в одной и той же памяти. Поэтому компьютер не различает, что хранится в данной ячейке памяти – число, текст или команда. Над командами можно выполнять такие же действия, как и над данными. Это открывает целый ряд возможностей. Например, **программа в процессе своего выполнения также может подвергаться переработке**, что позволяет задавать в самой программе правила получения некоторых ее частей (так в программе организуется выполнение циклов и подпрограмм). Более того, **команды одной программы могут быть получены как результаты исполнения другой программы.** На этом принципе основаны методы трансляции – перевода текста программы с языка программирования высокого уровня на язык конкретной машины.

3. Принцип адресности. Структурно основная память состоит из перенумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка. Отсюда следует возможность давать имена областям памяти, так, чтобы к запомненным в них значениям можно было впоследствии обращаться или менять их в процессе выполнения программ с использованием присвоенных имен.

Компьютеры, построенные на этих принципах, относятся к типу **фон-неймановских**. Но существуют компьютеры, принципиально отличающиеся от фон-неймановских. Для них, например, может не выполняться принцип

программного управления, т.е. они могут работать без “счетчика команд”, указывающего текущую выполняемую команду программы. Для обращения к какой-либо переменной, хранящейся в памяти, этим компьютерам **не обязательно давать ей имя**. Такие компьютеры называются **не-фон-неймановскими**.

3. Шинная структура связей

Для достижения максимальной универсальности и упрощения протоколов обмена информацией в микропроцессорных системах применяется так называемая шинная структура связей между отдельными устройствами, входящими в систему. Суть шинной структуры связей сводится к следующему.

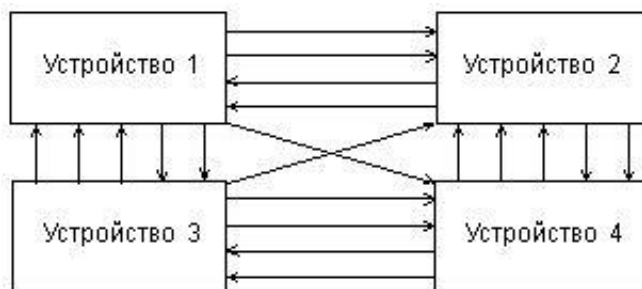


Рисунок 31. Классическая структура связей.

При классической структуре связей (Рис.) все сигналы и коды между устройствами передаются по отдельным линиям связи. Каждое устройство, входящее в систему, передает свои сигналы и коды независимо от других устройств. При этом в системе получается очень много линий связи и разных протоколов обмена информацией.

При шинной структуре связей (Рис. 12) все сигналы между устройствами передаются по одним и тем же линиям связи, но в разное время (это называется мультиплексированной передачей). Причем передача по всем линиям связи может осуществляться в обоих направлениях (так называемая двунаправленная передача). В результате количество линий связи существенно сокращается, а правила обмена (протоколы) упрощаются. Группа линий связи, по которым передаются сигналы или коды как раз и называется шиной (англ. bus).

Понятно, что при шинной структуре связей легко осуществляется пересылка всех информационных потоков в нужном направлении, например, их можно пропустить через один процессор, что очень важно для микропроцессорной системы. Однако при шинной структуре связей вся информация передается по линиям связи последовательно во времени, по очереди, что снижает быстродействие системы по сравнению с классической структурой связей.

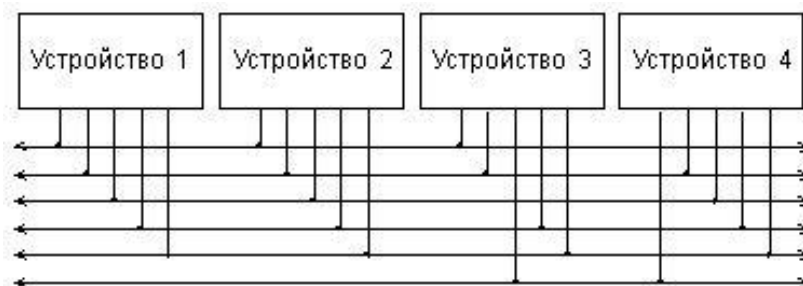


Рисунок 32. Шинная структура связей.

4. Материнская плата ПК

Системной (System Board) или материнской платой (Mother Board, m/b, MB) называют основную печатную плату ПК, на которой устанавливаются другие системные компоненты. Она представляет собой плоский лист фольгированного стеклотекстолита (в последние годы - пластика), на котором размещены основные компоненты. Компоненты подразделяются на:

Внутренние. Соединение этих элементов выполняется предварительным травлением медной фольги нанесенной на подложку стеклотекстолита. В технологическом цикле несколько листов соединяются в многослойную структуру, покрытую защищенным лаком.

Внешние. В системную плату устанавливается процессор и память, а также платы таких устройств, как внутренний модем, сетевая плата, видео-плата и т.п.

Системная плата содержит множество **внутренних компонентов**, определяющих архитектуру компьютера. Это:

- Разъём под центральный процессор;
- Cashe (кэш, L1-L2, напр. объёмом 256 Кбайт) и разъём под него;
- Контроллер винчестера и разъёмы шлейфов к жёстким дискам;

- Слоты (разъёмы) под оперативную память (два, три, четыре);
- Контроллер (коннектор) клавиатуры и мыши;
- Микросхема, обслуживающая флорру-дискковод, последовательные порты и параллельный порт;
- Перезаписываемая Bios (старое название Flash-память);
- Системные преобразователи напряжения;
- Слот (разъёмы) AGP/PCIEхpr-шины (для видео-карты, карты Internet и др.);
- Гнёзда расширения (слоты, Slot);
- Контроллеры внешних устройств;
- Разъёмы прочих шин (интерфейсные схемы);
- CNR-слот (либо для сетевой карты, либо для модема);
- AMR-слот (либо для звуковой карты [Audio Riser], либо для модема [Modem Riser]);
- Прочие компоненты.

Просмотрите и изучите ролик «Комплектация современной ЭВМ».



Рисунок 33 Комплектация современной ЭВМ

В рамках одного семейства производители выпускают большое число плат-близнецов, отличающихся только наличием или отсутствием дополнительных компонентов (SCSI-адаптер, звуковая или сетевая карта и т.п.). И производительность, и другие характеристики таких "близняшек" одинаковы.



Видеоурок 1. Материнская плата

5. Организация и основные характеристики памяти компьютера.

Постоянная память. Оперативная память

Компьютер (англ. computer — вычислитель, лат.computo—считаю, вычисляю) это универсальное (многофункциональное) электронное автоматическое устройство для накопления, обработки и передачи информации, представленной в виде двоичного кода.

В компьютере используется память нескольких типов, отличающихся по своему функциональному назначению и способами хранения информации. Память компьютера подразделяется на основную и внешнюю.

В состав внутренней памяти входят постоянная память, оперативная память и кэш-память.

В современных компьютерах устройства внешней памяти позволяют сохранять информацию после выключения компьютера.

Основная память, называемая иногда внутренней, располагается внутри системного блока. Она является обязательной составной частью компьютера. Основная память состоит из постоянной, оперативной и кэш-памяти.

Постоянная память, или постоянное запоминающее устройство - ПЗУ (англ. ROM, Read Only Memory),- память только для чтения.

Постоянная память энергонезависимая память, используется для хранения данных, которые никогда не потребуют изменения. Содержание памяти специальным образом “зашивается” в устройстве при его изготовлении для постоянного хранения. Из ПЗУ можно только читать.

В ПЗУ находятся загрузочные программы операционной системы в оперативную память, совокупность программ, предназначенных для автоматического тестирования устройств после включения питания компьютера BIOS (Basic Input/Output System — базовая система ввода-вывода). Роль BIOS двоякая: с одной стороны это неотъемлемый элемент аппаратуры (Hardware), а с другой стороны — важный модуль любой операционной системы (Software).

Разновидность постоянного ЗУ — CMOS RAM. CMOS RAM — это память с невысоким быстродействием и минимальным энергопотреблением от батарейки. Используется для хранения информации о конфигурации и составе оборудования компьютера, а также о режимах его работы. Содержимое CMOS изменяется специальной программой Setup, находящейся в BIOS (англ. Set-up — устанавливать, читается "сетап").

Оперативная память (ОЗУ, англ. RAM, Random Access Memory — память с произвольным доступом) - это быстрое запоминающее устройство не очень большого объёма, непосредственно связанное с процессором и предназначенное для записи, считывания и хранения выполняемых программ и данных, обрабатываемых этими программами.

В оперативном запоминающем устройстве происходит вся текущая деятельность компьютера. Туда загружаются копии различных программ. Там же с их помощью пользователем создаётся конечный продукт – тексты, рисунки, электронные таблицы, звуковые файлы и т. д., и т.п.

Оперативная память используется только для временного хранения данных и программ, так как, когда машина выключается, все, что находилось в ОЗУ, пропадает.

Структурно оперативную память можно представить себе как совокупность ячеек памяти, разделенных на разряды для хранения в каждом из них бита информации - регистр.

- register - простое устройство, предназначенное для временного хранения данных ограниченного размера.

Регистр предназначен для хранения данных, пересылаемых между компонентами компьютера. Важной характеристикой регистра является высокая скорость приема и выдачи данных. Регистр состоит из ячеек, в которые можно быстро записывать, запоминать и считывать слово, команду, двоичное число и т.д. Любой регистр характеризуется скоростью работы, числом бит, которые он может хранить. Регистр, накапливающий данные, именуют аккумулятором.

Регистр, обладающий способностью перемещать содержимое своих разрядов, называют сдвиговым. В этих регистрах за один такт хранимое слово поразрядно сдвигается на одну позицию справа налево, либо наоборот. Сдвиговые регистры используются при обработке сигналов, кодировании и декодировании.

Некоторые регистры служат счетчиками. Счетчик является устройством, которое на своих выходах выдает (в дискретной форме) сумму числа импульсов, подаваемых на его единственный вход. Максимальное число импульсов, которое счетчик может подсчитать, называется его емкостью.

Часто регистр имеет тот же размер, что и машинное слово. **Машинное слово** - фиксированная, упорядоченная последовательность битов, рассматриваемая аппаратной частью компьютера как единое целое. Машинное слово может быть разной длины в зависимости от типа компьютера (от 8 до 64 бит).

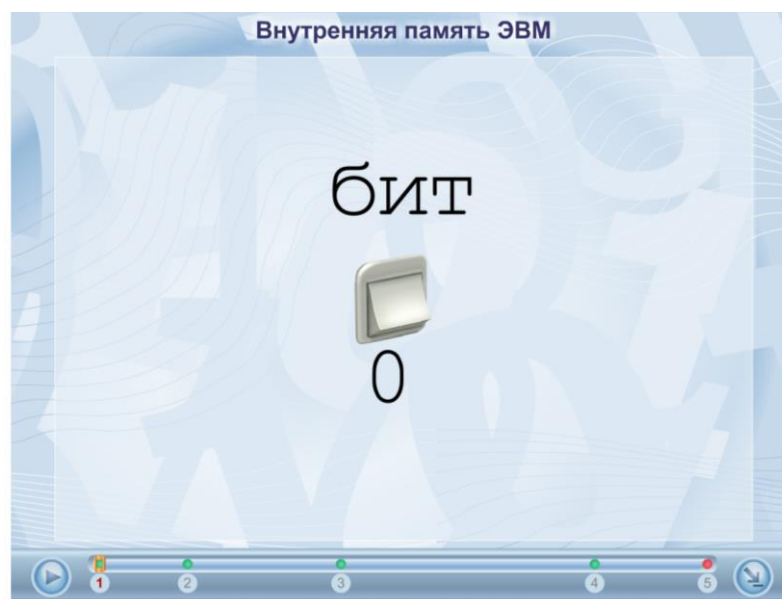


Рисунок 34 Строение внутренней памяти ЭВМ

Объем ОЗУ обычно составляет 4 - 128 Мбайта, а для эффективной работы современного программного обеспечения желательно иметь не менее 16 Мбайт ОЗУ. Обычно ОЗУ исполняется из интегральных микросхем памяти DRAM (Dynamic RAM — динамическое ОЗУ). Микросхемы DRAM работают медленнее, чем другие разновидности памяти, но стоят дешевле.

Каждый информационный бит в DRAM запоминается в виде электрического заряда крохотного конденсатора, образованного в структуре полупроводникового кристалла. Из-за токов утечки такие конденсаторы быстро разряжаются, и их периодически (примерно каждые 2 миллисекунды) подзаряжают специальные устройства. Этот процесс называется регенерацией памяти (Refresh Memory).

Современные микросхемы имеют ёмкость 1-16 Мбит и более. Они устанавливаются в корпуса и собираются в модули памяти.

Наиболее распространены модули типа SIMM (Single In-Line Memory Module — модуль памяти с однорядным расположением микросхем).

SIMM. Модуль памяти с однорядным расположением микросхем В модуле SIMM элементы памяти собраны на маленькой печатной плате длиной около 10 см. Ёмкость таких модулей неодинаковая — 256 Кбайт, 1, 2, 4, 8, 16,

32 и 64 Мбайта. Различные модули SIMM могут иметь разное число микросхем — девять, три или одну, и разное число контактов — 30 или 72.

Правильно памятью компьютера называть его оперативное запоминающее устройство – ОЗУ. Напомним, что находится оперативное запоминающее устройство на материнской плате и состоит из специальных модулей памяти (SIMM'ов, а в новых моделях компьютеров - DIMM'ов). Суммарный объём установленных в компьютере модулей памяти и определяет объём памяти конкретного компьютера. Важная характеристика модулей памяти — время доступа к данным, которое обычно составляет 60 – 80 наносекунд.

Кэш-память

Кэш (англ. cache), или сверхоперативная память — очень быстрое ЗУ небольшого объема, которое используется при обмене данными между микропроцессором и оперативной памятью для компенсации разницы в скорости обработки информации процессором и несколько менее быстродействующей оперативной памятью.

Кэш-памятью управляет специальное устройство — контроллер, который, анализируя выполняемую программу, пытается предвидеть, какие данные и команды вероятнее всего понадобятся в ближайшее время процессору, и подкачивает их в кэш-память. При этом возможны как "попадания", так и "промахи". В случае попадания, то есть, если в кэш подкачаны нужные данные, извлечение их из памяти происходит без задержки. Если же требуемая информация в кэше отсутствует, то процессор считывает её непосредственно из оперативной памяти. Соотношение числа попаданий и промахов определяет эффективность кэширования.

Кэш-память реализуется на микросхемах статической памяти SRAM (Static RAM), более быстродействующих, дорогих и малоемких, чем DRAM.

Современные микропроцессоры имеют встроенную кэш-память, так называемый кэш первого уровня размером 8–16 Кбайт. Кроме того, на

системной плате компьютера может быть установлен кэш второго уровня ёмкостью от 64 Кбайт до 256 Кбайт и выше.

6. Основные устройства

Аудиоадаптер (Sound Blaster или звуковая плата) это специальная электронная плата, которая позволяет записывать звук, воспроизводить его и создавать программными средствами с помощью микрофона, наушников, динамиков, встроенного синтезатора и другого оборудования.

Аудиоадаптер содержит в себе два преобразователя информации:

- **аналого-цифровой**, который преобразует непрерывные (то есть, аналоговые) звуковые сигналы (речь, музыку, шум) в цифровой двоичный код и записывает его на магнитный носитель;
- **цифро-аналоговый**, выполняющий обратное преобразование сохранённого в цифровом виде звука в аналоговый сигнал, который затем воспроизводится с помощью акустической системы, синтезатора звука или наушников.

Профессиональные звуковые платы позволяют выполнять сложную обработку звука, обеспечивают стереозвучание, имеют собственное ПЗУ с хранящимися в нём сотнями тембров звучаний различных музыкальных инструментов. Звуковые файлы обычно имеют очень большие размеры. Так, трёхминутный звуковой файл со стереозвучанием занимает примерно 30 Мбайт памяти. Поэтому платы **Sound Blaster**, помимо своих основных функций, обеспечивают автоматическое сжатие файлов.

Область применения звуковых плат – компьютерные игры, обучающие программные системы, рекламные презентации, "голосовая почта" (voice mail) между компьютерами, озвучивание различных процессов, происходящих в компьютерном оборудовании, таких, например, как отсутствие бумаги в принтере и т.п.

Видеоадаптер и графический акселератор

Видеоадаптер – это электронная плата, которая обрабатывает видеоданные (текст и графику) и управляет работой дисплея. Содержит видеопамять, регистры ввода вывода и модуль BIOS. Посылает в дисплей сигналы управления яркостью лучей и сигналы развертки изображения.

Наиболее распространенный видеоадаптер на сегодняшний день – **адаптер SVGA** (Super Video Graphics Array – супервидеографический массив), **который может отображать на экране дисплея 1280x1024 пикселей при 256 цветах и 1024x768 пикселей при 16 миллионах цветов.**

С увеличением числа приложений, использующих сложную графику и видео, наряду с традиционными видеоадаптерами широко используются разнообразные **устройства компьютерной обработки видеосигналов:**



Рисунок 35. Графический акселератор

Графические акселераторы (ускорители) – специализированные графические **сопроцессоры**, увеличивающие эффективность видеосистемы. Их применение освобождает центральный процессор от большого объема операций с видеоданными, так как акселераторы самостоятельно вычисляют, какие пиксели отображать на экране и каковы их цвета.

Фрейм-грабберы, которые позволяют отображать на экране компьютера видеосигнал от видеомэганитофона, камеры, лазерного проигрывателя и т. п., с тем, чтобы **захватить нужный кадр в память и впоследствии сохранить его в виде файла.**

TV-тюнеры – видеоплаты, превращающие компьютер в телевизор. TV-тюнер позволяет выбрать любую нужную телевизионную программу и отображать ее на экране в масштабируемом окне. Таким образом можно следить за ходом передачи, не прекращая работу.

Клавиатура

Клавиатура компьютера – устройство для ввода информации в компьютер и подачи управляющих сигналов. Содержит стандартный набор клавиш печатной машинки и некоторые дополнительные клавиши – управляющие и функциональные клавиши, клавиши управления курсором и малую цифровую клавиатуру.

Все символы, набираемые на клавиатуре, немедленно отображаются на мониторе в позиции курсора (курсор – светящийся символ на экране монитора, указывающий позицию, на которой будет отображаться следующий вводимый с клавиатуры знак).

Наиболее распространена сегодня клавиатура с раскладкой клавиш **QWERTY** (читается "кверти"), названная так по клавишам, расположенным в верхнем левом ряду алфавитно-цифровой части клавиатуры:



Рисунок 36. Клавиатура компьютера

Такая клавиатура имеет **12 функциональных клавиш**, расположенных вдоль верхнего края. Нажатие функциональной клавиши приводит к посылке в компьютер не одного символа, а целой совокупности символов. Функциональные клавиши могут программироваться пользователем. Например,

во многих программах для получения помощи (подсказки) задействована клавиша **F1**, а для выхода из программы – клавиша **F10**.

Управляющие клавиши имеют следующее назначение:

- **Enter** – клавиша **ввода**;
- **Esc** (Escape – выход) клавиша для **отмены** каких-либо действий, выхода из программы, из меню и т.п.;
- **Ctrl** и **Alt** – эти клавиши самостоятельного значения не имеют, но при нажатии совместно с другими управляющими клавишами изменяют их действие;
- **Shift** (регистр) – обеспечивает **смену регистра клавиш** (верхнего на нижний и наоборот);
- **Insert** (вставлять) – **переключает режимы вставки** (новые символы вводятся посреди уже набранных, раздвигая их) и **замены** (старые символы замещаются новыми);
- **Delete** (удалять) – **удаляет символ** с позиции курсора;
- **Back Space** или ← – **удаляет символ** перед курсором;
- **Home** и **End** – обеспечивают **перемещение курсора в первую и последнюю позицию строки**, соответственно;
- **Page Up** и **Page Down** – обеспечивают **перемещение по тексту на одну страницу** (один экран) назад и вперед, соответственно;
- **Tab** – **клавиша табуляции**, обеспечивает перемещение курсора вправо сразу на несколько позиций до очередной позиции табуляции;
- **Caps Lock** – фиксирует верхний регистр, обеспечивает **ввод прописных букв вместо строчных**;
- **Print Screen** – обеспечивает **печать информации**, видимой в текущий момент на экране.
- **Длинная нижняя клавиша без названия** – предназначена для **ввода пробелов**.

- Клавиши \uparrow , \downarrow , \leftarrow и \rightarrow служат для перемещения курсора **вверх, вниз, влево и вправо** на одну позицию или строку.

Малая цифровая клавиатура используется в двух режимах – **ввода чисел и управления курсором**. Переключение этих режимов осуществляется клавишей **Num Lock**.

Клавиатура содержит встроенный **микроконтроллер** (местное устройство управления), который выполняет следующие функции:

- последовательно опрашивает клавиши, считывая введенный сигнал и вырабатывая двоичный **скан-код** клавиши;
- управляет световыми индикаторами клавиатуры;
- проводит внутреннюю диагностику неисправностей;
- осуществляет взаимодействие с центральным процессором через порт ввода-вывода клавиатуры.

Клавиатура имеет **встроенный буфер** — промежуточную память малого размера, куда помещаются введенные символы. В случае переполнения буфера нажатие клавиши будет сопровождаться звуковым сигналом — это означает, что символ не введен (отвергнут). Работу клавиатуры поддерживают специальные программы, "защитые" в BIOS, а также драйвер клавиатуры, который обеспечивает возможность ввода русских букв, управление скоростью работы клавиатуры и др.

Видеосистема компьютера

Видеосистема компьютера состоит из трех компонент:

- **монитор** (называемый также дисплеем);
- **видеоадаптер**;
- **программное обеспечение** (драйверы видеосистемы).

Видеоадаптер посылает в монитор сигналы управления яркостью лучей и синхросигналы строчной и кадровой разверток. **Монитор** преобразует эти сигналы в зрительные образы. А **программные средства** обрабатывают

видеоизображения — выполняют кодирование и декодирование сигналов, координатные преобразования, сжатие изображений и др.

Монитор – устройство визуального отображения информации (в виде текста, таблиц, рисунков, чертежей и др.).



Рисунок 37. Монитор

Подавляющее большинство мониторов сконструированы на базе **электронно-лучевой трубки (ЭЛТ)**, и принцип их работы аналогичен принципу работы телевизора. Мониторы бывают алфавитно-цифровые и графические, монохромные и цветного изображения. Современные компьютеры комплектуются, как правило, цветными графическими мониторами.

Вопрос. Кодовая шина данных является составной частью:

- а) генератора тактовых импульсов;
- б) системной шины;
- в) микропроцессора;
- г) общей шины.

Ответ: б) системной шины.

Контрольные вопросы

1. Назовите устройства ввода информации.
2. Какое значение для компьютера имеет процессор?
3. Какое значение имеет ПЗУ (постоянная память) в компьютере?
4. Назовите устройства хранения информации объемом больше 1 Гбайт?
5. Назовите устройства вывода информации.

6. Какое значение для компьютера имеет жесткий диск?
7. Какое значение имеет ОЗУ (оперативная память) в компьютере?
8. Назовите устройства хранения информации объемом меньше 10 Гбайт?
9. Что размещается на материнской плате?
10. Что такое системная шина? Зависит ли от нее быстродействие компьютера?

Задания для самостоятельной работы

Задание 1.

Внимательно рассмотреть представленную материнскую плату и указать основные (выделенные желтым цветом) компоненты, а также их назначение.

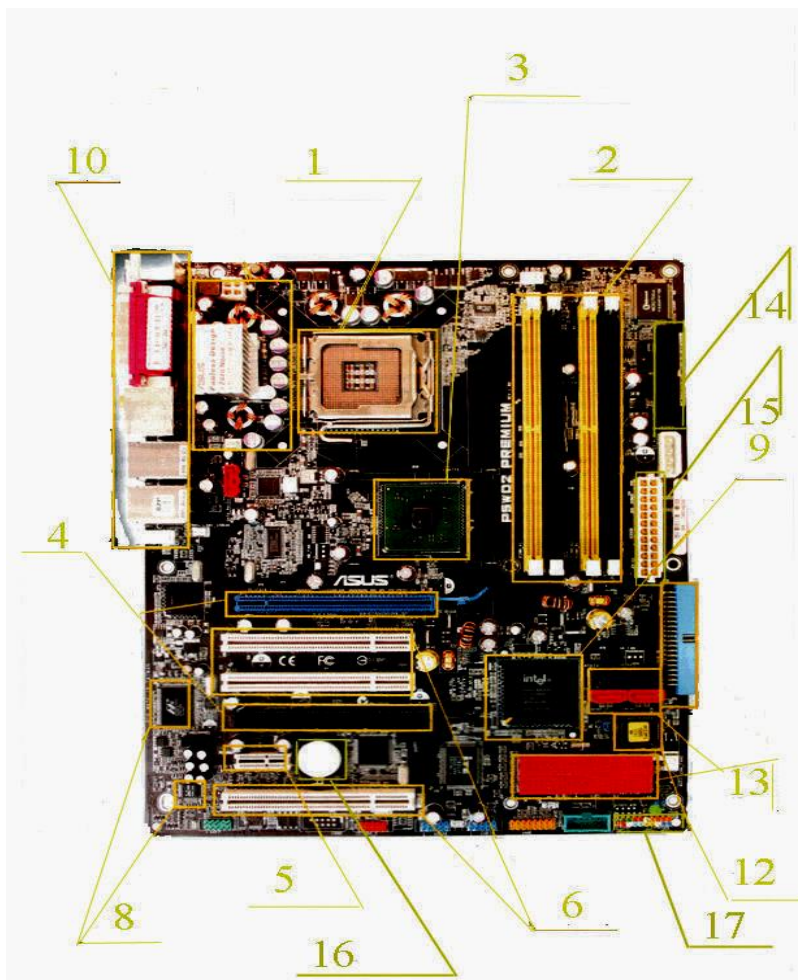


Рисунок 38

Раздел 4. Программные средства реализации информационных процессов

Тема 5. Классификация программного обеспечения ЭВМ

Лабораторная работа №7. Программное обеспечение ЭВМ (software)

1. Программное обеспечение компьютера (системное и прикладное).

Компьютер представляет собой вычислительную систему, состоящую из аппаратной части (ее часто называют английским словом hardware) и программного обеспечения (software). Программное обеспечение (ПО) — это совокупность всех программ и соответствующей документации, обеспечивающая целесообразное использование компьютерной системы. Полный комплект программного обеспечения, необходимый для организации, например, автоматизированного рабочего места (АРМ) инженера-проектировщика, по стоимости может в несколько раз превосходить стоимость компьютера адекватного класса.

ПО обеспечивает функционирование, диагностику и тестирование аппаратных средств компьютерных систем, а также разработку, отладку и выполнение любых задач пользователя.

Основными характеристиками программного обеспечения являются:

- алгоритмическая сложность;
- полнота и системность функций обработки;
- глубина реализованных функций обработки;
- объем файлов программ;
- требования к техническим средствам;
- размер оперативной памяти для запуска программ;
- тип процессора;
- требования к операционной системе;

- использование локальной и/или глобальной сети и др.

В ПО можно выделить системное программное обеспечение (СПО) и прикладное программное обеспечение (ППО).

Системное программное обеспечение управляет всеми ресурсами компьютера (центральный процессор, память, внешние устройства и др.) и осуществляет общую организацию процесса обработки информации и взаимодействие с пользователем. К СПО относят:

- операционные системы (ОС);
- утилиты ОС;
- оболочки ОС;
- средства тестирования и диагностики;
- системы программирования.

Прикладное программное обеспечение составляют пакеты прикладных программ, предназначенных для решения определенного круга задач производственного, научного, управленческого, учебного и др. характера из различных проблемных областей. К ППО относят:

- пакеты прикладных программ общего назначения (текстовые редакторы, графические редакторы, СУБД, табличные процессоры, игровые программы);
- пакеты прикладных программ специального назначения (математические пакеты, САПР, бухгалтерские пакеты, экспертные системы);
- электронные издания образовательного назначения (энциклопедии, автоматизированные обучающие системы).

Выделяют также инструментальное программное обеспечение, предназначенное для создания оригинальных программных средств в любой предметной области, с том числе и производства системного программного обеспечения.

Компьютер - это устройство, не способное «думать» самостоятельно, как человек. Для того чтобы компьютер стал «умным», и мог работать с информацией - получать, хранить, передавать, обрабатывать — его необходимо научить исполнять все эти действия. Научить - значит построить работу

компьютера по инструкции, в которой указано, что надо делать. Такая инструкция должна содержать строгую последовательность команд на языке, понятном компьютеру, т.е. сообщать ему, как надо обрабатывать данные для получения, желаемого результата. Такая инструкция называется *программой*. Компьютер без программ - это, груда железа. И только программы делают его нашим помощником, другом.



Рисунок 39

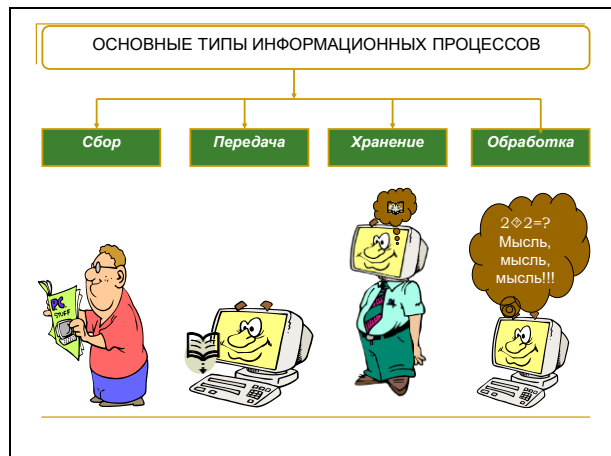


Рисунок 40

Программа должна быть написана на языке, понятном компьютеру, и такой язык существует. Он так и называется - *машинный язык*. Слова этого языка (команды) представляют собой последовательности нулей и единиц.

Как вы знаете, связано это с тем, что вся информация в компьютере представлена в знаковой двоичной форме.

Так что же такое «программа»?

Программа – это последовательность команд, которую должен выполнить компьютер, чтобы решить поставленную задачу обработки информации.

Каким образом создаются программы?

Человек сначала продумывает, каким образом должен действовать компьютер для достижения желаемого результата, и представляет предполагаемый план действий в виде команд. Далее эти команды необходимо преобразовать в форму, понятную компьютеру, т.е., опираясь на доступные компьютеру команды, составить программу действий компьютера. Такой процесс называется программированием, а люди, которые его выполняют, называются *программистами*. Можно сказать, что они являются компьютерными учителями.

Программирование — деятельность человека по созданию программы.

Все программы хранятся во внешней или постоянной памяти компьютера. Но, для того, чтобы компьютер мог выполнить ту или иную обработку информации по соответствующей программе, она должна быть помещена (загружена) в оперативную память, т. к. обработка ведется процессором, который взаимодействует напрямую только с оперативной памятью, и доступа к внешней памяти процессор не имеет.

Вся совокупность программ, хранящаяся на всех устройствах долговременной памяти компьютера составляет его программное обеспечение.

Программное обеспечение (ПО) — все многообразие программ, используемых в современном компьютере.

Совокупность программ, подготовленных к работе, называют установленным программным обеспечением.

Программы создавались в течение нескольких десятилетий и поэтому сегодня их насчитывается огромное количество.

2. Немного истории.

Самые первые программы, разрабатывались непосредственно на машинном языке и не все люди могли заниматься их разработкой, только профессиональные программисты.

В 60 -е годы началась разработка языков программирования высокого уровня, которые позволили существенно облегчить работу программистов, и создание программ стало доступно даже для начинающих пользователей.



Рисунок 41.

Изначально, компьютеры (50-60-е) могли только вычислять, т.е. обрабатывать числовую информацию.

В 70 – е годы компьютер «научился» работать с текстом и до сих пор чаще всего компьютеры используются для работы с текстовой информацией.



Рисунок 42.

В 80-е годы появились первые компьютеры, способные работать с графической информацией. Сейчас компьютерная графика является интересной областью информационных технологий и используется в компьютерных экспериментах, в Интернете, телевидении, кино и т.д.

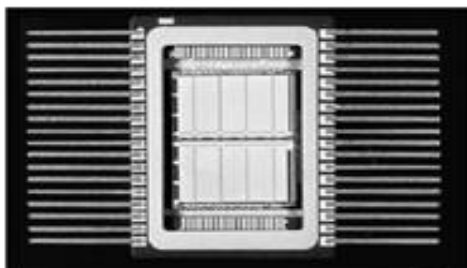


Рисунок 43.

В 90-е годы компьютер получил возможность обрабатывать звуковую информацию. Практически любой современный пользователь может записывать и редактировать аудиоклипы.

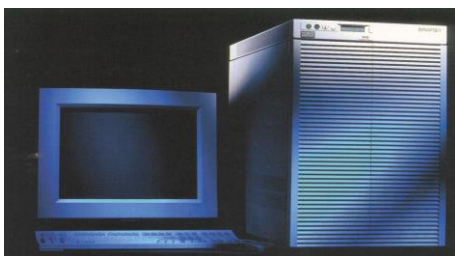


Рисунок 44.

Вывод: таким образом, для обработки информации на компьютере необходимо иметь не только аппаратное обеспечение - устройства компьютерной системы («**hardware**»), но и программное обеспечение («**software**») которое наделяет компьютер мыслями и интеллектом.

Компьютер представляет собой единство двух составляющих: аппаратуры и программного обеспечения (ПО).



Рисунок 45

3. Виды ПО.

Все программное обеспечение делится на три класса:

ВИДЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ



Рисунок 46.

Системное ПО - является основным ПО, неотъемлемой частью компьютера. Без него невозможно взаимодействовать ни с одним устройством компьютера. Именно системное ПО руководит слаженной работой всех элементов компьютерной системы, как на аппаратном уровне, так и на программном.

Ни с одним из других видов ПО Вы не сможете работать, если в Вашем компьютере отсутствуют некоторые системные программы. Именно они заставляют компьютер «действовать», «не спать», «дышать».

Прикладное ПО — предназначено для выполнения конкретных задач пользователя. Это те программы, которые превращают компьютер:

- в пишущую машинку для набора текстов;
- в калькулятор для выполнения вычислений;
- в рабочее место художника, дизайнера, монтажера;
- в средство общения с другими людьми на расстоянии или в режиме реального времени;
- в рабочее место ученого-исследователя или инженера-конструктора;
- в учителя, который обучает нас чему-либо и многое-многое другое.

Инструментарий программирования - это средства, предназначенные для создания ПО, т.е. того же системного и прикладного ПО. Его составляют разнообразные языки и среды программирования, такие как Бейсик, Паскаль. С, С ++ , Delphi и другие.

Рассмотрим более подробную классификацию программного обеспечения:

Программное обеспечение			
СИСТЕМНОЕ ПО		ПРИКЛАДНОЕ ПО	
базовое	сервисное	Текстовые процессоры	Трансляторы
Операционные системы	Программы диагностики работоспособности ПК	Табличные процессоры	Отладчики
Операционные оболочки	Антивирусные программы	СУБД	Интегрированные среды разработки приложений
Сетевые операционные системы	Программы обслуживания дисков	Компьютерная графика и анимация	Языки программирования
	Программы архивирования данных	Коммуникационные пакеты	
	Утилиты	Средства создания презентаций	
	Драйверы	Системы автоматизированного проектирования (САПР)	
		Средства автоматизации производства	
		Обучающие программы	
		Игры	

Для чего же предназначен тот, или иной вид программного обеспечения:

Таблица 12

Программы	Пояснения	Примеры
Системное программное обеспечение		

Программы	Пояснения	Примеры
Операционные системы	Комплекс программ, распределяющих ресурсы компьютерной системы и организующих работу других программ	MS-DOS WINDOWS UNIX
Программы оболочки	программы, обеспечивающие более комфортное общение пользователя с командами ОС	Norton Commander Windows Commander Far
Программы диагностики	Проверяют работу основных устройств компьютера	
Антивирусные программы	Программы обнаружения компьютерных вирусов и их уничтожения	DrWeb, AIDTEST, Антивирус Касперского
Программы обслуживания дисков	Программы проверки целостности логической и физической структуры дисков, дефрагментация	
Архиваторы	Программы упаковки файлов и группы файлов для уменьшения занимаемого ими места на диске	WinRar WinZip ARJ
Драйверы	Расширяют возможности ОС по управлению устройствами ввода/вывода компьютера (клавиатурой, жестким диском, мышью и т.д.)	
Утилиты	Программы вспомогательного назначения	Norton Utilites
Прикладное программное обеспечение		
Текстовые процессоры	Программы для создания, редактирования и оформления текстовых	Microsoft Word

Программы	Пояснения	Примеры
	документов	
Табличные процессоры	Программы, позволяющие выполнять операции над данными, представленными в табличной форме	Microsoft Excel Бухгалтерские программы Бухгалтерия 1С:
СУБД	Средства ввода, поиска, размещения и выдачи больших массивов данных	Microsoft Access
Компьютерная графика и анимация	Средства создания неподвижных и движущихся объектов	Графические редакторы Paint Adobe Photoshop Corel Draw
Коммуникационные пакеты	Программы для работы в компьютерной сети	Internet Explorer Opera Outlook Express ISQ
Средства создания презентаций	Программы создания и показа наборов слайдов	Microsoft Power Point
Системы автоматизированного проектирования (САПР)	Средства проектирования электронных схем, машин, механизмов	AutoCad КОМПАС
Средства автоматизации производства	Программы, позволяющие использовать компьютер в производственном процессе	
Обучающие программы	Помогают процессу обучения	Клавиатурные тренажеры Тесты
Игры	Программы для организации досуга и обучения	Стратегии Лабиринты Логика
Инструментарий программирования		
Трансляторы	Переводчики программ языков программирования в машинные коды	
Отладчики	Средства поиска и исправления ошибок	

Программы	Пояснения	Примеры
Интегрированные среды разработки приложений	Объектно-программированные языки программирования	Visual Basic Delpfi
Языки программирования	Средства создания программ для компьютера	Basic, Pascal. C, C + +

Кроме вышеприведенной классификации все программное обеспечение можно систематизировать по другому параметру, а именно по длительности нахождения в оперативной памяти компьютера. С этой точки зрения различаются *резидентные* и *нерезидентные* программы.

Резидентные программы — это программы, на протяжении всей работы компьютера находящиеся в его оперативной памяти. Их постоянное присутствие в ОЗУ необходимо в связи с тем, что эти программы на протяжении всего периода включения компьютерной системы следят за ее состоянием. К таким программам, например, относятся операционная система и антивирусные программы, т.к. под управлением операционной системы вообще работает компьютер, а антивирусная программа постоянно должна контролировать изменения, происходящие в файловой системе, чтобы предупредить проникновение компьютерных вирусов.

Нерезидентные программы - это программы, которые по окончании своей работы выгружаются полностью или частично из ОЗУ. Это, например, прикладные программы.

Что было бы, если бы все программы оставались в ОЗУ? Дело в том, что чем ближе ПО к «железу» компьютерной системы, тем сложнее его роль в организации работы устройств и тем сложнее пользователю работать в такой среде. Поэтому наиболее дружелюбно пользователю прикладное ПО.

Совокупность программ, работающих в тот или иной момент времени, называют программной конфигурацией.



Рисунок 47

4. Назначение и состав операционной системы компьютера. Загрузка компьютера.

Компьютер представляет собой вычислительную систему, состоящую из аппаратной части (ее часто называют английским словом hardware) и программного обеспечения (software). Программное обеспечение состоит из большого количества самых разнообразных готовых программ, а также программ, написанных самим пользователем. Для работы компьютера требуется основное программное обеспечение — операционная система (ОС), поставляемая вместе с ним. Без операционной системы компьютер работать не может. ОС относится к системному программному обеспечению (см. билет № 7, вопрос 1).

Операционная система — это комплект программ, организующих работу компьютера и управляющих им. Операционная система выполняет следующие функции:

- управление памятью;
- управление вводом/выводом;
- управление файловой системой;
- защита памяти и программ;
- учет использования ресурсов компьютера;
- управление работой прикладных программ.

В состав современных операционных систем обычно входят следующие основные модули:

- программный модуль, управляющий файловой системой;
- командный процессор, выполняющий команды пользователя (запуск программы, копирование, удаление, переименование файла, вывода документа на печать и так далее);
- драйверы устройств — специальные программы, управляющие работой подключаемых к компьютеру устройств (дискководы, монитор, клавиатура, мышь, принтер и др.);
- программные модули, обеспечивающие графический пользовательский интерфейс;
- сервисные программы или утилиты, обслуживающие диски (проверка, сжатие, дефрагментация), архивирующие файлы и так далее;
- справочная система, позволяющая оперативно получать необходимую информацию о функционировании операционной системы.

По способу организации диалога с пользователем различают:

- *ОС с интерфейсом командной строки;*
- *ОС с графическим интерфейсом.*

Первые операционные системы (например, MS-DOS) вели с пользователем на экране текстового дисплея диалог, в ходе которого человек и компьютер обменивались сообщениями: человек вводил очередную команду, а компьютер, проверив её, либо выполнял, либо отвергал по причине ошибки. Такие системы принято называть ОС с интерфейсом командной строки.

С развитием графических возможностей компьютерной техники командная строка практически повсеместно была заменена графическим интерфейсом (например, Windows), когда объекты манипуляций в ОС изображаются в виде небольших рисунков (пиктограмм), а необходимые действия тем или иным образом выбираются из предлагаемого машиной списка — так называемого меню. При подобном способе организации диалога набор

текста команд на клавиатуре не требуется; вся нагрузка по вводу информации возлагается на манипулятор «мышь».

Рассмотрим подробно, как осуществляется *загрузка компьютера*.

В состав компьютера входит постоянное запоминающее устройство (ПЗУ), содержащее программы тестирования компьютера и первого этапа загрузки операционной системы BIOS (Basic Input/Output System — базовая система ввода/вывода). ПЗУ расположено на системной плате и питается от батарейки, поэтому записанные в нем программы не стираются при выключении компьютера.

После включения компьютера программы BIOS начинают выполняться; информация о ходе этого процесса отображается на экране монитора. Сначала производится тестирование и настройка аппаратных средств. Современные компьютеры в основном используют внешние устройства «plug and play» (переводится — «включил и работай»), поэтому они способны сообщить процессору свои основные характеристики и условия работы. В случае, если все оборудование функционирует нормально, происходит переход к следующему этапу — поиску начального загрузчика операционной системы.

Диск (жесткий, гибкий или лазерный), на котором находятся файлы операционной системы и с которого производится ее загрузка, называется системным. На этапе загрузки происходит поочередное обращение к имеющимся в компьютере дискам (гибким, жестким, CD-ROM) и поиск на определенном месте (в первом, так называемом *загрузочном секторе* диска) специальной программы Master Boot (программа-загрузчик операционной системы).

Если диск системный, то программа-загрузчик считывается в оперативную память. Затем эта программа ищет файлы операционной системы на системном диске и загружает их в оперативную память в качестве программных модулей.

Если системный диск по какой-то причине в компьютере отсутствуют, на экране монитора появляется сообщение Non system disk, загрузка операционной системы прекращается, и компьютер остается неработоспособным.

После окончания загрузки операционной системы управление передается командному процессору. В случае использования интерфейса командной строки на экране появляется приглашение системы, в противном случае загружается графический интерфейс. После этого можно приступать к работе.

6. Вспомните структуру дисков внешней памяти?



Рисунок 48

Хранится информация в компьютере на дисках внешней памяти. Объем внешней памяти достаточно большой, поэтому порядок в расположении информации на дисках очень важен. Информация хранится в структурированном виде, обеспечивающем ее быстрый поиск. Структурировать информацию на диске возможно благодаря форматированию диска, т. е. разбиению его на сектора и дорожки. Такая структура поверхности диска напоминает структуру книги, в которой информация расположена упорядоченно:

- книга разделена на страницы;
- страницы имеют номера,

поэтому для поиска нужной информации в книге достаточно знать номер страницы, на которой она находится.

Управление хранением информации на дисках внешней памяти осуществляет функциональная часть ОС, которая называется файловой системой.

Для удобства работы с данными на дисках их размещают в файлах.

Файл — это данные или программа, имеющие собственное имя и занимающие определенное место на диске.

Файл – это наименьшая единица хранения информации.

Нужный файл мы находим в компьютере по его *адресу*

Все адреса запоминаются на диске в так называемой **Таблице Размещения Файлов** (FAT – таблица)

Каждый диск (физический или логический) имеет свою таблицу размещения файлов

На магнитном диске минимальным адресуемым элементом является сектор

При записи файла на магнитном диске будет занято всегда целое количество секторов, соответственно минимальный размер файла — это размер одного сектора, а максимальный соответствует общему количеству секторов на диске.

Файл записывается в произвольные свободные сектора, которые могут находиться на различных дорожках. Например, **Файл1** объемом 2 Кбайта может занимать сектора 34, 35 и 47, 48, а **Файл2** объемом 1 Кбайт — сектора 36 и 49.

Жесткие диски имеют **большие объемы!!!!**

Минимальным адресуемым элементом жесткого диска является кластер, который может включать в себя несколько секторов.

Размер кластера зависит от *типа* используемой таблицы FAT и от емкости жесткого диска.

Файлу на жестком диске выделяется целое число блоков (кластеров)!

Чем больше кластер, тем больше пустого места, которое занято!

5. Файловая система. Папки и файлы. Имя, тип, путь доступа к файлу.

Все программы и данные хранятся в долговременной (внешней) памяти компьютера в виде файлов. Файл — это информация, хранящаяся во внешней памяти как единое целое и обозначенная одним именем.

Файл характеризуется набором параметров (имя, расширение, размер, дата создания, дата последней модификации) и атрибутами, используемыми операционной системой для его обработки (архивный, системный, скрытый, только для чтения и др.).

Размер файла выражается в байтах.

Имя файла состоит из двух частей, разделенных точкой: собственно имя файла и расширение, обозначающее его тип (программа, текст, рисунок и так далее). Собственно имя файлу дает пользователь, а расширение обычно задается программой автоматически при создании файла.

В операционных системах семейства DOS имя файла может содержать от 1 до 8 символов, можно использовать символы латинского алфавита, арабские цифры и некоторые другие символы. В операционной системе Windows имя файла может иметь до 255 символов, причем можно использовать буквы национальных алфавитов и пробелы.

Расширение имени файла записывается после точки и может содержать от 1 до 3 символов в MS DOS и больше 3 — в Windows. Чаще всего в расширение вкладывается определенный смысл (хотя пользователь может задавать и бессмысленные расширения) — оно указывает на содержимое файла или на то, какой программой был создан данный файл.

В таблице приведены наиболее распространенные типы файлов и их расширений:

<i>Тип файла</i>	<i>Расширения</i>
Системный	drv, sys
Готовая к выполнению программа	exe, com
Графический	bmp, jpg, gif
Текстовый	txt, doc
Звуковой	mid, wav
Видеофайл	Avi

Файловая система — это функциональная часть операционной системы, обеспечивающая выполнение операций над файлами.

На каждом носителе информации (гибком, жестком или лазерном диске) может храниться большое количество файлов. Для удобства поиска информации файлы, по принадлежности к какому-либо одному разделу, объединяют в группы, называемые каталогами (DOS) или папками (Windows).

Каталог, так же, как и файлы, находится на диске. Каталог содержит названия файлов и указание на их место размещения на диске. Каталог также получает собственное имя. Он сам может входить в состав другого, внешнего по отношению к нему каталога. Каждый каталог может содержать внутри себя множество файлов и вложенных каталогов. Таким образом, каталог — это поименованная совокупность файлов и подкаталогов (вложенных каталогов). Каталог самого верхнего уровня называется корневым каталогом. Его имеет изначально любой носитель; корневой каталог создается операционной системой без нашего участия.

Файловая структура диска — это совокупность файлов на диске и взаимосвязей между ними. Файловые структуры бывают простыми и многоуровневыми (иерархическими).

Простые файловые структуры могут использоваться для дисков с небольшим (до нескольких десятков) количеством файлов. В этом случае оглавление диска представляет собой линейную последовательность имен файлов.

Многоуровневые файловые структуры используются для хранения большого (сотни и тысячи) количества файлов. Начальный, корневой, каталог содержит файлы и вложенные каталоги 1-го уровня. Каждый из каталогов 1-го уровня может содержать не только файлы, но и вложенные каталоги второго уровня и так далее.

Графическое изображение иерархической файловой структуры называется *деревом*.

Чтобы найти нужный файл в файловой системе, имеющей иерархическую структуру, необходимо указать путь к файлу. В *путь к файлу* входит последовательность имен каталогов от корневого до того, в котором

непосредственно находится файл, разделяющихся при записи обратной косой чертой (\).

Полное имя файла состоит из пути к файлу (начиная с имени диска) и имени самого файла.

Все файлы можно разделить на *исполнимые* (программы) и *неисполнимые* (файлы данных и документов). Исполнимые файлы могут запускаться операционной системой на выполнение, а неисполнимые файлы могут только изменять свое содержимое в процессе выполнения программы.

Можно разделить файлы на:

- *основные* — их наличие необходимо для работы операционной системы и программных продуктов;
- *служебные* — хранящие конфигурацию и настройки основных файлов;
- *рабочие* — их содержимое изменяется в результате работы основных программных файлов; именно ради них и создаются все остальные файлы;
- *временные* — создающиеся в момент работы основных файлов и хранящие промежуточные результаты.

В зависимости от значений атрибутов файлов операционная система разрешает или запрещает те или иные действия над файлами.

В процессе работы на компьютере над файлами наиболее часто проводятся следующие операции:

- *копирование* (создается копия в другом каталоге или на другом носителе);
- *перемещение* (производится копирование файла в другой каталог или на другой носитель, в исходном каталоге объект уничтожается);
- *удаление* (в исходном каталоге объект уничтожается);
- *переименование* (изменяется имя файла).

Фрагментация - разбиение файла на несколько частей, хранящихся в разных областях диска. Фрагментация порождается в процессе удаления-

добавления файлов на диске. Она замедляет доступ к диску и уменьшает (как правило, не существенно) общее быстродействие диска.

Дефрагментация - процесс перезаписи частей файла в соседние сектора на жестком диске для ускорения доступа и загрузки. При обновлении файла компьютер стремится сохранить изменения в наибольшей свободной области на жестком диске. При этом происходит фрагментация, которая приводит к дополнительным затратам на поиск всех частей открываемого файла, что в свою очередь увеличивает время отклика системы.

Для операционной системы очень важно различать, что хранится внутри файла. Человек визуально различает в книге тексты и рисунки, графики и таблицы, инструкции и правила. На диске все виды информации представлены совершенно одинаково, любая информация в компьютере – текст, рисунки, программы, звук кодируется последовательностью нулей и единиц, т. е. представлены в двоичном виде.

Что, например, может означать следующая последовательность , 00001110 01001111 01010010 01001101 01000001 01010100? Может это 2 цветные точки? Или шесть серых точек? Или закодированная дата и время запуска первого спутника Земли? Или музыкальная мелодия? Или слово FORMAT закодированное по стандарту ASCII? Так вот, для того, чтобы операционная система различала, что хранится внутри файла, и не пыталась рисовать текст на экране или печатать мелодию, или озвучивать программу.

У каждого файла, помимо имени, должен быть специальным образом присваивающийся формат, указывающий на тип хранимой в нем информации. На содержимое файла указывает расширение, записываемое после имени файла и отделяемое от него точкой.

Таблица 13. Таблица некоторых форматов (расширений) файлов:

Тип	Значение	Примеры приложений, открывающие файлы с данными расширениями.
doc или	Файл содержит текстовую	MS Word, WordPad

Тип	Значение	Примеры приложений, открывающие файлы с данными расширениями.
txt	информацию	
bmp.jpg, gif	Файл содержит графическую информацию	Adobe PhotoShop, ACDSee
avi	Файл содержит видеoinформацию	WInAmp, Windows Медиа Проигрыватель
wav	файл содержит звуковую информацию	WinAmp, Windows Медиа Проигрыватель
bak	Файл-копия	Открывается программой, в которой был создан оригинал
htmi, him	Файл содержит Web-страницу	Internet Explorer
arj, rar, zip	Файл содержит архив (сжатую информацию)	WinRar, WnZip
exe, com	Файл содержит программу - исполняемый файл	Открывает саму программу

Таким образом, делаем следующий вывод по всему вышесказанному:

1. Вся информация на дисках хранится в файлах.
2. На содержимое файла указывает расширение, которое состоит из трех символов.
3. Файл открывается той программой, в которой был создан, или универсальной программой.

Кроме того:

4. Файлы со схожим содержимым или одинаковым назначением можно «складывать» в папки (каталоги), так как их бывает очень много, и бессистемное хранение может сильно затруднить работу пользователя. В каталогах на дисках регистрируются имена файлов, сведения об их размерах, дата и время создания.
5. Каталоги (папки), в свою очередь также можно объединять в каталоги.
6. На каждом диске имеется так называемый корневой каталог, который обозначается именем диска. В него занесены все записанные на этом диске каталоги и файлы, образуя целое «дерево» каталогов.

На диске формируется упорядоченная структура файлов и каталогов, которая позволяет быстро находить нужную информацию. Такая структура называется файловой системой и выглядит она следующим образом:

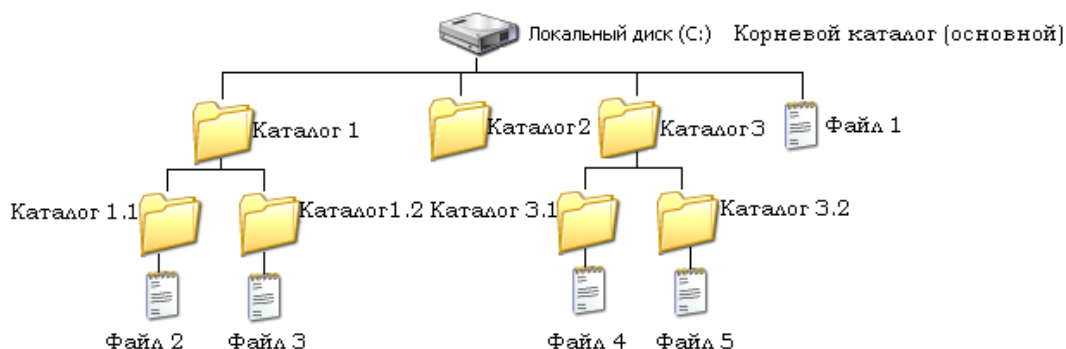


Рисунок 49

Многоуровневая иерархическая файловая система

Для дисков с небольшим количеством файлов (до нескольких десятков) удобно применять *одноуровневую файловую систему*, когда каталог (оглавление диска) представляет собой линейную последовательность имен файлов. Для отыскания файла на диске достаточно указать лишь имя файла.



Рисунок 50

Одноуровневая файловая система

Каталог, в котором работает пользователь, называется текущим.

Для того чтобы работать с каким-либо файлом, находящемся не в текущем каталоге, необходимо указать, на каком диске он находится и в каких каталогах всех уровней, — то есть указать **путь к файлу** — последовательность из имен каталогов, разделенных символом «\».

Например, нужный файл Файл 4 находится в подкаталоге Каталог 3.1, подкаталога Каталог 3 на диске С, тогда путь к файлу нужно записать так:

C:\Каталог 3\Каталог 3.1\Файл 4

Путь к файлу вместе с именем файла называют *полным именем файла*.

Шаблоны поиска

- При поиске файла по имени можно использовать шаблон: звездочка «*» и знак вопроса «?».
- Символ шаблона звездочка «*» заменяет любое количество символов, знак вопроса «?» - один символ.

Например, после ввода для поиска «Мышь?.doc» будут найдены файлы: «Мышь.doc» и «Мыши.doc».

После ввода «*.jpg» будут найдены все рисунки в формате jpg.

Маски имен файлов (шаблоны) используются также для групповых операций с файлами.

Операции с файлами и папками

- ***Копирование*** (копия файла помещается в другой каталог);
- ***Перемещение*** (сам файл перемещается в другой каталог);
- ***Удаление*** (запись о файле удаляется из каталога);
- ***Переименование*** (изменяется имя файла).

Такие действия с файлами, как «создать», «сохранить», «закрыть» можно выполнить только в прикладных программах («Блокнот», «Paint», ...).

Действия «открыть», «переименовать», «переместить», «копировать», «удалить» можно выполнить в системной среде.

Примеры решения задач:

Пример 1:

Определите, какое из указанных имен файлов удовлетворяет маске:

?hel*lo.c?*

1) **hello.c** 2) **hello.cpp** 3) **hhelolo.cpp** 4) **hhelolo.c**

Решение:

будем проверять соответствие файлов маске по частям, записывая результаты в таблицу

начнем с первой части маски, «**?hel**»; эта часть означает, что перед сочетанием «**hel**» в начале имени стоит один любой символ;

сразу видим, что первые два имени не подходят (начинаются прямо с «**hel**», без стартового символа), отмечаем их крестиком в таблице и больше не рассматриваем:

	?hel
hello.cp	×
hello.cpp	×
hhelolo.cpp	hhelolo.cpp
hhelolo.c	hhelolo.c

желтым и фиолетовым маркером в таблице выделены соответствующие части маски и имен файлов (где есть совпадение)

для двух последних имен проверяем второй блок маски: после «**hel**» должна быть цепочка «**lo.c**», или вплотную (и это возможно!) или через произвольную «вставку», на которую указывает звездочка в маске; видим, что оба имени прошли проверку:

	?hel	?hel*lo.c
hello.cp	×	
hello.cpp	×	
hhelolo.cpp	hhelolo.cpp	hhelolo.cpp
hhelolo.c	hhelolo.c	hhelolo.c

последняя часть маски, «**?***», означает, что после «**lo.c**» должен стоять по крайней мере один любой символ (на это указывает знак «**?**»); проверяя это правило, обнаруживаем, что для последнего имени, «**hhelolo.c**», маска не подходит, поскольку после «**lo.c**» ни одного символа нет:

	?hel	?hel*lo.c	?hel*lo.c?*
hello.cp	×		
hello.cpp	×		
hhelolo.cpp	hhelolo.cpp	hhelolo.cpp	hhelolo.cpp
hhelolo.c	hhelolo.c	hhelolo.c	×

таким образом, правильный ответ – 3.

Возможные ловушки и проблемы:

можно забыть, что звездочка «*» может соответствовать и пустой последовательности; например, в рассмотренной задаче имя «**hhelolo.cp**» также соответствует маске

можно забыть, что знак «?» НЕ может соответствовать пустой последовательности, а заменяет ровно 1 символ

Пример 2:

*Перемещаясь из одного каталога в другой, пользователь последовательно посетил каталоги **DOC, USER, SCHOOL, A:\, LETTER, INBOX**. При каждом перемещении пользователь либо спускался в каталог на уровень ниже, либо поднимался на уровень выше. Каково полное имя каталога, из которого начал перемещение пользователь?*

- 1) **A:\DOC**
- 2) **A:\LETTER\INBOX**
- 3) **A:\SCHOOL\USER\DOC**
- 4) **A:\DOC\USER\SCHOOL**

Решение:

сразу отметим, что здесь предполагается, что пользователь не переходил на другой диск;

в задачах, где нужно определить полный адрес файла или каталога, нужно начинать с поиска имени диска, в данном случае каталог находится на диске **A:** (так обозначается корневой каталог)

поскольку в списке посещенных каталогов перед **A:** стоит **SCHOOL**, пользователь мог попасть в корень диска **A:** только через каталог **SCHOOL**, поэтому адрес стартового каталога начинается с **A:\SCHOOL**

так как среди предложенных вариантов только один удовлетворяет этому условию, можно остановиться (правильный ответ – 3), однако, давайте все же доведем процедуру до конца, это позволит выяснить некоторые интересные моменты, которые не сработали здесь, но могут сработать в других задачах

как пользователь попал в каталог **SCHOOL**? по условию он мог как

подниматься, так и спускаться по дереву каталогов

явно он не мог спуститься из **USER** в **SCHOOL**, потому что **SCHOOL** находится в корневом каталоге и каждый каталог имеет только одного «родителя»; значит, пользователь поднялся из **USER** в **SCHOOL**, и начальная часть полного адреса **A:\SCHOOL\USER**

аналогично можно доказать, что пользователь поднялся в каталог **USER** из каталога **DOC**

таким образом, мы доказали, что правильный ответ – 3.

Возможные ловушки и проблемы:

можно забыть, что названия каталогов в полном имени перечисляются в порядке входа в них, начиная от корневого; пользователь *выходил* из каталогов, поднимаясь к корню диска **A:**, поэтому проходил каталоги в обратном порядке (на эту ошибку рассчитан неверный ответ **A:\DOC\USER\SCHOOL**)

можно перепутать каталог, из которого вышел пользователь, и каталог, где он в конечном счете оказался (на эту ошибку рассчитан неверный ответ **A:\LETTER\INBOX**)

в условии есть лишние данные, которые только запутывают дело; например, имена каталогов **LETTER**, **INBOX** никак не влияют на ответ, потому что пользователь попал в них уже *после* выхода в корневой каталог диска **A:**, то есть, пройдя весь путь исходного каталога в обратном порядке

Пример 3:

Каталог содержит файлы с именами

- а) **q.c**
- б) **qq.cpp**
- в) **qq.c**
- г) **q1.c1**
- д) **qaa.cmd**
- е) **q12.cpp**

Определите, в каком порядке будут показаны файлы, если выбрана сортировка по типу (по возрастанию).

- 1) **авгдбе** 2) **авгдеб** 3) **абвгде** 4) **авдбег**

Решение:

при сортировке по типу сравниваются расширения имен файлов

при сравнении используют коды символов

отсутствие символа (когда расширение закончилось) считается «меньше» любого символа, то есть, файл с расширением **.c** будет находиться в списке выше, чем файлы с расширениями **.c1** и **.cmd**

коды цифр размещаются в таблице символов раньше, чем коды букв, то есть, файл с расширением **.c1** будет находиться в списке выше, чем файл с расширением **.cmd**

теперь можно распределить имена файлов по расширениям

- а) **q.c**
- в) **qq.c**
- г) **q1.c1**
- д) **qaa.cmd**
- б) **qq.cpp**
- е) **q12.cpp**

осталась еще одна проблема – решить, что делать, если расширения совпадают; в этом случае в большинстве программ для определенности используется дополнительная сортировка по имени, поэтому файл с именем **q12.cpp** будет стоять в списке выше, чем файл **qq.cpp** (код цифры ‘1’ меньше, чем код буквы ‘q’)

в итоге получаем

- а) **q.c**
- в) **qq.c**
- г) **q1.c1**
- д) **qaa.cmd**
- е) **q12.cpp**
- б) **qq.cpp**

таким образом, мы доказали, что правильный ответ – 2.

Возможные ловушки и проблемы:

можно забыть правильно расставить имена файлов с одинаковыми расширениями (неверный ответ 1)

нельзя сравнивать числовые значения: например, интуитивно кажется, что файл с расширением **.c10** «больше», чем файл с расширением **.c2**, однако это неверно, потому что код цифры '2' больше, чем код цифры '1'; поэтому файл с расширением **.c10** будет стоять в списке выше файла с расширением **.c2** (при сортировке по типу в порядке возрастания)

можно забыть, что отсутствие кода (имя или расширение закончилось) «меньше» любого кода

можно забыть, что коды цифр меньше, чем коды букв

очень легко по невнимательности выбрать не тот ответ

Выполнение практических заданий.

Задание 1. Найди лишнее

- 1) Текстовый процессор, Microsoft Access, графический редактор, отладчик, переводчик;
- 2) Операционная система, архиваторы, табличный процессор, антивирусная программа;
- 3) Отладчик, транслятор, языки программирования, программа-оболочка.

Задание 2. Сравни чего больше:

- 1) Графических редакторов или прикладных программ?
- 2) Антивирусных программ или системных программ?
- 3) Отладчиков или языков программирования?

Задание 3. Укажите, какое ПО необходимо людям в следующих ситуациях (заполните таблицу):

Таблица 14

Ситуация	Системное ПО	Прикладное ПО	Инструментальное ПО
Создание проекта городского ландшафта			
Программист пишет компьютерную программу			
Подготовка фотоальбома к выпускному вечеру			
Подготовка Web сайта			

Ситуация	Системное ПО	Прикладное ПО	Инструментальное ПО
Ребенок играет в компьютерную игру			
Авиаконструктор создает модель самолета			
Архитектор создает проект нового здания			
Учитель пишет компьютерный тест по предмету			
В рекламном агентстве создается рекламный ролик			

Задание 4. Найди и исправь ошибки в предложениях:

1. Самые первые программы разрабатывались непосредственно на объектно-ориентированном языке и все люди могли заниматься их разработкой, а не только профессиональные программисты.
2. Программа - это последовательность команд, которую выполняет компьютер в процессе хранения информации.
3. Все используемые в компьютере программы называются аппаратным обеспечением.
4. Прикладное ПО руководит слаженной работой всех элементов компьютерной системы как на аппаратном уровне, так и на программном уровне.
5. Системные программы - это средства, предназначенные для создания ПО.

Задание 5. Впиши пропущенные слова в предложениях:

1. Компьютер без.....- это бесполезная, груда железа. И только.....делают его нашим помощником, другом, советчиком.
2. Программирование —деятельность человека по.....программы.
3. Для обработки информации на компьютере необходимо иметь— устройства компьютерной системы («hardware»), но

и.....(«software»), которое наделяет компьютер мыслями и интеллектом.

4.....ПО - предназначено для выполнения конкретных задач пользователя.

5. Наиболее дружелюбно пользователю.....ПО.

Задание 6. Укажите полное имя следующих файлов:

1) Petrov.txt; 2) Andreev.doc; 3) Sidorov. jpg

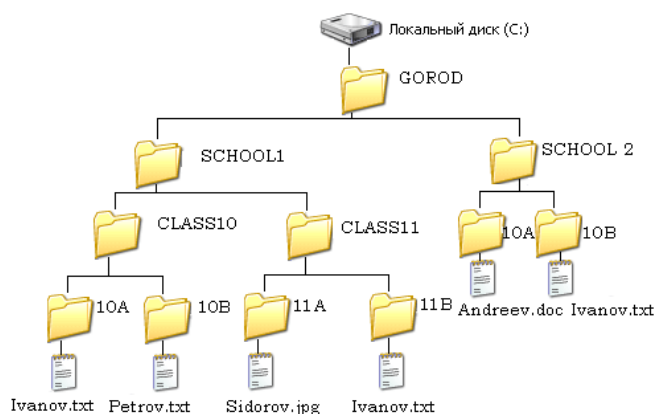


Рисунок 51

Задание 7. Постройте дерево каталогов

C:\Рисунки\Природа\Небо.bmp

C:\Рисунки\Природа\Снег.bmp

C:\Рисунки\Компьютер\Монитор.bmp

C:\Мои документы\Доклад.doc

Задание 8. В некотором каталоге хранился файл Задача5. После того, как в этом каталоге создали подкаталог и переместили в созданный подкаталог файл Задача5, полное имя файла стало E:\Класс9\Физика\Задачник\Задача5. Запишите полное имя этого файла до перемещения?

Задание 9. В некотором каталоге хранился файл Дневник.txt. В этом

каталоге создали подкаталог и переместили в созданный подкаталог файл Дневник.txt. Запишите полное имя файла после перемещения.

Задания для самостоятельной работы (по вариантам):

Для групповых операций с файлами используются маски имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один произвольный символ. Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность. Определите, по какой из масок будет выбрана указанная группа файлов:

1234.xls

23.xml

234.xls

23.xml

1) *23*.?x* 2) ?23?.x?? 3) ?23?.x* 4) *23*.???

Для групповых операций с файлами используются маски имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один произвольный символ. Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

В каталоге находятся пять файлов:

fort.docx

ford.docx

lord.doc

orsk.dat

port.doc

Определите, по какой из масок будет выбрана указанная группа файлов:

fort.docx

ford.docx

lord.doc

port.doc

- 1) ***o?*.d?*** 2) **?o*?.d*** 3) ***or*.doc?** 4) **?or?.doc?**

Для групповых операций с файлами используются маски имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один произвольный символ. Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность. Определите, какое из указанных имен файлов удовлетворяет маске: **?a???***

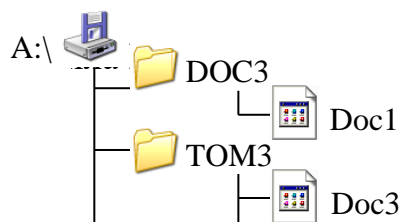
- 1) **dad1** 2) **dad22** 3) **3daddy** 4) **add444**

В некотором каталоге хранился файл **Задача5**. После того, как в этом каталоге создали подкаталог и переместили в созданный подкаталог файл **Задача5**, полное имя файла стало **Е:\Класс9\Физика\Задачник\Задача5**. Каково было полное имя этого файла до перемещения?

- 1) **Е:\Физика\Задачник\Задача5**
2) **Е:\Физика\Задача5**
3) **Е:\Класс9\Задачник\Задача5**
4) **Е:\Класс9\Физика\Задача5**

Дано дерево каталогов. Определите полное и

- 1) **А:\DOC3**
2) **А:\DOC3\Doc3**
3) **А:\DOC3\Doc1**
4) **А:\ТОМ3\Doc3**



В некотором каталоге хранится файл **Список_литературы.txt**. В этом

каталоге создали подкаталог с именем **10_CLASS** и переместили в него файл **Список_литературы.txt**. После чего полное имя файла стало **D:\SCHOOL\PHYSICS\10_CLASS\Список_литературы.txt**.

Каково полное имя каталога, в котором хранился файл до перемещения?

- 1) **D:\SCHOOL\PHYSICS\10_CLASS**
- 2) **D:\SCHOOL\PHYSICS**
- 3) **D:\SCHOOL**
- 4) **SCHOOL**

Пользователь, перемещаясь из одного каталога в другой, последовательно посетил каталоги **LESSONS, CLASS, SCHOOL, D:\ , MYDOC, LETTERS**. При каждом перемещении пользователь либо спускался в каталог на уровень ниже, либо поднимался на уровень выше. Каково полное имя каталога, из которого начал перемещение пользователь?

- 1) **D:\MYDOC\LETTERS**
- 2) **D:\SCHOOL\CLASS\LESSONS**
- 3) **D:\LESSONS\CLASS\SCHOOL**
- 4) **D:\LESSONS**

В некотором каталоге хранится файл **Задачи_по_программированию.txt**. В этом каталоге создали подкаталог и переместили в него файл **Задачи_по_программированию.txt**. После этого полное имя файла стало

D:\INFORM\LESSONS\10_CLASS\Задачи_по_программированию.txt

Каково полное имя каталога, в котором хранился файл до перемещения?

- 1) **D:\INFORM**
- 2) **D:\INFORM\LESSONS**
- 3) **10_CLASS**
- 4) **LESSONS\10_CLASS**

Учитель работал в каталоге **D:\Материалы к урокам\10 класс\Практические работы**.

Затем перешел в дереве каталогов на уровень выше, спустился в подкаталог **Лекции** и удалил из него файл **Введение**. Каково полное имя файла, который удалил преподаватель?

- 1) **D:\Материалы к урокам\10 класс\Введение**
- 2) **D:\Материалы к урокам\10 класс\Лекции\Введение**
- 3) **D:\Материалы к урокам\Лекции\Введение**
- 4) **D:\Материалы к урокам\Введение\Лекции**

В некотором каталоге хранится файл **Список_10_класса.txt**. В этом каталоге создали подкаталог и переместили в него файл **Список_10_класса.txt**, после чего полное имя файла стало

D:\USER\CLASS\DOC\Список_10_класса.txt.

Каково полное имя каталога, в котором хранился файл до перемещения?

- 1) **D:\USER\CLASS**
- 2) **DOC**
- 3) **D:\USER\CLASS\DOC**
- 4) **CLASS**

Пользователь, перемещаясь из одного каталога в другой, последовательно посетил каталоги **ACADEMY**, **COURSE**, **GROUP**, **E:**, **PROFESSOR**, **LECTIONS**. При каждом перемещении пользователь либо спускался в каталог на уровень ниже, либо поднимался на уровень выше. Каково полное имя каталога, из которого начал перемещение пользователь?

- 1) **E:\PROFESSOR\LECTIONS\ACADEMY**
- 2) **E:\ACADEMY\COURSE\GROUP**
- 3) **E:\ACADEMY**
- 4) **E:\GROUP\COURSE\ACADEMY**

Каталог содержит файлы с именами

- а) **p5.pas**
- б) **p4.ppt**
- в) **p12.pas**
- г) **pq.p**

д) **pq.pas**

е) **p12.ppt**

Определите, в каком порядке будут показаны файлы, если выбрана сортировка по типу (по возрастанию).

1) **вадгб** 2) **гавдбе** 3) **вадгбе** 4) **гвадеб**

В некотором каталоге хранится файл **Шпора.txt**. В этом каталоге создали подкаталог и переместили в него файл **Шпора.txt**. После этого полное имя файла стало

D:\Документы\Физика\Контрольная\Шпора.txt

Каково полное имя файла до перемещения?

1) **D:\Документы\Контрольная\Шпора.txt**

2) **D:\Физика\Шпора.txt**

3) **D:\Документы\Физика\Шпора.txt**

4) **D:\Физика\Контрольная\Шпора.txt**

В некотором каталоге хранится файл **Шпора.txt**. В этом каталоге создали подкаталог и переместили в него файл **Шпора.txt**. После этого полное имя файла стало

D:\Документы\Химия\Контрольная\Шпора.txt

Каково полное имя каталога, в котором хранился файл до перемещения?

1) **D:\Документы\Химия\Контрольная**

2) **D:\Документы\Химия**

3) **D:\Документы**

4) **D:**

Полное имя файла было **C:\Задачи\Физика.C**. Его переместили в каталог **Tasks** корневого каталога диска D. Каково полное имя файла после перемещения?

1) **D:\Tasks\Физика.C**

2) **D:\Tasks\Физика.D**

3) **D:\Задачи\Tasks\Физика.C**

4) **D:\Tasks\Задачи\Физика.C**

Для групповых операций с файлами используются маски имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один произвольный символ. Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность. Определите, какое из указанных имен файлов удовлетворяет маске: **?ba*r.?xt**

- 1) **bar.txt** 2) **obar.txt** 3) **obar.xt** 4) **barr.txt**

Находясь в корневом каталоге только что отформатированного диска, ученик создал 3 каталога. Затем в каждом из них он создал еще по 4 каталога. Сколько всего каталогов оказалось на диске, включая корневой?

- 1) 12 2) 13 3) **15** 4) **16**

Для групповых операций с файлами используются маски имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один произвольный символ. Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность. Определите, какое из указанных имен файлов удовлетворяет маске: **F??tb*.d?***

- 1) **Fructb.d** 2) **Football.ddd** 3) **Football.mdb** 4) **Futbol.doc**

Для групповых операций с файлами используются маски имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один произвольный символ. Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность. Определите, какое из указанных имен файлов удовлетворяет маске: **A?ce*s.m***

- 1) **Acess.md** 2) **Accesst.dbf** 3) **Access.mdb** 4) **Akcces.m1**

Для групповых операций с файлами используются маски имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один произвольный символ. Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность. Определите, какое из указанных имен файлов не удовлетворяет маске: **??pri*.***

- 1) **caprika.wow** 2) **weprik.cpp** 3) **otopri.c** 4) **reprint.be**

Для групповых операций с файлами используются маски имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один произвольный символ. Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность. Определите, какое из указанных имен файлов не удовлетворяет маске: **sys??.***

- 1) **system** 2) **sys23.exe** 3) **system.dll** 4) **syszx.problem**

Для групповых операций с файлами используются маски имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один произвольный символ. Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность. Определите, какое из указанных имен файлов не удовлетворяет маске: **?ell*.***

- 1) **yell.ow** 2) **fellow.ra** 3) **tell_me.tu** 4) **bell.lab**

Для групповых операций с файлами используются маски имен файлов. Маска представляет собой последовательность букв, цифр и прочих

допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один произвольный символ. Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность. Определите, какое из указанных имен файлов не удовлетворяет маске: **?*di.t?***

- 1) **poydi.t** 2) **pogudi.tanx** 3) **2di.t9** 4) **melodi.theme**

Для групповых операций с файлами используются маски имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один произвольный символ. Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность. Определите, по какой из масок будет выбрана указанная группа файлов

make.cpp, name.c, age.pc, name.cpp

- 1) ***a*e.?** 2) **a?e.*** 3) ***a?e.*** 4) **?a?e.***

Для групповых операций с файлами используются маски имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один произвольный символ. Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность. Определите, какое из указанных имен файлов удовлетворяет маске: **re*_a?09.do?**

- 1) **refa-09.doc** 2) **ref_a_09.do** 3) **refe_a09.doc** 4) **ref_a_09.dot**

Для групповых операций с файлами используются маски имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один

произвольный символ. Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность. Определите, какое из указанных имен файлов удовлетворяет маске: **d?cf*.jp*g**

- 1) **dscf3456.jpeg** 2) **dcf1234.jpg** 3) **dsscf6754.jpg** 4) **dcsf1111.jpeg**

Для групповых операций с файлами используются маски имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы: Символ «?» (вопросительный знак) означает ровно один произвольный символ. Символ «*» (звездочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность. Определите, какое из указанных имен файлов удовлетворяет маске: **?n*i??t?*.?i**

- 1) **annimtion.jpj** 2) **animate.gif** 3) **animation.gi** 4) **a_nimation.gi**

СПИСОК ЛИТЕРАТУРЫ

1. Антопольский, А.Б. Информационные ресурсы России: Научно-методическое пособие / А.Б. Антопольский. — М.: Либерия, 2014. — 424 с.
2. Антошин, М.К. Учимся рисовать на компьютере / М.К. Антошин. — М.: Айрис, 2016. — 160 с.
3. Волкова В.Н. Теория информационных процессов и систем. — М.: Юрайт, 2016. — 504 с.
4. Горячев, А.В. Информатика в играх и задачах. 2 класс. Учебник-тетрадь / А.В. Горячев, Т.О. Волкова, К.И. Горина, и др.. — М.: Баласс, 2015. — 128 с.
5. Горячев, А.В. Практикум по информационным технологиям / А.В. Горячев, Ю.А. Шафрин. — М.: Бином, 2016. — 272 с.
6. Информатика. Базовый курс / Под ред. С.В. Симоновича. — СПб.: Питер, 2015. — 640 с.
7. Информационные технологии в экономике управления / Под ред. В.В. Трофимова. — М.: Юрайт, 2015. — 482 с.
8. Каймин, В.А. Информатика: практикум на ЭВМ / В.А. Каймин, Б.С. Касаев. — М.: ИНФРА-М, 2016. — 216 с.
9. Ляхович В.Ф., Молодцов В.А., Рыжикова Н.Б. Основы информатики. — М.: КноРус, 2016. — 348 с.
10. Ляхович, В.Ф. Информатика 10-11 кл / В.Ф. Ляхович. — М.: Просвещение, 2015. — 352 с.
11. Просветов Г.И. Анализ данных с помощью Excel. Задачи и решения. — М.: Альфа-Пресс, 2015. — 160 с.
12. Семакин, И. Информатика. Базовый курс. 7-9 классы / И. Семакин, Л. Залогова, С. Русаков. — М.: Бином. Лаборатория знаний, 2015. — 390 с.
13. Семакин, И.Г. Информатика 10 класс / И.Г. Семакин, Е.К. Хеннер. — М.: Бином. Лаборатория знаний, 2017. — 165 с.

14. Угринович, Н. Информатика и информационные технологии / Н. Угринович. — М.: Бином. Лаборатория знаний, 2017. — 512 с.
15. Шестакова Информатика и информационно-коммуникационные технологии. Базовый курс. 8 класс / Шестакова, Л.В. и. — М.: Бином, 2017. — 176 с.

Еремина Ирина Ильинична, кандидат педагогических наук, доцент, доцент кафедры бизнес-информатики и математических методов в экономике Набережночелнинского института ФГАОУ ВО «Казанский (Приволжский) федеральный университет», профессор РАЕ, имеет почетное звание "Заслуженный работник науки и образования"

Савицкий Сергей Константинович, кандидат педагогических наук, доцент, доцент кафедры электроэнергетики и электротехники, заместитель заведующего отделением информационных технологий и энергетических систем по воспитательной работе Набережночелнинского института ФГАОУ ВО «Казанский (Приволжский) федеральный университет»

Умаров Марат Файзуллаевич, заместитель директора по общим вопросам Набережночелнинского института ФГАОУ ВО «Казанский (Приволжский) федеральный университет»

Хаустов Сергей Леонидович, кандидат педагогических наук, доцент, доцент кафедры технологии и общетехнических дисциплин Стерлитамакского филиала ФГБОУ ВО «Башкирский государственный университет», отличник образования Республики Башкортостан

**Информатика и информационные технологии.
Экономическая информатика.
Лабораторный практикум
Часть 1
(серия «Бакалавриат»)
для студентов вузов**



В авторской редакции

ПОДПИСАНО В ПЕЧАТЬ 05.04.2019г.
ФОРМАТ 60X84/16 БУМАГА ОФСЕТНАЯ ПЕЧАТЬ РИЗОГРАФИЧЕСКАЯ
УЧ.-ИЗД.Л. 7,9375 УСЛ.-ПЕЧ.Л. 7,9375 ТИРАЖ 500 ЭКЗ.
ЗАКАЗ 050419-1
АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ «ЦЕНТР ПОДДЕРЖКИ
ПРОГРАММ РАЗВИТИЯ КАЗАНСКОГО ФЕДЕРАЛЬНОГО УНИВЕРСИТЕТА»
РФ, РТ, Г.КАЗАНЬ УЛ.САЛИХА САЙДАШЕВА, Д.12 ОФ.13