

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**УНИВЕРСИТЕТ ИТМО**

**И.А. Хахаев**

**ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ, СЕТИ  
И СИСТЕМЫ ТЕЛЕКОММУНИКАЦИЙ  
В ТАМОЖЕННОМ ДЕЛЕ**

**Учебное пособие**

 **УНИВЕРСИТЕТ ИТМО**

**Санкт-Петербург**

**2015**

Хахаев И.А. Вычислительные машины, сети и системы телекоммуникаций в таможенном деле: учебное пособие. – СПб: Университет ИТМО, 2015. – 86 с.

В пособии рассматриваются базовые понятия вычислительных систем и сетей телекоммуникаций, а также современные технологии организации вычислительных систем (в том числе высокопроизводительных) и особенности их использования в Единой Автоматизированной Информационной системе (ЕАИС) ФТС России. Пособие акцентирует внимание на отдельных существенных аспектах дисциплины и не заменяет основную учебную литературу по курсу. После учебно-информационных разделов издания предлагаются задания для самостоятельной работы.

Пособие подготовлено на кафедре «Таможенного дела и логистики».

Учебное пособие разработано в соответствии с программой дисциплины «Вычислительные машины, сети и системы телекоммуникаций в таможенном деле» и предназначено для студентов, обучающихся по специальности 38.05.02 (036401) «Таможенное дело», для использования при подготовке семинарских занятий, курсовых проектов, отчетов по практике, дипломных работ.

Рекомендовано к печати Учёным советом факультета «ИМБИП», протокол № 8 от 14.04.2015г.

**Университет ИТМО** – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2015

© Хахаев И.А., 2015

## Содержание

Введение.....	5
Логические основы построения вычислительных систем.....	5
Высказывания. Истинность повествовательных предложений.....	5
Логические переменные и функции.....	5
Основные логические функции.....	6
Таблицы истинности.....	6
Сложные функции и порядок действий.....	7
Законы логики.....	8
Теоремы логики (Булевы теоремы).....	8
Реализация логики в вычислительных системах.....	9
Задания для самостоятельной работы.....	16
Основные сведения о вычислительных машинах и системах.....	16
Основные определения.....	16
Принципы организации (архитектуры) вычислительных систем.....	17
Модель представления данных в ЭВМ.....	19
Ячейки и регистры памяти.....	19
Карта памяти.....	19
Представление чисел и символов в ЭВМ.....	21
Представление символов (букв). Кодировочные таблицы (кодировки)..	22
Типы данных и их особенности.....	23
Задания для самостоятельной работы.....	26
Многоуровневая организация ВМ.....	26
Варианты разделения на уровни.....	26
Языки программирования.....	26
Вычислительная система как открытая система.....	28
Понятие открытой системы.....	28
Спецификации POSIX.....	28
Модель OSI/RM.....	29
Протоколы и интерфейсы.....	30
Разделение ресурсов вычислительных систем.....	31
Аппаратные средства ЭВМ: основные устройства.....	31
Облачные вычисления.....	35
Задания для самостоятельной работы.....	37
Операционные системы.....	38
Многозадачность в операционных системах.....	40
Задания для самостоятельной работы.....	40
Внешняя память и файловые системы.....	41
Лицензирование операционных систем.....	44
Обеспечение производительности и надежности вычислительных систем	48
Понятие производительности.....	48

Оценки производительности.....	48
Факторы, влияющие на производительность.....	49
Тесты SPEC.....	49
Кластерные технологии.....	49
Состав кластера.....	50
Задания для самостоятельной работы.....	57
Компьютерные сети.....	58
Понятие, назначение, состав.....	58
Топологии компьютерных сетей.....	58
Серверы сети.....	61
Авторизация пользователей сети.....	62
Адресация в компьютерных сетях.....	63
Глобальная логическая адресация DNS.....	67
Прикладной уровень TCP/IP.....	70
Магистральные сети коммуникаций.....	73
Оборудование для организации сетей.....	75
Задания для самостоятельной работы.....	77
Беспроводные сети.....	79
Задания для самостоятельной работы.....	81
Пример оснащения таможни.....	82
Литература.....	83

# Введение

Данное учебно-методическое издание предназначено для студентов специальности 38.05.02 (036401) «Таможенное дело» в качестве базового пособия по теоретической части курса. Оно акцентирует внимание на отдельных существенных аспектах дисциплины и не заменяет основную учебную литературу по курсу «Вычислительные машины, сети и системы телекоммуникаций в таможенном деле».

Материал, изложенный в пособии, подразумевает наличие знаний, полученных в ходе изучения дисциплины «Информатика».

В пособии раскрываются основы организации вычислительных машин и систем, компьютерных сетей и вычислительных комплексов с повышенной надёжностью и производительностью. Также обращается внимание на особенности применения вычислительных систем и сетей телекоммуникаций в Единой автоматизированной информационной систем таможенных органов (ЕАИС ТО). Материал излагается в тезисной (конспективной) форме и является основой для дальнейшей проработки на семинарских занятиях и в ходе самостоятельной работы.

## Логические основы построения вычислительных систем

### Высказывания. Истинность повествовательных предложений

**Высказывание** – основной объект математической логики.

Высказыванием называется повествовательное предложение (утверждение), которое может быть **истинным** или **ложным**.

Примеры высказываний:

- «Идёт дождь»
- «Вася Пупкин – девочка»

Истинность или ложность высказывания – **логическая переменная**.

### Логические переменные и функции

Значение логической переменной «истина» обозначается как (логическая) 1, «ложь» – как (логический) 0.

Другие обозначения: «Истина» – t или T (true), «Ложь» – f или F (false).

Логические переменные могут обозначаться по-разному: A, B, C ... или  $X_1, X_2 \dots$ .

**Логическая функция** – операция с логическими переменными (одной или несколькими).

Результат логической функции – тоже логическая переменная.

## Основные логические функции

- **Логические константы.** True(A,B) – всегда «истина», False(A,B) – всегда «ложь».

- **Отрицание.** Обозначается  $\neg A$  или  $\bar{A}$  ( $B = \neg A$ ) (логическое НЕ). Логическая переменная принимает противоположное значение.

- **Дизъюнкция.**  $C = A \vee B$  или  $C = A \cup B$  (логическое ИЛИ). Принимает значение «истина», когда хотя бы одно из высказываний (один из аргументов функции) – «истина».

- **Конъюнкция.**  $C = A \wedge B$  или  $C = A \cap B$  (логическое И, &). Принимает значение «истина», когда все высказывания (все аргументы функции) – «истина».

## Импликация и эквиваленция

Импликативное высказывание: если A то B ( $A \Rightarrow B$ ).

Иначе говоря, «A влечёт B», «B при условии, что A», где A – предпосылка, B – следствие.

Импликация принимает значение «ложь», когда A – «истина», а B – «ложь». Во всех остальных случаях импликация имеет значение «истина».

Эквивалентные высказывания: A тогда и только тогда, когда B ( $A \Leftrightarrow B$ ).

Иначе говоря, «если A, то B, и если B, то A», «A есть необходимое и достаточное условие для B».

Эквиваленция принимает значение «истина», когда A и B имеют одинаковые значения. Во всех остальных случаях эквиваленция имеет значение «ложь».

## Таблицы истинности

**Таблица истинности** – способ определения логической функции. Ниже приведена таблица истинности основных логических функций.

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

## Сложные функции и порядок действий

Сложная функция строится из основных. Таблица истинности создаётся с учётом порядка действий. Для изменения порядка действия при необходимости используются скобки.

Порядок действий (приоритет операций):

1. Отрицание
2. Дизъюнкция и конъюнкция
3. Импликация
4. Эквиваленция.

При определении значений сложной логической функции необходимо построить таблицу истинности в соответствии с приоритетом операций при всех возможных комбинациях значений аргументов.

Для примера рассмотрим функцию

$$F(A, B, C) = (A \vee B) \wedge (B \Rightarrow \neg C) .$$

Данную функцию можно представить как  $F = F1 \wedge F2$ , где  $F1 = A \vee B$ , а  $F2 = B \Rightarrow \neg C$ .

Следовательно, при определении таблицы истинности для функции F следует сначала определить значения функций F1 и F2, а затем применить к этим значениям логическую операцию конъюнкции.

Результат показан в следующей таблице.

A	B	C	$F1 = A \vee B$	$F2 = B \Rightarrow \neg C$	$F = F1 \wedge F2$
0	0	0	0	1	0
0	0	1	0	1	0
0	1	0	1	1	1
0	1	1	1	0	0
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	0	0

Часто используется логическая операция *сумма по модулю 2*, иначе – **исключающее ИЛИ (XOR)**. Ниже показана таблица истинности для XOR

A	B	XOR(A,B)
0	0	0
0	1	1
1	0	1
1	1	0

## Законы логики

Высказывание, истинное при любых значениях его компонентов, называется **тождественно истинным** или **тавтологией**.

Высказывание, ложное при любых значениях его компонентов, называется **тождественно ложным** или **противоречием**.

Основные законы формальной логики в кратких формулировках и формальной записи приведены в следующей таблице.

	Название	Пример формулировки	Формальная запись
1.	Закон исключённого третьего	Истинным может быть либо высказывание «А», либо его противоположность	$A \vee \neg A$
2.	Закон противоречия	Высказывание и его отрицание не могут быть одновременно истинными	$\neg(A \wedge \neg A)$
3.	Закон силлогизма	Если первое влечет второе, и если второе влечет третье, то первое влечет третье	$[(A \Rightarrow B) \wedge (B \Rightarrow C)] \Rightarrow (A \Rightarrow C)$
4.	Закон двойного отрицания	Если неверно, что неверно «А», то «А» - верно	$A \Leftrightarrow \neg(\neg A)$
5.	Закон контрапозиции	Если первое влечет второе, то отрицание второго влечет отрицание первого	$(A \Rightarrow B) \Leftrightarrow (\neg B \Rightarrow \neg A)$

## Теоремы логики (Булевы теоремы)

При формальной записи основных логических теорем в качестве логических переменных будем использовать алгебраические обозначения (x, y и т. п.), а в качестве обозначений логических операций будем использовать символы сложения для дизъюнкции, умножения для конъюнкции и инверсии для отрицания.

В такой системе обозначений приведем формулировки основных логических теорем (без доказательств).

### Теоремы для одной переменной

- $x \cdot 0 = 0$

2.  $x \cdot 1 = x$
3.  $x \cdot x = x$
4.  $x \cdot \bar{x} = 0$
5.  $x + 0 = x$
6.  $x + 1 = 1$
7.  $x + x = x$
8.  $x + \bar{x} = 1$

### Теоремы для нескольких переменных

1. Коммутативность:  $x + y = y + x$  ;  $x \cdot y = y \cdot x$
2. Ассоциативность:  $x + (y + z) = (x + y) + z = x + y + z$  ;  
 $x \cdot (y \cdot z) = (x \cdot y) \cdot z = x \cdot y \cdot z$
3. Дистрибутивность:  $x \cdot (y + z) = x \cdot y + x \cdot z$  ;  
 $(w + x) \cdot (y + z) = w \cdot y + w \cdot z + x \cdot y + x \cdot z$
4.  $x + x \cdot y = x$
5.  $x + \bar{x} \cdot y = x + y$  ;  $\bar{x} + x \cdot y = \bar{x} + y$

### Теоремы де Моргана

1.  $\overline{(x + y)} = \bar{x} \cdot \bar{y}$
2.  $\overline{(x \cdot y)} = \bar{x} + \bar{y}$

## Реализация логики в вычислительных системах

### Построение логических схем

Сочетание математической логики и возможностей проводить вычисления в двоичной системе исчисления привело к возможности конструирования логических и вычислительных устройств на основе двух устойчивых состояний.

Дело в том, что **0** и **1** в логике не просто цифры, а обозначение состояний какого-то предмета нашего мира, условно называемых «ложь» и «истина». Таким предметом, имеющим два фиксированных состояния, может быть **электрический сигнал**. Устройства, фиксирующие два устойчивых состояния, называются бистабильными (например, выключатель, реле).

Наличие возможности переключения состояния электрического сигнала с помощью другого электрического сигнала как раз и явилось основной предпосылкой развития современной компьютерной техники.

Первые вычислительные машины были релейными.

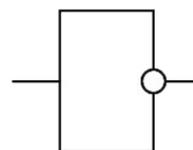
Затем электромеханические переключатели были заменены электронными переключателями – электронными лампами и полупроводниковыми транзисторами.

Позднее были созданы новые устройства управления электрическими сигналами – электронные схемы, состоящие из набора полупроводниковых элементов. Такие электронные схемы, которые преобразовывают сигналы только двух фиксированных напряжений электрического тока (бистабильные), стали называть **логическими элементами**.

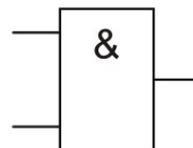
### Элементарные устройства логических схем

Логические элементы имеют один или несколько входов и один выход, через которые проходят электрические сигналы, обозначаемые условно как 0 при отсутствии электрического сигнала, и как 1 при наличии электрического сигнала.

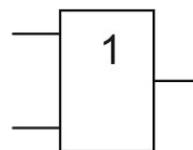
Простейшим логическим элементом является **инвертор**, выполняющий *функцию отрицания*. Если на вход поступает сигнал, соответствующий 1, то на выходе будет 0. И наоборот. У этого элемента один вход и один выход.



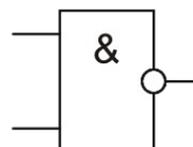
Логический элемент, выполняющий *логическое умножение*, называется **конъюнктор**. Он имеет, как минимум, два входа.



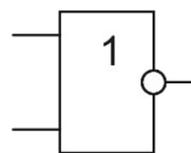
Логический элемент, выполняющий *логическое сложение*, называется **дизъюнктор**. Он имеет, как минимум, два входа.



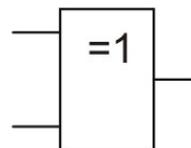
Логический элемент **И-НЕ** состоит из конъюнктора и инвертора.



Логический элемент **ИЛИ-НЕ** состоит из дизъюнктора и инвертора



Логический элемент **исключающее ИЛИ (XOR)**.



Логические элементы реализуются на комплементарных полевых транзисторах, сделанных по технологии «металл-окисел-полупроводник» или «металл-диэлектрик-полупроводник» (КМОП-транзисторы или КМДП-транзисторы). Комплементарными транзисторы называются потому, что они изготавливаются парами из транзисторов р-типа и n-типа.

Ниже приведены схемы основных логических элементов на КМОП-(КМДП)-транзисторах.

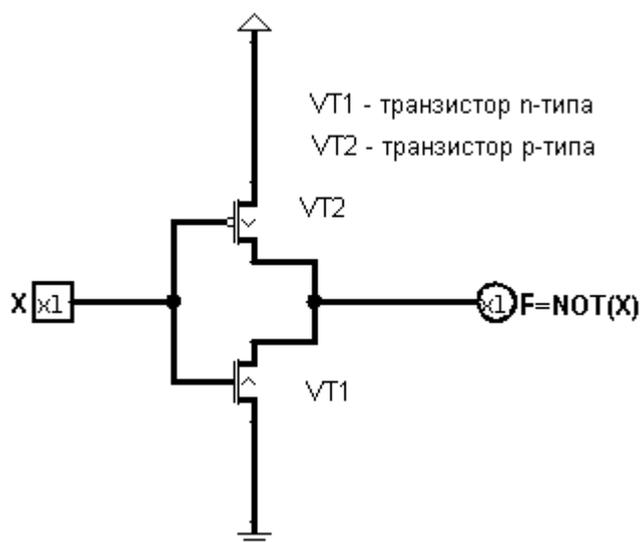


Рис. 1. Инвертор (логический элемент «НЕ») на комплементарной паре транзисторов

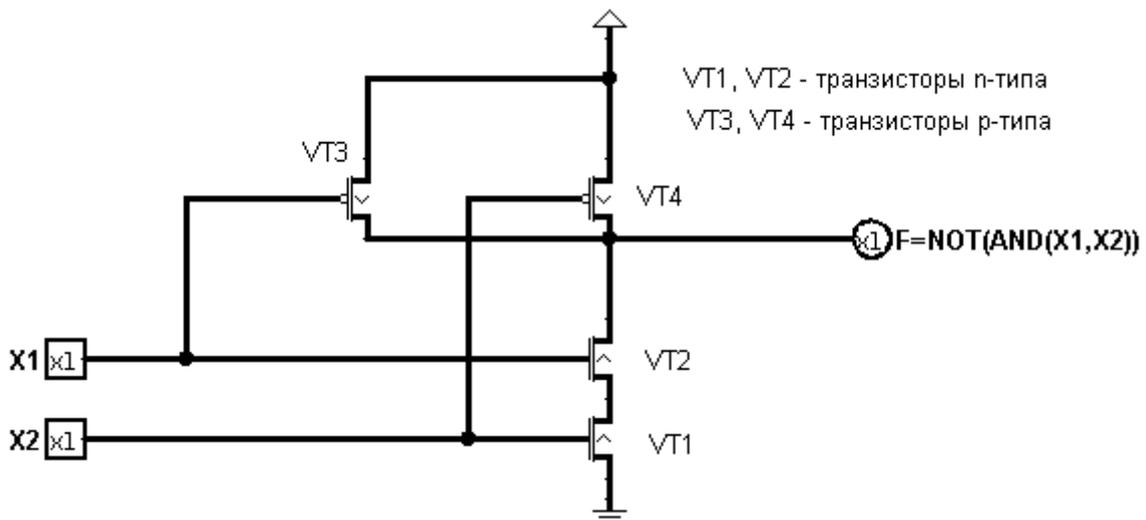


Рис. 2. Логический элемент «И-НЕ» на комплементарных парах транзисторов

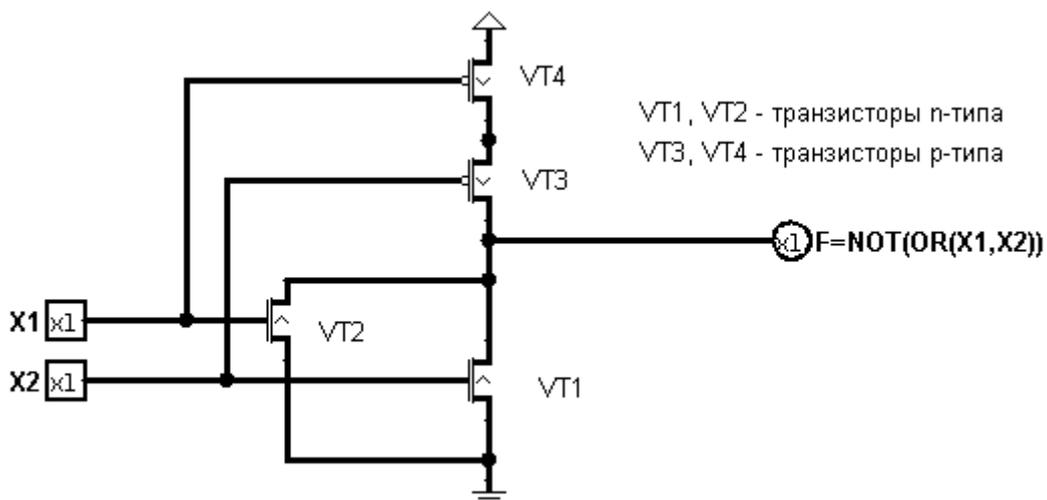


Рис. 3. Логический элемент «ИЛИ-НЕ» на комплементарных парах транзисторов

Для того, чтобы получить логические элементы «И» и «ИЛИ», нужно соединить выход (контакт F) элементов «И-НЕ» и «ИЛИ-НЕ» с входом (контактом X) элемента «НЕ».

Доказано, что любую логическую схему можно реализовать на элементах «И-НЕ» или на элементах «ИЛИ-НЕ». Выбранный тип элементов называют **базисом**. Ниже показана реализация логического элемента «Исключающее ИЛИ» на базисе «И-НЕ».

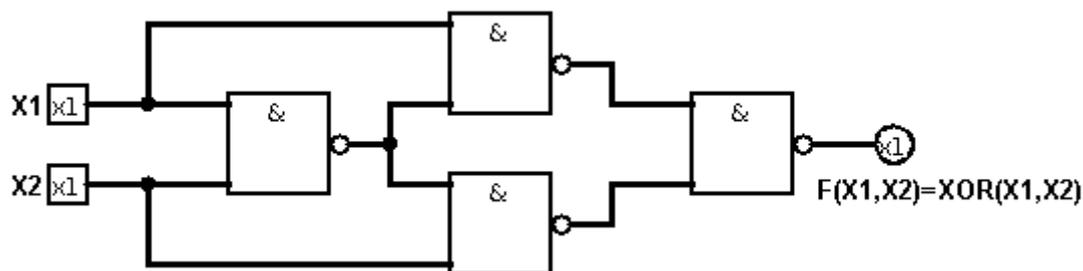


Рис. 4. Логический элемент «XOR» на ,базисе «И-НЕ»

## Моделирование логических схем

Современные программные средства вычислительных систем позволяют моделировать работу логических и цифровых схем. Средства моделирования помогают понять работу таких схем и построить таблицы истинности для сложных логических функций. Для моделирования логических и цифровых схем могут использоваться достаточно простые в понимании и освоении свободные программные пакеты QUCS (сайт проекта <http://qucs.sourceforge.net/>) и Logisim (сайт проекта на русском языке <http://www.cburch.com/logisim/ru/index.html>).

## Логические схемы как основа устройства вычислительной системы

На рис. 5 показана структурная схема простой современной вычислительной машины (компьютера).

Важнейшими элементами компьютера являются **триггеры, полусумматоры, сумматоры, шифраторы, дешифраторы, счетчики, регистры**, которые строятся из логических элементов.

Обработка любой информации на компьютере сводится к выполнению процессором различных арифметических и логических операций. Для этого в состав процессора входит так называемое **арифметико-логическое устройство (АЛУ)**.

**Триггер** – электронная схема, применяемая для хранения значения одного двоичного разряда.

Воздействуя на входы триггера, его переводят в одно из двух возможных состояний (0 или 1). С поступлением сигналов на входы триггера в зависимости от его состояния либо происходит переключение, либо исходное состояние сохраняется. При отсутствии входных сигналов триггер сохраняет свое состояние сколь угодно долго при наличии электропитания.

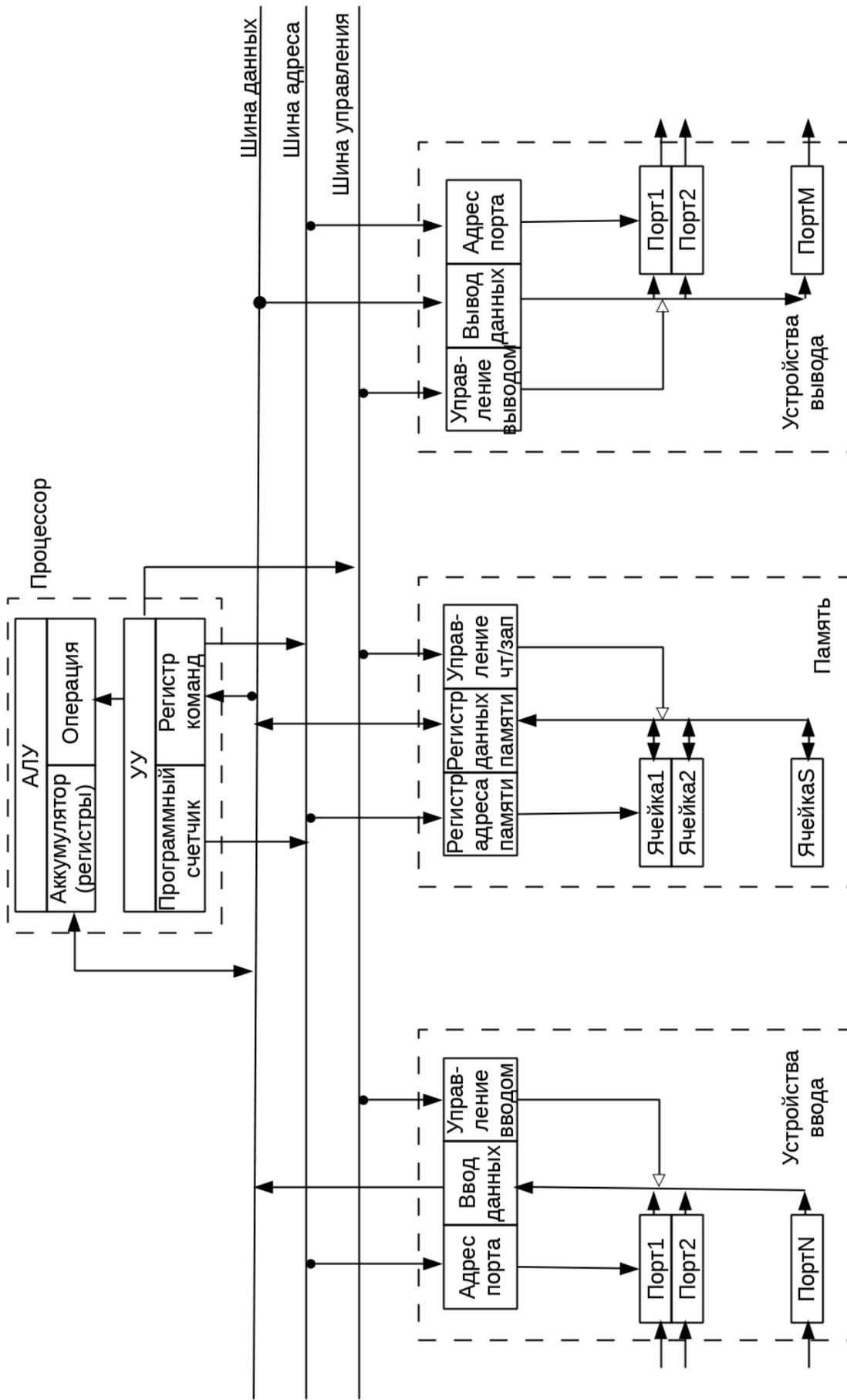


Рис. 5. Структурная схема простого компьютера

Простейший вариант триггера — RS-триггер — показан на рис. 6. Изменение состояния с 0 на 1 на входе S приводит к переводу состояния выхода Q в 1. Это состояние сохраняется в дальнейшем при любых изменениях состояния входа S (выход Q хранит значение 1) до тех пор, пока не изменится состояние входа R с 0 на 1. Тогда состояние выхода Q снова возвращается в 0 (происходит сброс состояния выхода Q). Для повторного перевода выхода Q в 1 необходимо снова установить состояние входа R в 0 и повторить цикл. Вход S называют «Set» (установка), вход R — соответственно, «Reset» (сброс). Состояние второго выхода ( $\bar{Q}$  или NOT(Q)) всегда является инверсией состояния выхода Q.

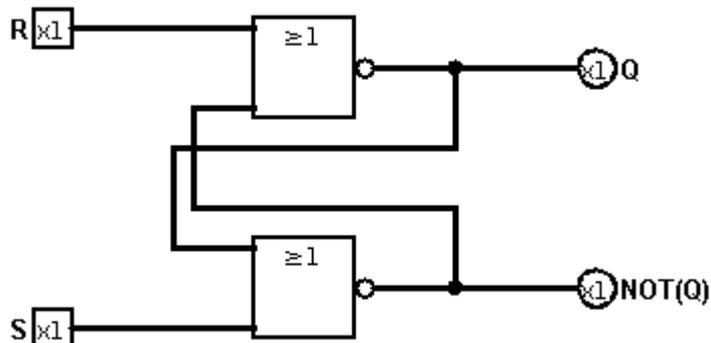


Рис. 6. RS-триггер на базисе «ИЛИ-НЕ»

Таким образом, использование триггеров позволяет организовать **энергозависимую** память. На практике основой такой памяти являются более сложные D-триггеры (рис. 7).

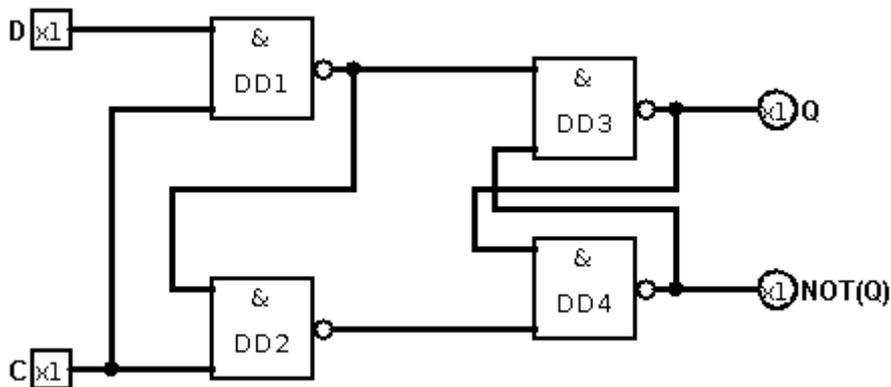


Рис. 7. D-триггер на базисе «И-НЕ»

В D-триггере изменение состояния выхода Q при изменении состояния входа D происходит только тогда, когда на вход C подается сигнал, соответствующий логической единице.

Нужно отметить, что сигнал на выходе триггера может инициировать **изменение состояния** какого-либо материала (носителя информации). Так получается **энергонезависимая память**.

## Задания для самостоятельной работы

Составить таблицы истинности для логических выражений в соответствии с вариантами заданий, а также составить логические схемы на указанных в вариантах заданий базисах

Варианты заданий:

1.  $F(A,B,C) = \text{AND}(\text{OR}(A, B), \text{OR}(\text{NOT}(B), C))$ , базис ИЛИ-НЕ
2.  $F(A,B,C) = \text{AND}(\text{NOT}(A), B, C, \text{NOT}(\text{OR}(A, B)))$ , базис И-НЕ
3.  $F(A,B,C) = \text{OR}(\text{AND}(A, C), \text{AND}(B, \text{NOT}(C)), \text{AND}(\text{NOT}(A), B))$ , базис И-НЕ
4.  $F(A,B,C) = \text{OR}(A, \text{NOT}(\text{AND}(\text{OR}(A, B), C)))$ , базис ИЛИ-НЕ
5.  $F(A,B,C,D) = \text{NOT}(\text{AND}(A, B, \text{NOT}(\text{OR}(C, D))))$ , базис И-НЕ
6.  $F(A,B,C) = \text{AND}(\text{NOT}(A), \text{NOT}(B), C)$ , базис ИЛИ-НЕ
7.  $F(A,B,C,D) = \text{AND}(\text{OR}(A, B), \text{OR}(C, D))$ , базис ИЛИ-НЕ
8.  $F(A,B,C,D) = \text{OR}(\text{AND}(A, B), \text{AND}(C, D))$ , базис И-НЕ
9.  $F(A,B,C) = \text{NOT}(\text{OR}(A, \text{AND}(B, \text{NOT}(C))))$ , базис ИЛИ-НЕ
10.  $F(A,B,C) = \text{AND}(A, B, \text{OR}(C, \text{NOT}(B)))$ , базис И-НЕ.

## Основные сведения о вычислительных машинах и системах

### Основные определения

**Вычислительная машина** – это электронное устройство, предназначенное для автоматического выполнения вычислительных (математических) операций.

**Вычислительная система** – это совокупность одной или нескольких вычислительных машин (компьютеров), *программного обеспечения* и *периферийного оборудования*, предназначенная для реализации информационно-вычислительных процессов.

**Архитектура вычислительной системы** – это логическая организация и структура аппаратных и программных ресурсов вычислительной системы.

**Шина** – это элемент конструкции, обеспечивающий *параллельное* соединение электрических и электронных устройств по проводам (проводникам, токоведущим линиям).

### **Вычислительная машина (компьютер, ЭВМ) как аппаратно-программный комплекс**

Хотя в потребительских качествах компьютера определяющую роль играет **программное обеспечение** (в том смысле, что тот же самый экземпляр компьютера с другими программами может обладать совершенно другими возможностями с позиции пользователя), для этих программ (в первую очередь для **системных**) требуется **среда функционирования**.

Средой функционирования **системных программ** являются **аппаратные средства** компьютера (ЭВМ).

Полностью разделить ПО и АС невозможно, т.к. эффективно работающее ПО использует особенности АС.

Поэтому ЭВМ рассматривается как **комплексная система** оборудования и работающих на этом оборудовании программ.

### **Принципы организации (архитектуры) вычислительных систем**

Вычислительная **система** строится из составных частей – **подсистем** (рис. 8).

- Подсистема выполнения логических и арифметических операций (вычислитель)
- Подсистема хранения программ и данных для вычислителя (память)
- Подсистема формирования программ и данных, отображения результатов вычислений (устройства ввода/вывода)

Принципиально отличаются две архитектуры: **гарвардская** и **принстонская (фон Неймана)**.

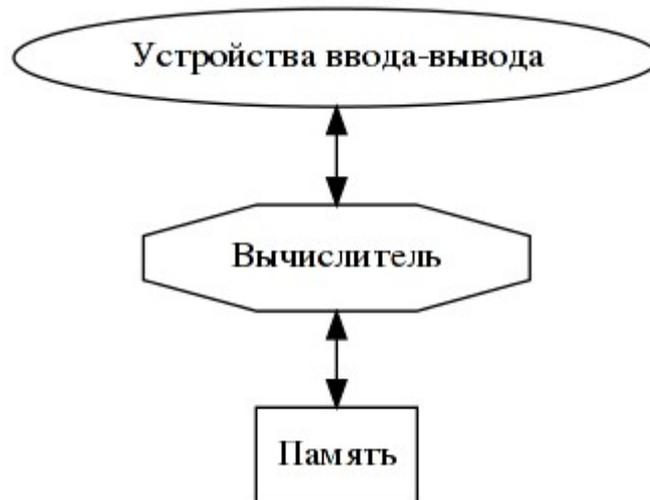


Рис. 8. Основные подсистемы вычислительной системы

В гарвардской архитектуре (рис. 9) разделены **память программ** и **память данных**, соответственно, разделены шины **инструкций** и данных. Это устраняет ошибки, связанные с нарушениями **распределения памяти**, но усложняет конструкцию.

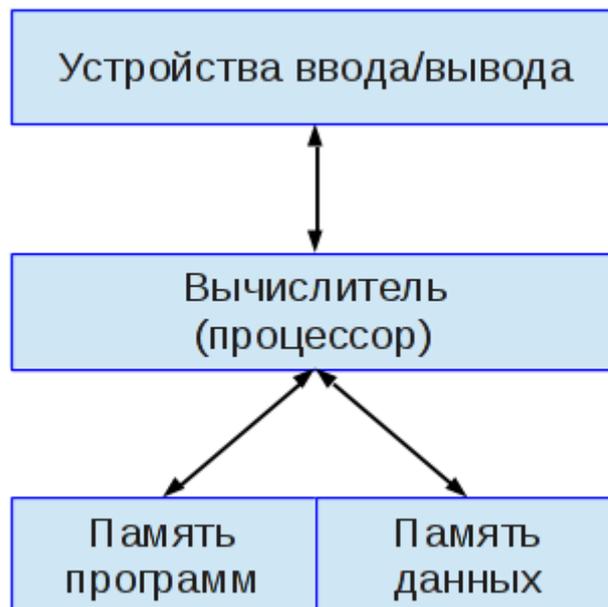


Рис. 9. Общая схема гарвардской архитектуры вычислительной системы

В архитектуре фон Неймана для данных и программ используется общая память (рис. 10). Устройство управления (УУ) и арифметико-логическое устройство (АЛУ) образуют вычислитель, традиционно называемый «Центральное Процессорное Устройство» (Central Processor Unit — CPU).

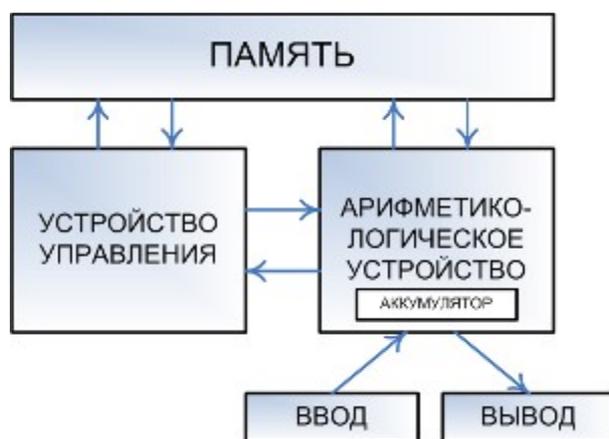


Рис. 10. Общая схема пристонской архитектуры вычислительной системы

Важнейшим принципом организации современных вычислительных машин и систем является **разделение программ и данных**. Следствием этого принципа является свойство **массовости алгоритмов**. При этом обеспечивается нужный уровень абстракции при составлении алгоритмов, когда в алгоритмах используются не конкретные значения, а **переменные**, которые означают данные различных типов.

## Модель представления данных в ЭВМ

### Ячейки и регистры памяти

Все программы и данные, с которыми работает вычислительная система, заносятся в память этой системы.

Память вычислительной системы представляется в виде ячеек, в каждой из которых может содержаться одна из цифр двоичной системы счисления – 0 или 1. Последовательность таких ячеек образует так называемый **регистр**. Длина регистра (количество ячеек) называется **длиной машинного слова** или **разрядностью** и в современных системах может быть **16, 32, 64** или **128** ячеек. Соответственно, существуют понятия **16-, 32-, 64-** или **128-**разрядных вычислительных систем (или архитектур).

Одна цифра двоичного числа соответствует одному биту информации.

Сейчас существует соглашение о том, что 8 бит равны одному байту информации.

### Карта памяти

Память компьютера представляется как **нумерованный набор регистров**, номера регистров называются **адресами памяти** и

записываются в шестнадцатеричном коде (в виде чисел шестнадцатеричной системы счисления).

Количество возможных адресов определяется разрядностью системы и способом указания адресов (адресации).

Пример карты памяти абстрактной 16-ти разрядной системы показан на рис. 11.

Самый правый разряд регистра называется **младшим битом**, самый левый, соответственно – **старшим битом**. Кроме того, выделяют старший и младший байты.

Регистры условно нумеруются снизу вверх, номера регистров записываются в шестнадцатеричной системе.

Для того, чтобы в вычислительной системе можно было различить, какие наборы битов следует интерпретировать как числа, какие – как соответствующие символам (буквам), а какие представляют из себя команды, вводится понятие **типа данных**.

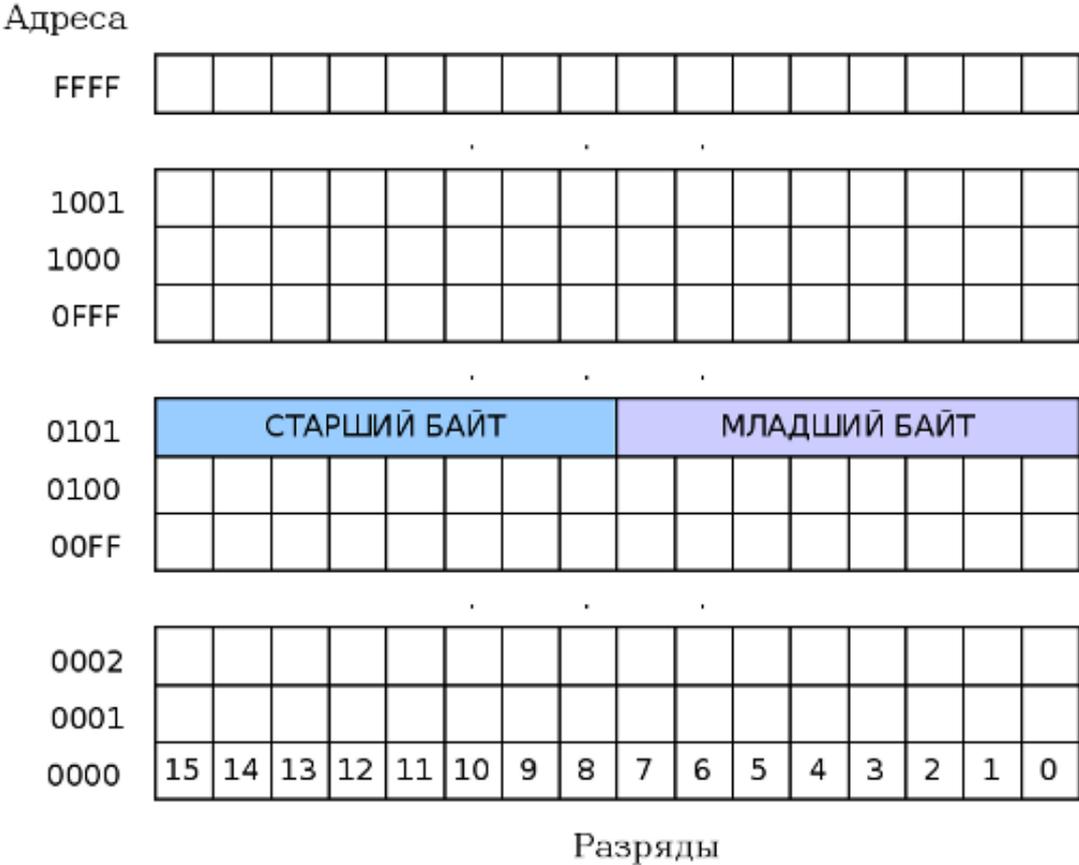


Рис. 11. Карта памяти абстрактной 16-ти разрядной вычислительной системы

# Представление чисел и символов в ЭВМ

## Представление чисел

Самый простой вариант чисел для представления в памяти ЭВМ — натуральные числа.

Натуральное число – это целое положительное число.

Оно занимает полностью весь диапазон разрядов (регистр).

Поэтому возможный диапазон натуральных чисел – от **0** до  $2^N-1$ , где **N** – разрядность системы.

Пример представления числа 123: **0000000001111011**

Представление целых чисел уже несколько сложнее.

Для целых чисел необходимо указывать знак (положительное или отрицательное), поэтому на указание знака отводится один бит – старший бит старшего байта. Установка этого бита в 1 означает отрицательное число.

Остальные **N-1** разрядов обеспечивают представление модуля числа.

Таким образом, диапазон целых чисел в системе с разрядностью **N** – от  $-2^{N-1} + 1$  до  $2^{N-1} - 1$ .

Пример представления числа -123: **1000000001111011**

Еще более сложным является представление вещественных (действительных) чисел (чисел с плавающей точкой)

Для записи вещественного числа необходимо хранить знак, цифры до десятичной точки и цифры после десятичной точки.

Любое вещественное число хранится в приближенном представлении, точность которого ограничена разрядностью системы.

Отсюда следует, что **два вещественных числа никогда не могут быть равны друг другу**, поэтому вещественные числа могут сравниваться только с точностью до некоторой заданной малой величины (например, с точностью до **0.0000001**).

Вещественные числа представляются в виде «**мантисса+порядок**» и требуют двух регистров – один для записи мантиссы как целого числа, а другой – для записи порядка как целого числа (форма с плавающей точкой).

Пример: -123.45 может быть представлено как  $-1.2345 \cdot 10^2$ , где -1.2345 – мантисса, 2 – порядок.

Однако то же самое число может быть представлено как  $-12.345 \cdot 10^1$ . Такое представление называют **ненормализованным**.

**Нормализованное** представление означает, что целая часть равна 0 и первый десятичный знак не равен нулю. Таким образом, для рассматриваемого примера нормализованным представлением будет  $-0.12345 \cdot 10^3$ .

В памяти мантисса и порядок хранятся в виде целых чисел, в младшем регистре — мантисса, в старшем — порядок.

Регистр 1	0000000000000010	(Порядок)
Регистр 0	1011000000111001	(Мантисса)

Поскольку на представление вещественных чисел требуется больше памяти, чем для целых, то вычисления с вещественными числами происходят медленнее, чем с целыми и натуральными.

## Представление символов (букв). Кодировочные таблицы (кодировки).

При нажатии на клавишу в компьютер передается код клавиши, который интерпретируется программой, предназначенной для ввода текста, как номер символа.

Возможное количество символов, а также соответствие номера (кода) символа тому или иному символу, появляющемуся на экране, определяется **кодировочной таблицей**, являющейся составной частью программного обеспечения.

Код символа записывается в память как натуральное число.

В настоящее время распространены кодировочные таблицы, в которых на код символа отводится 8 или 16 разрядов (битов).

Если код символа представляется 8 битами (1 байтом), то всего возможно закодировать 256 различных символов (коды от 0 до  $2^8 - 1 = 255$ ).

Фактическим стандартом кодирования символов является 8-битная таблица **ASCII** (American Standard Code for Information Interchange). Эта таблица складывается из двух частей – базовой таблицы и расширений.

Базовая таблица - символы с кодами от 0 до 127 – одинакова для всех программ и включает в себя буквы английского алфавита, знаки препинания и ряд специальных значков.

Расширения таблицы ASCII (символы от 128 до 255) предназначены для кодирования символов национальных алфавитов

Поскольку работа по созданию расширений никак не координировалась, для некоторых алфавитов (в частности, для русского) одновременно существует несколько вариантов расширений.

Для русского алфавита распространены расширения, называемые ГОСТ-альтернативная кодировка (DOS 437), кодировка КОИ-8 и кодировка Windows-1251. Их отличие можно проиллюстрировать простыми примерами.

Символ «А» имеет код 128 в кодировке DOS 437, 225 в КОИ-8 и 192 в Windows-1251.

Передача кода 230 приведет к появлению на экране символа «ц» в кодировке DOS 437, символа «Ф» в КОИ-8 и «ж» в Windows-1251.

Идея расширить количество символов в одной кодировочной таблице для упрощения работы привела к созданию системы кодирования UNICODE.

Первоначально UNICODE была 16-разрядной кодировкой ( $2^{16} - 1 = 65535$  символов), что позволяло в одной таблице хранить самые «популярные» национальные алфавиты. Затем пространство кодов было расширено до  $2^{31}$  возможных символов, хотя на практике используется около 110 000 символов.

## Типы данных и их особенности

### Основные типы данных

К основным типам данных относятся **числа** (натуральные, целые и вещественные) и **символы** (буквы).

С числами можно совершать арифметические действия и операции сравнения, а с символами возможна только операция сравнения, причем символы сравниваются в соответствии с их кодами.

### Производные типы

Производные типы данных – строки и даты.

**Строка** – последовательность символов, включая пробелы.

Для резервирования места в памяти для строк может задаваться максимальная длина (т.е. строка может быть не больше заданного количества символов).

**Календарная дата** – три числа (номер дня месяца, номер месяца и номер года), вычисляемые на основе количества секунд, прошедших с момента начала отсчета. В современных системах началу отсчета, как

правило, соответствует 00 часов 00 минут 00 секунд 1 января 1980 года (эра ПК) или 00 часов 00 минут 00 секунд 1 января 1970 года (эра UNIX).

Для дат определены операции сравнения и вычитания (разница – в днях).

При движении по шкале времени от прошлого к будущему значения дат возрастают.

Для строк определены операции сравнения и слияния (конкатенации).

При сравнении строк применяются следующие соглашения:

- Строки сравниваются в алфавитном порядке (порядок следования в словаре, А < Я)
- Длинная строка больше короткой (при одинаковом начале).

**Примеры сравнения строк:**

- «дом» < «домик»
- «президент» < «президиум».

Слияние (конкатенация) строк заключается в том, что следующая строка присоединяется к концу предыдущей. Эта операция условно обозначается знаком «+».

**Пример слияния строк:** «ветер» + «и» + «нары» = «ветеринары».

## **Индексированные типы данных (массивы)**

Индексированные (перечислимые) типы данных (**массивы**) состоят из конечного числа элементов данных одного и того же типа. Конкретное значение элемента определяется его положением (**индексом**) в массиве. Количество индексов называется **размерностью** массива.

**Одномерный массив (вектор):**

ДНИНЕДЕЛИ = (пн, вт, ср, чт, пт, сб, вс)

Элемент массива: ДНИНЕДЕЛИ[5] = пт (если счёт начинается с 1).

**Двумерный массив (матрица):**

$$A = \begin{bmatrix} 1.23 & 2.34 & 3.45 & 4.56 \\ 6.54 & 5.43 & 4.32 & 3.21 \\ 2.31 & 5.64 & 3.42 & 4.35 \end{bmatrix}$$

Элемент массива: A[3,2] = 4.32

## Структурированные типы – записи

Запись состоит из различных типов данных, причем для каждого типа определяется длина (если это необходимо).

Записи используются для хранения в одной структуре данных характеристик какого-либо объекта, если эти характеристики имеют различный тип (числа, строки и т.д.).

Например, объект типа «город» может характеризоваться численностью населения, географическими координатами и названием.

Составим соответствующую запись:

**ГОРОД = запись из**  
**Название: строка длиной 20 символов;**  
**Население: натуральное число;**  
**Широта: вещественное число;**  
**Долгота: вещественное число;**  
**КОНЕЦ**

Различные типы и структуры данных, образующие запись, называются **полями**.

С записями возможны операции выбора по значению поля и сортировки по какому-то полю.

Можно обращаться также к отдельным полям записей и использовать их в вычислениях в соответствии с типами данных в этих полях.

Выражение «ГОРОД.Название = Санкт-Петербург» присваивает конкретное значение полю «Название» в записи ГОРОД.

Если где-то хранится множество записей типа ГОРОД с информацией о различных городах, то можно получить данные по городу, зная его название.

Если «ГОРОД.Название = Санкт-Петербург», можно определить значение поля «ГОРОД.Население».

Записи как структуры данных (Record) впервые были введены в языке программирования PASCAL.

В современных языках программирования они также используются либо в том же самом виде, либо имитируются другими структурами (например, кортежи в Python).

В явном виде записи используются в хранилищах информации, называемых **базами данных**.

## Задания для самостоятельной работы

1. Определить минимальное и максимальное натуральное и целое числа в 16-разрядной вычислительной системе.
2. Определить минимальное и максимальное натуральное и целое числа в 32-разрядной вычислительной системе.
3. Определить объем памяти при прямой адресации в 16-разрядной вычислительной системе.
4. Определить объем памяти при прямой адресации в 32-разрядной вычислительной системе.
5. Вычислите разницу дат 31 августа 2015 и 12 апреля 2010.
6. Объясните причину, по которой рекомендуется указывать длины строк при их описании при составлении программ.

## Многоуровневая организация ВМ

### Варианты разделения на уровни

#### По аппаратным средствам:

- Логические элементы
- Интегральные схемы (большие ИС – БИС и сверхбольшие ИС – СБИС)
- Функциональные узлы (процессор, модули памяти, контроллеры и адаптеры)
- Блоки вычислительной системы (системный блок, монитор, принтер и т. п.).

#### По программным средствам:

- Базовые программы: обеспечивают работу отдельных узлов и возможность работы системных программ
- Системные программы: обеспечивают человеко-машинный интерфейс и работу прикладных программ
- Прикладные программы: обеспечивают решение пользовательских задач.

### Языки программирования

**Программа** – набор инструкций для процессора (АЛУ), обеспечивающий выполнение заданного **алгоритма** с имеющимися **данными**.

**Язык программирования** – способ представления программы человеком.

Программы могут быть представлены графически (в виде блок-схем), в машинных кодах, в кодах ассемблера или на языках программирования высокого уровня.

**Машинные коды** – содержание **регистров команд** процессора и изменение этого содержания на каждом такте работы процессора. Непосредственно связано с **моделью** процессора.

**Ассемблер** – мнемоническая запись (сочетаниями букв – *push*, *pop*, *jump*) операций, выполняемых процессором, с указанием шестнадцатеричных адресов **регистров данных**, с которыми нужно выполнять эти операции.

Ассемблер может применяться для **ряда процессоров** одной архитектуры.

**Языки программирования высокого уровня** – правила создания набора инструкций из человекочитаемых конструкций (*print(a)* или *list.reverse()*).

Программа на ЯП высокого уровня не зависит от конкретной реализации ВМ (процессора, операционной системы и пр.).

Перевод программы с ЯП в машинные коды обеспечивается специальными программами – трансляторами ЯП, которые могут быть реализованы как компиляторы или интерпретаторы.

**Компилятор** – транслятор, переводящий программу в **исполняемый модуль** (машинные коды). Исполняемый модуль может использоваться сам по себе, но только для такой же (или совместимой) реализации ВМ.

**Интерпретатор** – транслятор, обеспечивающий выполнение инструкций из программы без дополнительного преобразования.

Интерпретируемые программы работают всюду при наличии в ВС соответствующего интерпретатора.

Программы на компилируемых ЯП более эффективно используют возможности программно-аппаратной архитектуры ВС, а программы на интерпретируемых ЯП являются переносимыми.

Существует около 3000 ЯП высокого уровня. Это говорит о том, что идеальных ФП не существует и каждый ЯП позволяет решать какие-то классы задач более эффективно, чем другие ЯП, а остальные классы задач – менее эффективно.

# Вычислительная система как открытая система

## Понятие открытой системы

В широком смысле **открытой системой** может быть названа любая система (компьютер, вычислительная сеть, ОС, программный пакет, другие аппаратные и программные продукты), которая построена в соответствии с **открытыми спецификациями**.

**Открытые спецификации** – опубликованные, общедоступные спецификации, соответствующие стандартам и *принятые в результате достижения согласия* после всестороннего обсуждения всеми заинтересованными сторонами.

Использование при разработке систем **открытых спецификаций** позволяет любым заинтересованным лицам и организациям разрабатывать для этих систем различные аппаратные или программные средства расширения и модификации, а также создавать программно-аппаратные комплексы из продуктов разных производителей.

**Открытые спецификации**, зафиксированные в документах и утвержденные авторитетными организациями, являются **стандартами**.

## Спецификации POSIX

Для программных средств на уровне операционных систем и прикладного программного обеспечения открытые спецификации приводят к мобильности (переносимости) ПО с одного набора аппаратных средств (аппаратной архитектуры) на другие архитектуры. Тем самым обеспечивается существенная экономия ресурсов, используемых при разработке ПО.

Один из общепринятых способов повышения мобильности ПО - стандартизация окружения приложений: предоставляемых программных интерфейсов, утилит и т.п. На уровне системных сервисов подобное окружение описывает стандарт POSIX (Portable Operating System Interface – мобильный интерфейс операционной системы). В настоящее время данный стандарт закреплен в качестве международного стандарта ISO/IEC 9945 (IEEE Std 1003.1).

Стандарт POSIX описывает множество базовых, системных сервисов, необходимых для функционирования прикладных программ. Доступ к ним предоставляется посредством интерфейса, специфицированного для языка C, командного языка и общеупотребительных служебных программ.

У каждого интерфейса есть две стороны: вызывающая и вызываемая. Стандарт POSIX ориентирован в первую очередь на вызывающую. Его цель - сделать приложения мобильными на уровне исходного языка. Это значит, в частности, что при переносе С-программ на другую операционную платформу потребуется перекомпиляция. О мобильности выполнимых программ и/или объектных файлов речь не идет.

Существует большое число операционных систем (ОС), предоставляющих необходимые сервисы и тем самым поддерживающих выполнение POSIX-совместимых приложений. Такие системы называют POSIX-совместимыми. Можно утверждать, что следование стандарту POSIX облегчает перенос приложений практически на любую сколько-нибудь распространённую операционную платформу. Дополнительные усилия по повышению мобильности, прилагаемые на этапе разработки, безусловно, окупятся.

Определяя интерфейс к системным сервисам, POSIX оставляет за рамками рассмотрения их реализацию. В частности, не различаются системные вызовы и библиотечные функции. Не являются объектом стандартизации средства администрирования, аппаратные ограничения и функции, необходимые только суперпользователю, что еще раз подчеркивает направленность стандарта POSIX на приложения, а не на операционные системы.

POSIX нейтрален по отношению к системной архитектуре и разрядности процессора. Это очень важный аспект мобильности приложений.

Ориентация на международный стандарт языка С определила не только стиль описания функций, но и, до некоторой степени, направление развития спецификаций POSIX в плане синхронизации обоих стандартов.

Кроме того, стандарт IEEE Std 1003.2 (Shell and Utilities) определяет систему команд эталонного командного интерпретатора (POSIX-shell) и структуру аргументов команд. Таким образом, работа с POSIX-совместимыми системами требует минимальных затрат на переобучение.

## **Модель OSI/RM**

Тезис о пользе стандартизации, справедливый для всех отраслей, особое значение приобретает в компьютерных сетях. Суть сети – это соединение разного оборудования, а значит, проблема совместимости является одной из наиболее острых. Без соблюдения всеми производителями общепринятых правил разработки оборудования прогресс в деле создания сетей коммуникационных был бы невозможен. Поэтому все развитие компьютерной отрасли, в конечном счете, отражено

в стандартах – любая новая технология только тогда начинает использоваться широко, когда ее содержание закрепляется в соответствующем стандарте.

При описании взаимодействия двух вычислительных систем (узлов сети) с использованием вычислительной (компьютерной) сети с целью стандартизации такого описания принято разделять все процессы при взаимодействии на **уровни описания** – от электрических сигналов и контроля состояния линий связи до **сетевых сервисов (служб)**.

Процедура взаимодействия двух узлов может быть описана в виде набора правил взаимодействия каждой пары соответствующих уровней обеих участвующих сторон.

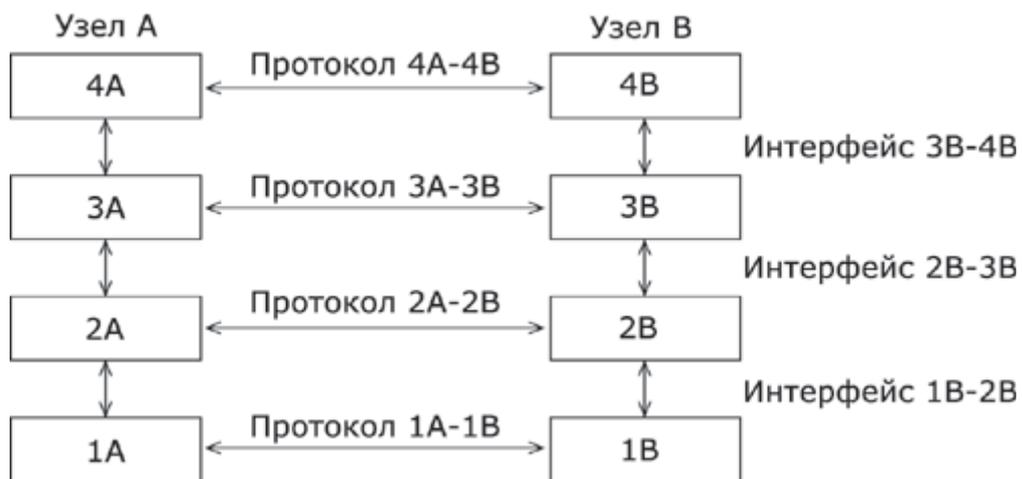


Рис. 12. Уровни вычислительной системы, интерфейсы и протоколы

## Протоколы и интерфейсы

Формализованные правила, определяющие последовательность и структуру (формат) сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах, называются **протоколом**.

Формализованные правила, определяющие взаимодействие компонентов, лежащие на разных (соседних) уровнях для одного узла, называются **интерфейсом**.

Иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети, называется **стеком коммуникационных протоколов**.

**Модель взаимодействия открытых систем (Open System Interconnection, OSI)** определяет различные уровни взаимодействия систем в вычислительных сетях, дает им стандартные имена и указывает, какие функции должен выполнять каждый уровень.

Модель OSI была разработана на основании большого опыта, полученного при создании компьютерных сетей, в основном глобальных, в 70-е годы XX века. Полное описание этой модели занимает более 1000 страниц текста.

Полностью данная модель никогда не реализуется, но она является основной для реальных программных и аппаратных средств, обеспечивающих взаимодействие, поэтому её называют **базовой (reference)** моделью и часто обозначают как **OSI/RM**.

## **Разделение ресурсов вычислительных систем**

Основной целью разделения ресурсов является повышение эффективности их использования. При совместной работе компьютеров в сети возможно перераспределение ресурсов от тех узлов сети, которые мало используют свои ресурсы, «в пользу» тех, кому своих ресурсов недостаточно.

К ресурсам относятся:

1. Оперативная память
2. Возможности процессора (CPU – центральный процессорный модуль)
3. Внешняя память (объём и файлы)
4. Программное обеспечение
5. Устройства вывода.

Разделение ресурсов реализуется с помощью вычислительных сетей (сетей ЭВМ). Существует две модели разделения ресурсов – централизованная и распределенная.

## **Аппаратные средства ЭВМ: основные устройства**

### **Внутренняя память**

- ПЗУ компонентов
- Оперативная память (энергозависимая)
- Память внутри процессора (кэш-память, тоже энергозависимая).

### **Внешняя память**

- Жёсткие диски
- Оптические приводы (DVD/CD)

- Сменные твердотельные носители (флэш-накопители).

### Устройства ввода

- Клавиатура и мышь
- Сенсорные экраны и тачпады
- Сканеры, виде- фото-камеры
- Дигитайзеры и т.п.

### Устройства вывода

- Мониторы
- Принтеры и подобные устройства.

## Централизованная модель разделения ресурсы («хост-терминал»)

Главный компьютер («хост») предоставляет все 5 видов ресурсов

Терминал не обладает никакими ресурсами.

Терминал – это **монитор + клавиатура + средства подключения к компьютерной сети** (дешевое и надежное устройство).

В идеальном случае у терминала отсутствуют программное обеспечение (ПО) и устройства внешней памяти.

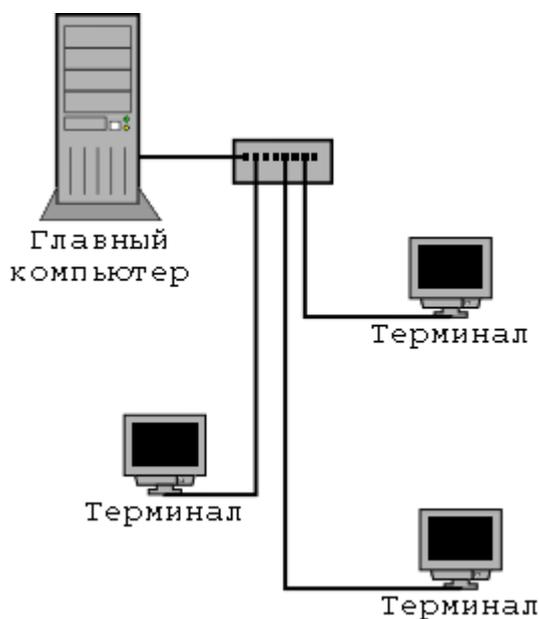


Рис. 13. Модель централизованных ресурсов

Централизованная модель обладает важными достоинствами:

- предельно упрощается и удешевляется сопровождение и администрирование (все ресурсы – в одном месте)
- полный контроль за действиями пользователей

- пользователи не могут делать того, что им не разрешено
- через терминал не занести вирус
- терминалы слабо загружают линии связи, поэтому их может быть много (истории известны конфигурации с несколькими тысячами терминалов).

Однако при таком подходе могут быть выделены и недостатки:

- при выходе из строя главного компьютера вся система перестает работать
  - неудобства для пользователей (графического интерфейса либо вообще нет, либо он ограничен по возможностям, осуществляется контроль за пользователями, отсутствуют ресурсы для личного пользования)
  - требуется квалифицированное администрирование (настройка программ, установка прав доступа и т. д.).

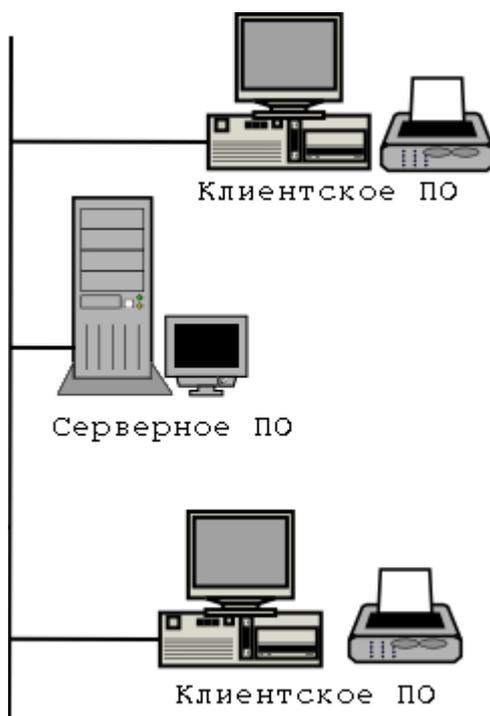
### Распределенная модель разделения ресурсов («клиент-сервер»)

**Сервер** – обычный или специализированный компьютер, снабжённый специальным (серверным) программным обеспечением.

На компьютерах пользователей устанавливаются программы-клиенты.

**Разделяемыми ресурсами** являются устройства ввода/вывода, внешняя память и программы (3 вида ресурсов)

И сервер, и клиенты – полноценные компьютеры, которые могут работать независимо от других. Список разделяемых ресурсов и порядок их использования определяется администратором вычислительной системы.



### Рис. 13. Модель распределенных ресурсов

Распределенная модель обладает важными достоинствами:

- выход из строя одного из узлов сети не приводит к катастрофе
- пользователи на своих компьютерах могут делать что угодно
- гибкая конфигурация – каждый узел может быть и клиентом, и сервером

Однако при таком подходе могут быть выделены и недостатки:

- затруднен контроль за действиями пользователей
- администрирование является достаточно трудоемким (требуется следить за большим количеством узлов)
- если разделяемыми ресурсами являются программы, то сильно загружаются линии связи и требуется быстрая сеть
- во многих случаях требуется согласование конфигураций сервера и клиентов, т.е. замена серверных программ приводит к модернизации компьютеров пользователей или установке дополнительных программ.

### Эволюция разделения ресурсов

Развитие подходов к разделению ресурсов в вычислительных системах прошло через несколько этапов.

**1 этап:** чисто терминальные системы (до появления персональных компьютеров). Очень дорогие «центральные» компьютеры с большим количеством терминалов, подключаемых через последовательные интерфейсы и телефонные линии (современная реализация – **мэйнфрейм**).

**2 этап:** «клиент-серверные» системы (с персональными компьютерами). Сети из сравнительно дешевых компьютеров с распределенными разделяемыми ресурсами и бесконтрольными пользователями.

**3 этап:** терминальные системы возрождаются в виде «тонких» клиентов. Используется либо web-интерфейс, либо специальные терминальные программы (терминальные серверы и клиенты Remote Desktop).

### Варианты клиентов в модели «клиент-сервер»

С технологической точки зрения введены понятия «толстый» и «тонкий» клиент. «Толстый» клиент – компьютер, который запрашивает у сервера **программы**, в дальнейшем работающие на этом компьютере (по сети передаются программы). «Тонкий» клиент – компьютер, посылающий

серверу **данные** и получающий результаты их обработки (по сети передаются данные).

Однако на рынке вычислительных систем «тонкий» клиент трактуется как компьютер с облегчённой версией ОС, не имеющий прикладных программ. Дополнительно существует понятие «сверхтонкий» клиент – вычислительная система, посылающая серверу данные и получающая результаты их обработки. Чаще всего «сверхтонкие» клиенты реализуются при использовании браузера в качестве единственной программы, с которой требуется работать пользователю на своей ВМ.

## **Облачные вычисления**

«**Облако**» – сеть из серверов, на которых размещаются программы обработки данных и персонализированные хранилища данных.

Компьютеры пользователей («сверхтонкие» клиенты) оснащаются заранее настроенными программами для связи с «облаками».

Можно сказать, что подход «облачных» вычислений реализует модифицированную модель «хост-терминал».

В отличие от моделей «хост-терминал» и «клиент-сервер» (в их исходном понимании) для «облачных» сервисов невозможно сказать, где именно хранятся и обрабатываются данные (к анонимности пользователей добавляется некоторая анонимность сервисов).

С позиции пользователя применение «облачных» технологий может быть охарактеризовано как возможность использования в своей деятельности чужих (внешних) ресурсов и информационной инфраструктуры.

С позиции ИТ-консалтинга применение «облачных» технологий может быть охарактеризовано как стиль, при котором масштабируемые ИТ-ресурсы предоставляются внешним пользователям в качестве сервиса с помощью интернет-технологий.

С позиции ИТ-специалиста использование «облачных» технологий может быть охарактеризовано как подход, при котором огромное количество серверов объединяются в один мощный вычислительный механизм с использованием технологий виртуализации, а ресурсы одного сервера подразделяются на виртуальные машины и применяются многими пользователями одновременно.

## **Преимущества и риски использования «облачных» технологий**

Использование «облачных» решений означает аутсорсинг всей ИТ-инфраструктуры. Использование технологий мобильной связи позволяет

вообще не иметь на балансе средств вычислительной техники (сотрудники могут использовать личные ноутбуки/планшеты с GSM-модемами или мобильным WiMAX).

При этом обеспечиваются следующие преимущества:

- Снижение (в пределе – до нуля) усилий на поддержку и развитие корпоративной информационной системы (КИС)
- Мобильность сотрудников (не нужен большой офис и обеспечивается работа в режиме 24 x 7)
- Не нужны системные администраторы.

Использование «облачных» технологий также влечет за собой риски, а именно:

- Отсутствие контроля над данными и сетевым обменом
- Обезличенная ответственность
- Затраты на переобучение и ликвидацию навыков «пользователя ПК».

### **«Облачные» сервисы**

Со стороны пользователя, работа с «облаком» ничем не отличается от работы с сайтом, т.е. требуется только браузер. В некоторых случаях используются специализированные программы-агенты, но это не обязательно.

В «облаках» могут предоставлять следующие сервисы:

- Хранение документов и прочих файлов
- Электронная почта
- Календарь-планировщик
- Он-лайн создание документов и совместная работа с ними
- Система обмена мгновенными сообщениями
- «Личный кабинет» пользователя

### **Внешние и внутренние «облака»**

**Внешнее «облако»** – это сервис или набор сервисов, доступный всем без исключения. При этом вычислительные услуги должны быть оплачены, и для получения доступа к сервисам необходимо иметь выход в Интернет

**Внутреннее (частное) «облако»** – корпоративные пользователи имеют доступ к сервисам только в стенах предприятия, защищенного сетевым экраном. «Точкой входа» может быть корпоративный портал – сервисы становятся доступными после авторизации по защищённому протоколу.

Для создания частного «облака» для корпоративных целей можно использовать следующие возможности:

- Использовать Google Web Toolkit и получить «личный» вариант Google Docs
- Использовать систему электронного документооборота (СЭД) любого класса с web-интерфейсом.

Для использования в управленческой деятельности принципиально различаются два варианта «облачных» решений:

- Нет он-лайн редактирования офисных документов (используются программы на компьютерах пользователей), доступ к файлам через WebDAV (пример – Google Drive).
- Есть он-лайн редактирование офисных документов, на компьютерах пользователей – только браузеры.

### **Три основных варианта «облачных» ресурсов**

- **Приложение как сервис (SaaS, Software as a Service):** провайдер предоставляет интерфейс к прикладному ПО (самый распространённый вариант). Самый известный пример – Google Docs. Есть специализированные сервисы для математиков, медиков и др.
- **Платформа как сервис (PaaS, Platform as a Service):** провайдер предоставляет «конструктор» (framework) для разработки собственных приложений. Пример – Google Apps.
- **Инфраструктура как сервис (IaaS, Infrastructure as a Service):** использование сервера и дискового пространства, удаленных от пользователя. На этой основе можно использовать любое пользовательское ПО (с учетом лицензионных ограничений).

### **Задания для самостоятельной работы**

Подготовить письменное сообщение (эссе) по указанному преподавателем варианту заданий из приведенного ниже списка. При подготовке сообщения выписать термины и сокращения, дать их объяснение и расшифровку.

1. Микропроцессоры: понятие и назначение. Технологии производства и области применения
2. Характеристики современных (микро)процессоров. Закон Мура. Применимость закона Мура.
3. Процессоры Intel: от 8088 до 486.
4. Процессоры Intel: от Pentium до Core i7.

5. Процессоры AMD.
6. RISC-процессоры. Особенности реализаций.
7. Технологические решения и параметры процессоров Alpha (DEC/HP).
8. Конвейерная обработка в процессорах.
9. Процессоры VLIW. Принципы построения и реализации.
10. Эволюция процессоров «Эльбрус».
11. Проектная норма как универсальная характеристика уровня технологии. Влияние проектной нормы на уровень интеграции.
12. Многооперационная обработка (суперскалярность). Способы суперскалярной обработки данных в современных процессорах.
13. Понятие интерфейса в вычислительной системе. Контроллер (адаптер) аппаратного интерфейса. Параллельные и последовательные интерфейсы. Характеристики аппаратных интерфейсов.
14. Последовательные интерфейсы: COM, USB, E-SATA. Параметры контроллеров, скорости обмена данными, применение.
15. Параллельные интерфейсы: LPT, IDE, SCSI. Применение, разрядность, скорость обмена данными.
16. Интерфейсы устройств внешней памяти: SCSI, iSCSI, SAS, FiberChannel. Применение, скорости обмена данными.
17. Шина PCI. Варианты, характеристики, возможности подключения адаптеров.
18. Интерфейсы видеоустройств: аналоговое видео, HDMI, DVI. Особенности, области применения.
19. Универсальный интерфейс USB. Первоначальная идея и развитие. Особенности USB 3.0. Конструкционные решения.
20. Принципы организации синхронной и асинхронной передачи данных. Синхронные и асинхронные интерфейсы в вычислительной системе.

## Операционные системы

**Операционная система (ОС):** набор программ (в том числе и *микропрограмм*), выполняющий три основные функции

1. Управление аппаратными средствами вычислительной системы
2. Управление программными средствами

### 3. Обеспечение взаимодействия пользователя с вычислительной системой (*пользовательского интерфейса*)

#### **Варианты классификации ОС**

- По количеству одновременно решаемых задач (однозадачные или многозадачные);
- По количеству одновременно работающих пользователей (однопользовательские или многопользовательские);
- По возможности работы в компьютерных сетях (несетевые, сетевые или распределенные);
- По разрядности (16-, 32- или 64-разрядные);
- По возможности защиты (незащищенные или защищенные);
- По порядку лицензирования (коммерческие или свободные).

**Разрядность** определяет диапазон представления чисел и влияет на количество адресов (объем) оперативной памяти (**почему?**), возможности использования больших объемов внешней памяти и диапазоны представления календарных дат.

В **однопользовательской** ОС пользователь имеет полный доступ ко всем ресурсам системы (настройкам, файлам, папкам) и может как угодно переделывать систему. Он является «хозяином» и несет полную ответственность за свои действия.

В **многопользовательской** ОС всегда существует «главный» пользователь (администратор, суперюзер), который назначает права доступа к ресурсам для остальных пользователей. Наличие различных прав доступа означает возможность защиты программ и данных.

**Несетевые** ОС не имеют средств работы в компьютерных сетях, и для подключения к сети компьютеров с такими ОС требуется установка дополнительных программ.

**Сетевые** ОС содержат средства для работы в сети в своем составе, и компьютеры с такими ОС могут работать как отдельно, так и в составе сети.

**Распределенные** ОС не могут работать без компьютерной сети. Для работы им требуется не менее 2-х компьютеров.

16-разрядные, несетевые, однопользовательские ОС в настоящее время используются только на сильно устаревших или узкоспециализированных аппаратных платформах.

## Многозадачность в операционных системах

*В однозадачной ОС каждая программа полностью захватывает всю систему, и пока она не прекратит работу, нельзя запустить другую программу.*

В многозадачной ОС запускается одновременно много программ, и можно переключаться между ними (например, с помощью Alt+Tab).

Многозадачность может быть принудительная (**режим разделения времени**) и вытесняющая (выбранная программа - –активное окно– - использует большую часть ресурсов системы, а все остальные программы работают в –фоновом режиме–).

**Системные программы (процессы)** имеют для систем с вытесняющей многозадачностью более высокий приоритет, чем **пользовательские программы (процессы)**, поэтому работа пользователя может существенно замедляться при выполнении системных задач, активно использующих ресурсы (индексация файлов, проверка на вирусы и т.п.).

### Режим разделения времени

Основной режим в высокопроизводительных ЭВМ. Каждому пользователю выделяется некоторый интервал времени («квант»), в течение которого система обслуживает этого пользователя. Затем система переключается на обслуживание следующего пользователя и т.д. Чаще всего переключение происходит циклически (по кругу). Промежуток времени между квантами обслуживания для конкретного пользователя – время ожидания обслуживания. Системы, в которых время ожидания никогда не превышает заданной величины – **системы реального времени**.

## Задания для самостоятельной работы

Подготовить ответы на следующие вопросы для самопроверки:

1. Сколько пользователей может одновременно обслужить мэйнфрейм с производительностью 10 GFLOP/S, если время ожидания не превышает времени реакции пользователя (0,1 сек), а квант составляет 100 000 операций?
2. Что такое «интерфейс»?
3. Что такое «протокол»?
4. Какие виды ресурсов предоставляются для совместного использования в модели «хост-терминал»?

5. Какие виды ресурсов предоставляются для совместного использования в модели «клиент-сервер»?
6. В чем отличие «толстого», «тонкого» и «сверхтонкого» клиентов?
7. Как может быть охарактеризован в понятиях модели «клиент-сервер» смартфон, использующий GoogleDrive для хранения файлов?
8. Как может быть охарактеризован в понятиях модели «клиент-сервер» смартфон, использующий GoogleDocs для работы с офисными документами?

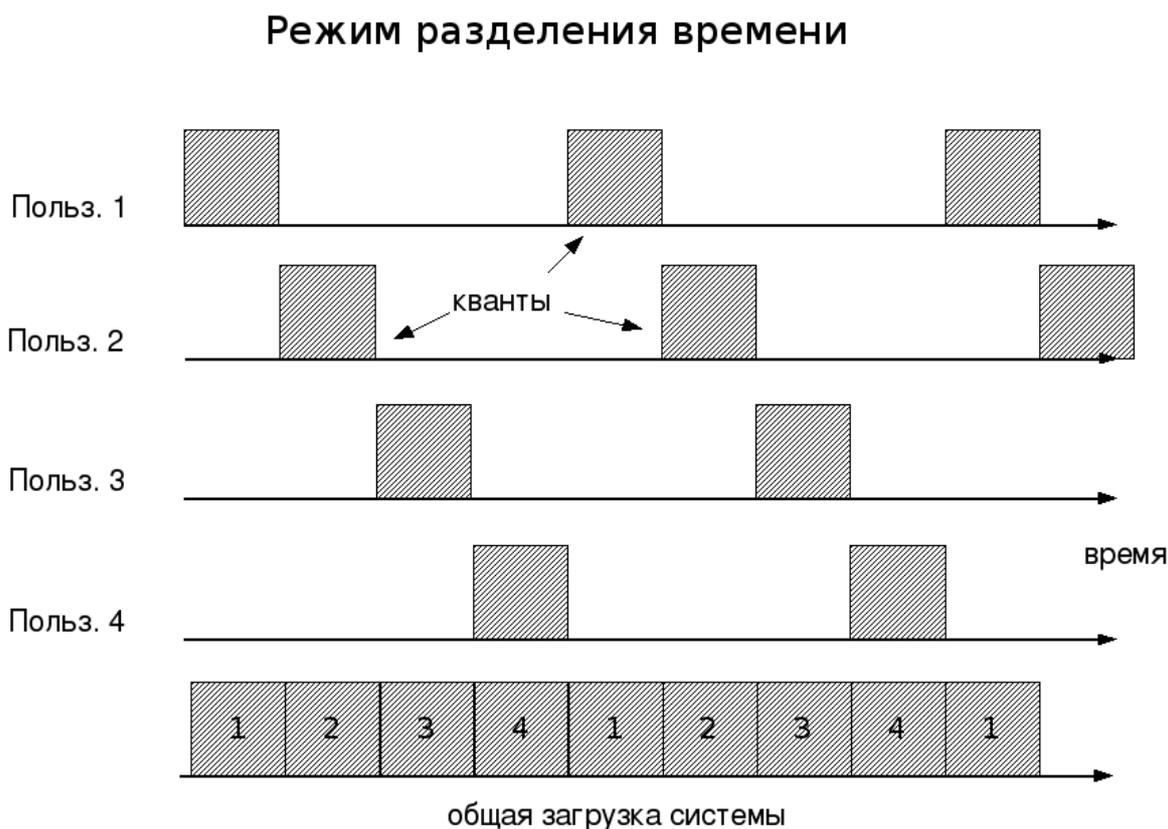


Рис. 14. График загрузки системы в режиме разделения времени

## Внешняя память и файловые системы

### Файлы и дескрипторы

**Файл** – поименованная совокупность данных (байтов), размещенная на устройстве внешней памяти. Файлы можно создавать, открывать, закрывать, копировать, переименовывать/перемещать, выводить на устройства вывода и удалять.

С каждым файлом связывается **дескриптор** – служебная информация о файле. Дескриптор обычно содержит *тип файла* (данные, программа или принадлежность к какой-то прикладной программе), *дату и время создания*, *права доступа* (владелец/группа) и *атрибуты* (скрытый, ссылка и пр.).

Файлы объединяются в *каталоги (папки)*.

## **Файловая система**

**Файловая система (ФС)** – правила и методы именования файлов и папок (каталогов), доступа к файлам и папкам, составления дескрипторов файлов, обеспечения защиты и целостности файлов.

Целостность файла означает, что в нем хранится только та информация, которая в нем должна быть, и не хранится того, чего не должно быть.

Каждая операционная система (ОС) работает с одной или несколькими ФС.

## **Полное имя файла**

*Полное имя файла* включает в себя последовательный список всех папок, начиная от основной папки в разделе, в порядке их вложенности, которые нужно «пройти», чтобы добраться до конкретного файла. Каждый файл имеет собственное имя, но файлы с одинаковыми именами в разных папках для ФС являются различными, т.к. у них разные полные имена.

Конкретный вид полного имени файла зависит от ФС.

В некоторых ФС для файлов нужно в явном виде указывать расширение (тип), т.е. принадлежность файла к какой-либо программе.

## **Внешняя память: разделы MBR и GPT**

В соответствии со спецификацией MBR на каждом физическом устройстве внешней памяти (диске) может находиться не более 4-х основных (первичных, физических) разделов.

Если необходимо разместить более 4-х разделов, то последний основной объявляется расширенным и затем в нем создаются логические разделы. Количество логических разделов (томов, дисков) в расширенном разделе теоретически не ограничено, хотя существуют ограничения самих операционных систем на максимальное количество логических дисков. Например, в Windows число доступных дисков ограничено количеством букв.

Один из основных разделов обязательно должен быть объявлен активным (А). Это необходимо для загрузки операционной системы типа Windows (в этом случае активным должен быть первый основной раздел).

В каждом разделе выделяется две области – **область файловой системы** (в начале раздела) и **область данных**. В области ФС размещается информация о файлах и папках, дескрипторы файлов и другая служебная информация.

Для работы с областью ФС используются служебные программы создания файловых систем (format, mkfs) и прикладные программы – менеджеры файлов.

Помимо ограничения на 4 основных раздела спецификация MBR имеет еще одно ограничение – нельзя адресовать дисковое пространство более 2 Тбайт.

Эти ограничения снимаются при переходе к другому варианту формирования таблиц разделов – GPT (GUID Partition Table), где GUID (Globally Unique Identifier) – статистически уникальный 128-битный идентификатор.

Для дисков GPT нет ограничений на количество основных разделов и нет практических ограничений на адресуемый объем дисковой памяти (теоретически — до  $9.4 \cdot 10^{21}$  байт).

Первые 512 байт по-прежнему содержат таблицу MBR (т.н. «защищенный MBR») для совместимости со старыми аппаратными платформами (BIOS).

На дисках GPT загрузчик ОС находится не в первом разделе.

## **Блоки и фрагментация**

Вне зависимости от размера файла информация на диски записывается блоками. Размеры блоков фиксированы и бывают 512, 1024, 2048, 4096 или 8192 байта. В различных файловых системах размеры блоков различаются. Размеры файлов не равны размерам блоков и изменяются в течение жизни файлов. Возникает эффект фрагментации.

## **Особенности некоторых файловых систем**

- **FAT32**: основная файловая система для современных USB-носителей и карт памяти. Размер блока (кластера) зависит от размера логического раздела. Отличается высокой скоростью чтения-записи. Не позволяет ограничивать права пользователей. Предельный размер файла –  $2^{32}$  байт. Не предусмотрено восстановление файлов после сбоя ОС (штатными средствами). Размер раздела, создаваемого

средствами ОС Microsoft – не более 32 Гбайт. Поддерживается всеми современными ОС.

- **exFAT**: разработана Microsoft для устранения ограничений FAT32. Предельный размер файла –  $2^{64}$  байт. Для полной поддержки в устройствах (планшетах, фото-видеокамерах) требуется лицензирование у Microsoft.

- **NTFS**: основная ФС в современных ОС Microsoft. В первоначальном варианте ребрендинг HPFS фирмы IBM. Размер блока (кластера) 512 байт. Предельный размер файла –  $2^{64}$  байт. Позволяет разделять права пользователей (ACL – Access Control List). Позволяет восстанавливать данные после сбоев ОС за счет ведения *журнала* ФС. В ОС Microsoft не предусмотрено создание NTFS на сменных носителях. Поддерживается всеми современными ОС (кроме ОС для мобильных устройств).

- **ext2/3/4**: основные ФС в ОС семейства GNU/Linux. Позволяют разделять права пользователей. ext3/4 – журналируемые.

- **CDFS (ISO 9660)**: Это ФС для цифровых компакт-дисков (дисков с данными). Поддерживается любыми ОС. Чтобы CD читался где угодно, нужно при записи создавать на нем именно эту ФС (сделать соответствующие настройки в программе записи CD). Имеет ограничения на длину имени файла и на использование национальных кодировок в именах файлов.

- **UDF**: (Universal Disk Format, универсальный дисковый формат) – спецификация ФС, независимой от операционной системы для хранения файлов на оптических носителях. UDF является реализацией стандарта ISO/IEC 13346. Формат UDF снимает ограничения ISO 9660. UDF также применяется для DVD, так как нет ограничений на размер файла. Поддерживается во всех современных ОС.

## Лицензирование операционных систем

### Виды лицензионных соглашений

- **Несвободные лицензии** (название «коммерческие» не совсем корректно): авторский договор обеспечивает **право использования программных средств при определенных условиях** (на количество компьютеров, на срок и т.п.). Как правило, такое право передается за деньги. Техническая поддержка оплачивается отдельно.

- **Свободные лицензии**: авторский договор обеспечивает **право распоряжаться программными средствами по усмотрению пользователя** (кроме нарушения законодательства). Такие права

могут передаваться как за деньги, так и бесплатно. Техническая поддержка оплачивается отдельно.

### **Политика лицензирования Microsoft Windows**

- На предприятиях для настольных компьютеров можно использовать версии Windows Basic, Home, Professional или Business (Windows Basic и Home не позволяют в полной мере использовать централизованную авторизацию)
  - Количество одновременных подключений к ресурсам настольного компьютера – не более 20 (для Windows Prof.)
  - Для обеспечения количества одновременных подключений к ресурсам компьютера (в модели «клиент-сервер») нужна лицензия Windows Server
  - Для обеспечения подключений к ресурсам компьютера (в модели «хост-терминал») нужны отдельные лицензии для Windows Server

В таблице ниже приведены ориентировочные цены на различные варианты лицензий на право использования Microsoft Windows.

<b>Продукт (цены на октябрь 2014 г., руб.)</b>	<b>Примерная стоимость</b>
Microsoft Windows 8.1 (электронная версия, без носителя)	4283
Windows Server 2012 Standard (на сервер, минимальное количество пользовательских лицензий — 5)	30220
Microsoft Windows Server Standard 2012 User CAL (дополнительная пользовательская лицензия)	1151
Microsoft Windows Server Standard 2012 Remote Desktop Server Device CAL (лицензия на подключение в режиме терминала)	3472

### **Свободные лицензии**

Понятие «свободного произведения», первоначально сформулированное для программного обеспечения, а потом распространившееся на программную документацию, литературные произведения, графические материалы и фотографии, видео- и аудиопродукцию, подразумевает четыре основных права («свободы») пользователей таких произведений (точные формулировки приведены в ГОСТ Р 54593-2011 «Информационные технологии. Свободное программное обеспечение. Общие положения»):

1. право использования произведения в любых не противоречащих законам целях

2. право распространения произведения
3. право доработки или иной модификации произведения
4. право распространения модифицированных (производных) произведений

Свободное программное обеспечение в подавляющем большинстве случаев является результатом коллективного труда международных групп разработчиков (сообществ), поэтому авторские договоры (лицензии) также имеют международный статус.

Существует Приказ Минкомсвязи от 13.01.2012 N10 «О создании рабочей группы по вопросу адаптации международных свободных лицензий к национальному законодательству и их применения на территории Российской Федерации».

Основные варианты лицензий на свободное программное обеспечение (СПО) приведены в таблице ниже.

<b>Сравниваемые условия</b>	<b>GNU GPL</b>	<b>BSD</b>	<b>MIT/X11</b>	<b>Mozilla public license (MPL)</b>	<b>Apache software license (ASL)</b>
Требуется указывать имя автора	Да	Да	Да	Да	Да
Измененные файлы должны быть помечены	Да	Нет	Нет	Да	Нет
Наименование производного ПО должно отличаться от наименования продукта создателей лицензии	Нет	Нет	Нет	Нет	Да (если нет письменного разрешения лицензиара)
Производные произведения должны распространяться на условиях первоначальной лицензии	Да	Нет	Нет	Да (только для исходного текста)	Нет
Указана территория, на которую предоставляется лицензия	Нет	Нет	Нет	Да	Нет
Отсутствие гарантий на ПО	Да	Да	Да	Да	Да

<b>Сравниваемые условия</b>	<b>GNU GPL</b>	<b>BSD</b>	<b>MIT/X11</b>	<b>Mozilla public license (MPL)</b>	<b>Apache software license (ASL)</b>
Предоставляется право применить другую лицензию	Нет	Не указано	Не указано	Да	Не указано

Противоречие для MPL в условиях распространения производных произведений и права применять другую лицензию связано с тем, что требование сохранения исходной лицензии относится к производным произведениям, распространяемым в виде исходных текстов. Однако если производное произведение было скомпилировано как составная часть (модуль, библиотека) другого программного продукта, то результат может распространяться как целостный программный продукт под любой другой лицензией (в том числе и несвободной).

Если какие-то из этих лицензий будут иметь юридическую силу в Российской Федерации, то к ним будет относиться положение части 1 статьи 1235 ГК РФ [6], говорящее о том, что «Право использования результата интеллектуальной деятельности или средства индивидуализации, прямо не указанное в лицензионном договоре, не считается предоставленным лицензиату». Таким образом, во всех случаях, в которых явно не указана возможность смены лицензии (а это все основные свободные лицензии на ПО, за исключением MPL), никто не сможет получить исключительные права на программный продукт, производный от распространяемого по одной из свободных лицензий.

## **Отечественное программное обеспечение**

С середины 2014 года идет работа по переводу государственных корпораций и федеральных органов исполнительной власти на программное обеспечение российского производства с целью обеспечения независимости в области информационных технологий. На начало 2015 года сформировалось определение отечественного (российского) программного обеспечения (ПО), как ПО, разрабатываемого российскими производителями..

Отечественными (российскими) производителями программного обеспечения (ПО), занимающимися производством собственного (проприетарного) ПО или «свободного программного обеспечения» (СПО), либо предоставляющих услуги по разработке, тестированию и поддержке ПО по заказам сторонних организаций, могут быть признаны российские юридические лица, в которых не менее чем 51% долей в уставном капитале

или акций, производных инструментов и других инструментов корпоративного контроля принадлежат прямо или косвенно российским гражданам или государственным образованиям, а также физическим лицам, являющимся гражданами и налоговыми резидентами РФ.

## **Обеспечение производительности и надежности вычислительных систем**

### **Понятие производительности**

**Производительность** – один из главных показателей эффективности вычислительной системы (ВС), определяет ее вычислительную мощность через количество той или иной вычислительной «работы», выполняемой системой в единицу времени, или в течение некоторого временного интервала.

Единицей измерения производительности компьютера является **время**: компьютер, выполняющий тот же объем работы за меньшее время, является более быстрым.

### **Оценки производительности**

Производительность ВС может оцениваться субъективно и объективно.

Субъективной оценкой является реакция ВС на действия пользователя (запуск приложений, открытие/заккрытие окон и т.п.). При этом производительность воспринимается относительно других ВС, с которыми сталкивается пользователь, и зависит от программного окружения, работы системного и служебного ПО.

Объективная оценка производительности получается как количественная оценка, полученная после прохождения **тестов производительности**. Для тестирования производительности возможно использование нескольких методик, в частности:

- Использование средств оценки производительности в составе утилит ОС
- Для серийных моделей, выпускаемых крупными компаниями – использование стандартных наборов тестов SPEC.
- Для суперкомпьютеров применяется оценка вычислительной производительности по количеству выполняемых операций с плавающей точкой (т.е. с вещественными числами) в секунду – T(FLOP/S)

## Факторы, влияющие на производительность

- Тактовая частота процессора
- Количество процессоров/ядер
- Объем оперативной памяти
- Скорость работы интерфейсов устройств внешней памяти (скорость обмена между оперативной и внешней памятью)
- Объем видеопамяти
- Операционная система
- Вариант графического пользовательского интерфейса (если он есть)

## Тесты SPEC

SPEC (Standard Performance Evaluation Corporation) выпускает наборы методик и данных для проведения тестирования различных аспектов работы вычислительной системы. Наиболее часто используются следующие оценки.

- CPU2006: «чистая» производительность процессора на вычислениях, различаются тесты для целых чисел (CINT2006) и для чисел с плавающей точкой (CFP2006)
- SPECviewperf: скорость обработки (рендеринга) 3D-моделей с использованием библиотеки OpenGL
- SPECparc: производительность трехмерной графики, для каждого 3D-редактора измеряется отдельно.
- SPECjvm и SPECjEnterprise: позволяют определять производительность Java-машин и узлов Internet в режиме серверов приложений Java
- SPECvirt\_cs: определение производительности средств виртуализации в центрах обработки данных (кластерах).

Производители вычислительных систем покупают наборы тестов, проводят тестирование по методике SPEC, а затем предоставляют SPEC результаты тестирования, которые публикуются в открытом доступе. Эти результаты можно найти на сайте SPEC в соответствующем разделе.

## Кластерные технологии

Кластерные вычисления – технология повышения производительности и надежности ВС, при которой компьютеры соединяются через скоростные каналы связи для совместного решения задач (*концепция виртуального суперкомпьютера*)

Вычислительные узлы этой сети ведут скоординированную работу, используют ресурсы друг друга и потенциально доступны из любой точки системы.

Компьютеры могут быть удалены друг от друга и использовать разные типы коммуникаций, однако для конечного программного продукта и пользователя они играют роль единой вычислительной машины.

### **Варианты кластеров**

- кластеры высокой доступности
- MPP-кластеры (с распределённой памятью)
- высокопроизводительные кластеры (объединение процессоров)
- кластеры распределённой нагрузки (балансировка задач)
- Grid-системы.

Grid-системы характеризуются удалённым расположением вычислительных узлов друг от друга, слабой связью узлов между собой через интернет-каналы, причём доступность того или иного узла в произвольный момент времени не гарантирована.

### **Состав кластера**

В состав кластера входят (см. рис. 15):

1. Управляющий сервер (обеспечивает управление восстановлением системы, мониторинг, резервное копирование)
2. Серверы хранения данных и дисковые накопители (один или несколько, предоставляют совместный доступ к системе хранения данных другим серверам кластера)
3. Система хранения данных (одна или несколько)
4. Вычислительные узлы (выполняют рабочую нагрузку кластера)
5. Узлы планирования (обеспечивают запуск нужных задач в нужное время на нужных вычислительных узлах)
6. Пользовательские узлы (виртуальные рабочие места пользователей)
7. Коммутатор, обеспечивающий внутренние и внешние соединения.

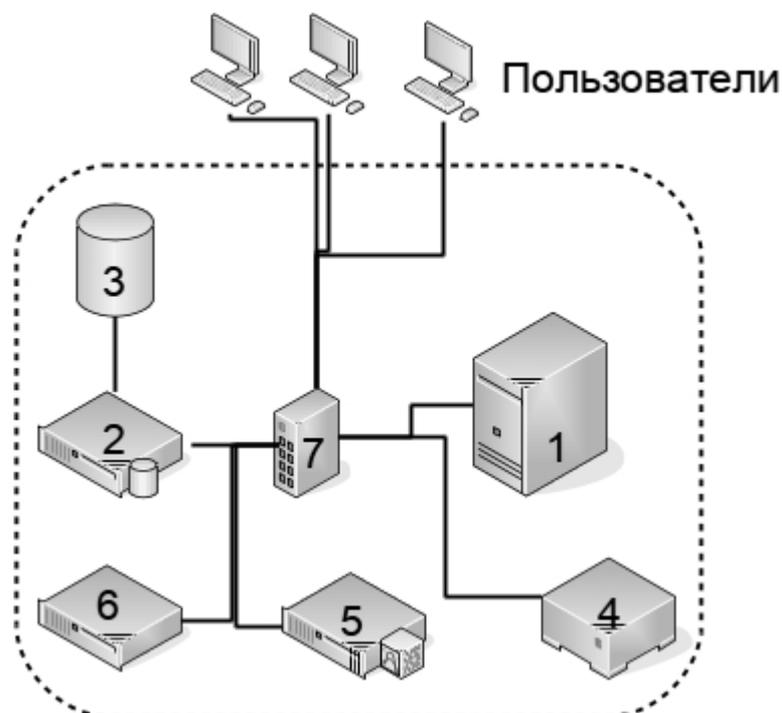


Рис. 15. Типовой состав вычислительного кластера

Функции узлов кластера могут совмещаться.

При создании вычислительных кластеров реализуются три основных принципа

- Масштабирование
- Резервирование
- Виртуализация

### Масштабирование

**Масштабирование** в ИТ – возможность увеличения количественных характеристик системы без изменения архитектуры (конструкции).

Масштабируемыми параметрами в вычислительных системах являются

- Количество пользователей
- Количество процессорных модулей (или компьютеров-узлов)
- Объём оперативной памяти
- Объём дисковой памяти
- Производительность (количество заданных операций в единицу времени)

Условиями реализации масштабирования являются

- Магистрально-модульная архитектура, стандартизация
- Предусмотренный рост скорости компьютерной сети
- Предусмотренный рост энергопотребления

- Программные средства (ОС, web-серверы, серверы баз данных), имеющие возможности масштабирования
- Решение проблемы лицензионных ограничений

### Резервирование в кластерных системах

В общем случае в кластерных системах реализуется три вида резервирования.

- Резервирование вычислительных узлов (заложено в идее кластера);
- Резервирование подсистемы внешней памяти;
- Резервирование данных (резервное копирование – backup).

### Резервирование подсистемы внешней памяти. Дисковые массивы RAID.

**RAID (Redundant Array of Independent disks – избыточный массив независимых дисков):** массив из нескольких дисков (запоминающих устройств), управляемых контроллером, связанных между собой скоростными каналами передачи данных и воспринимаемых внешней системой как единое целое. В зависимости от типа используемого массива может обеспечивать различные степени отказоустойчивости и быстродействия. Служит для повышения надёжности хранения данных и/или для повышения скорости чтения/записи.

В зависимости от способа организации избыточности различаются уровни RAID.

Часто используются следующие варианты (уровни) RAID-массивов.

**RAID 0:** данные (файлы) одновременно записываются на несколько дисков, так что каждый файл физически распределяется по различным устройствам HDD (*stripping*). Повышается скорость операций чтения/записи, но избыточности нет, и надёжность уменьшается. Минимальное количество дисков – 2.

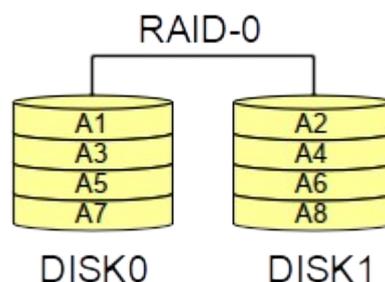


Рис. 16. Схема организации массива RAID 0

**RAID 1:** каждый два диска являются копиями друг друга (*mirroring*). При выходе из строя одного из дисков пары все данные без потерь

остаются на втором. Повышается надежность (максимальная избыточность), но выигрыша в производительности нет. Минимальное количество дисков – 2.

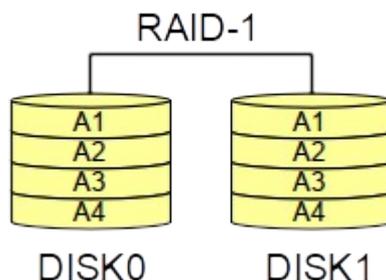


Рис. 17. Схема организации массива RAID 1

**RAID 3:** Один из дисков выделяется для хранения **контрольных сумм** данных, распределенных по остальным дискам. При выходе из строя одного из дисков данные могут быть восстановлены на основе оставшейся информации и контрольных сумм. Повышается надежность, есть выигрыш в производительности. Минимальное количество дисков – 3. Недостаток – повышенная нагрузка на диск с контрольными суммами.

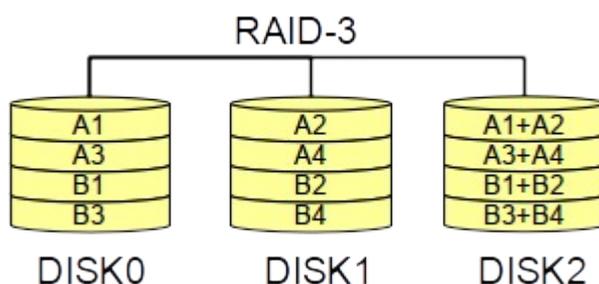


Рис. 18. Схема организации массива RAID 3

**RAID 5:** Контрольные суммы и данные распределены по всем дискам. При выходе из строя одного из данных могут быть восстановлены на основе оставшейся информации и контрольных сумм. Повышается надежность, есть выигрыш в производительности. Минимальное количество дисков – 3. Нагрузка на все диски равномерная.

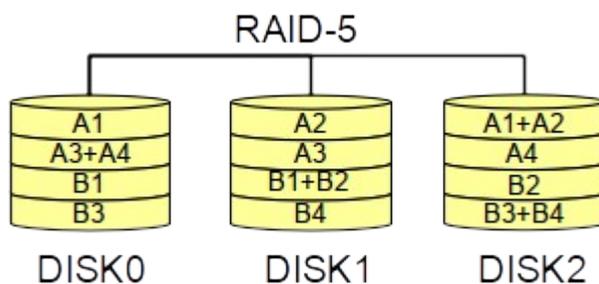


Рис. 19. Схема организации массива RAID 5

**RAID 6:** Контрольные суммы и данные распределены по всем дискам. Для данных вычисляется две различные контрольные суммы. Устойчив к одновременному выходу из строя половины дисков. Минимальное количество дисков – 4. Нагрузка на все диски равномерная.

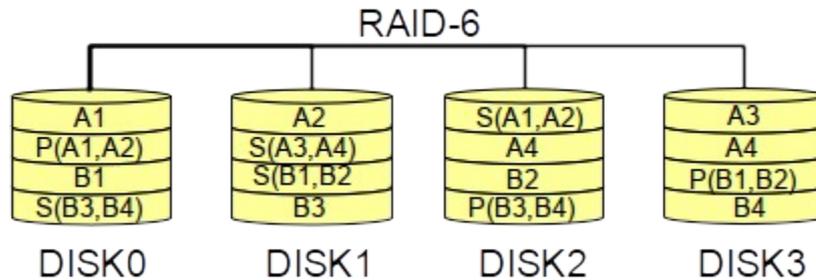


Рис. 20. Схема организации массива RAID 6

### Резервирование данных в кластерных системах

**Резервное копирование** (резервирование, *backup*) – процесс создания копии данных на носителе (жёстком диске, магнитной ленте и т.д.), предназначенном для восстановления данных в *оригинальном месте их расположения* в случае их повреждения или разрушения, соответствующими программами – резервными дубликаторами данных.

Различают *полное* резервирование и *инкрементальное* резервирование.

**Архивация** – подготовительная обработка, которая может включать сжатие данных, для долгосрочного хранения или передачи их по сети.

Архивация состоит из двух этапов: превращение набора файлов в один файл, а затем сжатие (компрессия).

В сетях предприятий и в кластерных центрах обработки данных (ЦОД) используются специализированные **серверы резервного копирования** с соответствующим программным обеспечением. На пользовательских узлах должны в таком случае устанавливаться **агенты (клиенты)** системы резервного копирования.

Резервирование производится автоматически и по расписанию (раз в сутки – инкрементальное, раз в неделю – полное).

### Виртуализация в кластерных системах

**Виртуализировать** означает взять нечто одной формы и заставить его казаться другой формы. *Виртуализировать компьютер* означает заставить компьютер казаться сразу несколькими компьютерами или совершенно другим компьютером.

Виртуализацией также называется ситуация, когда несколько компьютеров представляются как один отдельный компьютер. Обычно это называют **серверным кластером** (*grid computing*).

Возможны следующие варианты (технологии) виртуализации:

- **Виртуализация оборудования:** возможность предоставления *гостевой операционной системе (ОС)* некоторого стандартного набора оборудования и ресурсов вне зависимости от оборудования и ресурсов *гипервизора (хост-системы)*. Условием является избыточность аппаратных ресурсов хост-системы. *Примеры:* «облачные» технологии, использование одновременно нескольких операционных систем на одних и тех же аппаратных средствах.

- **Виртуализация процессора:** создание «прослойки» между приложением и операционной системой основного компьютера так, что выполнение приложений не зависит от основной операционной системы. От основной ОС зависит только реализация «прослойки». Пример – «виртуальная машина Java» (JVM) и Java-приложения.

- **Командная виртуализация:** предоставление *трансляторам языков программирования* возможности трансляции программ на *машинный язык* процессора, не совпадающего с реальным процессором компьютера, на котором выполняется такая трансляция (например, системные программы для планшетных компьютеров на процессоре ARM делаются на настольных компьютерах с процессорами Intel или AMD).

## Соединение узлов в кластерных системах

Для подключения высокопроизводительных хранилищ данных (дисковых массивов, доступных для всех узлов кластера) используются технологии **SAN (Storage Area Network)**.

Для таких массивов требуется обеспечить высокую скорость обмена данными со всеми узлами кластера.

Используется два варианта соединений (интерфейсов и протоколов):

- **iSCSI (Internet Small Computer System Interface)** – протокол, который базируется на TCP/IP (стандартный сетевой протокол) и разработан для установления взаимодействия и управления системами хранения данных, серверами и клиентами (стандартизованным по RFC 3720). Системы на основе iSCSI могут быть построены на любой достаточно быстрой физической основе, поддерживающей протокол IP, например Gigabit Ethernet или 10G Ethernet.

- **Fibre Channel (волоконный канал)** – использует волоконно-оптические линии связи. Используется как стандартный способ подключения к системам хранения данных уровня предприятия. Протокол FC заменяет TCP на транспортном уровне.

Для соединения узлов кластера применяются как обычные компьютерные сети (с пропускной способностью от 2 Мбит/с для GRID-систем до 1000 Мбит/с для кластеров высокой доступности), так и специальные каналы связи, обеспечивающие максимально достижимую пропускную способность.

**InfinyBand** – высокоскоростная коммутируемая последовательная шина, применяющаяся как для внутренних (внутрисистемных), так и для межсистемных соединений.

Максимальная пропускная способность (в серийных системах) – около 300 Гбит/с.

В Единой автоматизированной информационной системе таможенных органов (ЕАИС ТО) кластерные системы и технологии виртуализации применяются в центрах обработки данных (ЦОД) на уровне центрального аппарата (Главный ЦОД Центрального информационно-технического таможенного управления — ЦИТТУ) и региональных таможенных управлений (РТУ). На уровне таможни центры обработки данных обычно делаются на основе отдельных серверов.

Системы высокопроизводительных вычислений традиционно сокращают как **НРС (High Performance Computing)**. Такие системы применяются для решения различного класса задач, требующих больших вычислительных мощностей:

- Моделирование в задачах гидро- и аэродинамики, проектирования и испытаний
- Обработка больших объёмов информации в короткое время
- Обеспечение поиска данных в сверхбольших хранилищах
- Шифрование и дешифрование больших объёмов информации в потоковом режиме.

В качестве примера вычислительных ресурсов и производительности суперкомпьютеров в таблице ниже приведены 10 наиболее производительных НРС-систем мира (источник: <http://www.top500.org/>).

Место	Страна	Название	Число ядер	TFLOP/S
1	Китай	Tianhe-2 (MilkyWay-2)	3120000	33863
2	США	Titan - Cray XK7	560640	17590
3	США	Sequoia - BlueGene/Q	1572864	17173
4	Япония	K computer	705024	10510
5	США	Mira - BlueGene/Q	786432	8587
6	США	Piz Daint - Cray XC30	115984	6271
7	США	Stampede - PowerEdge C8220	462462	5168
8	Германия	JUQUEEN - BlueGene/Q	458752	5009
9	США	Vulcan - BlueGene/Q	393216	4293
10	США	Cray XC30	225984	3143

## Задания для самостоятельной работы

Подготовить письменное сообщение (эссе) по указанному преподавателем варианту заданий из приведенного ниже списка. При подготовке сообщения выписать термины и сокращения, дать их объяснение и расшифровку.

1. Понятие производительности вычислительной системы. Субъективная и объективная производительность. Средства определения (тестирования) производительности.
2. Тесты SPEC. Общая характеристика. Результаты тестов SPECсри для различных классов компьютеров: ноутбуков, настольных ПК, серверов.
3. Определение производительности графической подсистемы в 2D- и 3D-режимах. Результаты тестов для конкретных моделей компьютеров.
4. Многоядерные процессоры. Зависимость производительности от количества ядер. Увеличение количества ядер со временем. Перспективы наращивания количества ядер в процессорах.
5. Многопроцессорные системы. Зависимость производительности от количества процессоров для различных процессорных архитектур. Многопроцессорные системы в суперкомпьютерах.
6. Факторы, определяющие быстродействие дисковой подсистемы. Дисковые массивы RAID. Варианты и особенности RAID-массивов.
7. Асинхронная схема организации компонентов современных компьютеров. Тактовая частота, частоты интерфейсных шин. Назначение и уровни кэш-памяти процессора.

8. Кластерные вычислительные комплексы. Организация кластеров. Взаимодействие узлов в кластерах.

## Компьютерные сети

### Понятие, назначение, состав

**Компьютерная сеть:** программно-аппаратный комплекс, создаваемый из отдельных устройств (узлов), соединенных *каналами (линиями) связи* для решения задач эффективного разделения ресурсов, повышения производительности и надежности вычислительных систем и обеспечения коммуникаций между пользователями.

### Состав компьютерной сети

- Компьютеры пользователей (станции)
- Серверы (специализированные компьютеры, предоставляющие какие-то ресурсы)
- Коммуникационное оборудование (средства для организации связи)
- Линии (каналы) связи
- Сетевые операционные системы
- Серверные программы (сервисы)

### Топологии компьютерных сетей

**Топологией сети** называется физическая или электрическая конфигурация кабельной системы и соединений сети (способ соединений). В топологии сетей применяют несколько специализированных терминов:

- **узел сети:** компьютер, либо коммутирующее устройство сети;
- **ветвь сети:** путь, соединяющий два смежных узла;
- **оконечный узел:** узел, расположенный в конце только одной ветви;
- **промежуточный узел:** узел, расположенный на концах более чем одной ветви;
- **смежные узлы:** узлы, соединенные, по крайней мере, одним путём, не содержащим никаких других узлов.

В топологии «шина» (рис. 21) **пропускная способность** канала связи делится между всеми узлами сети. Разрыв линии связи приводит к потере работоспособности сети.

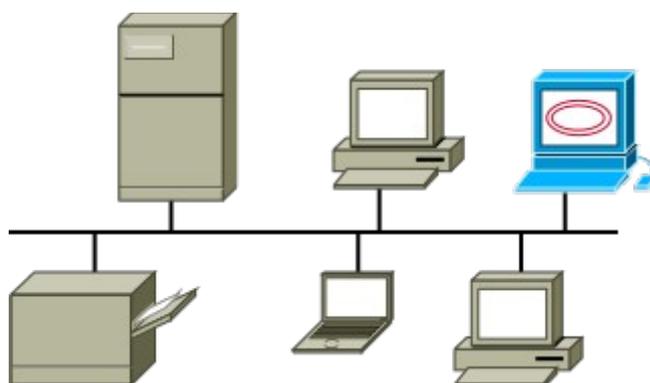


Рис. 21. Сетевая топология «шина»

В настоящее время такие топологии встречаются крайне редко.

Топология «звезда» (рис. 22) является более надежной, чем «шина». В этом случае разрыв одной линии приведет к отключению одного узла, а работоспособность сети в целом сохранится.

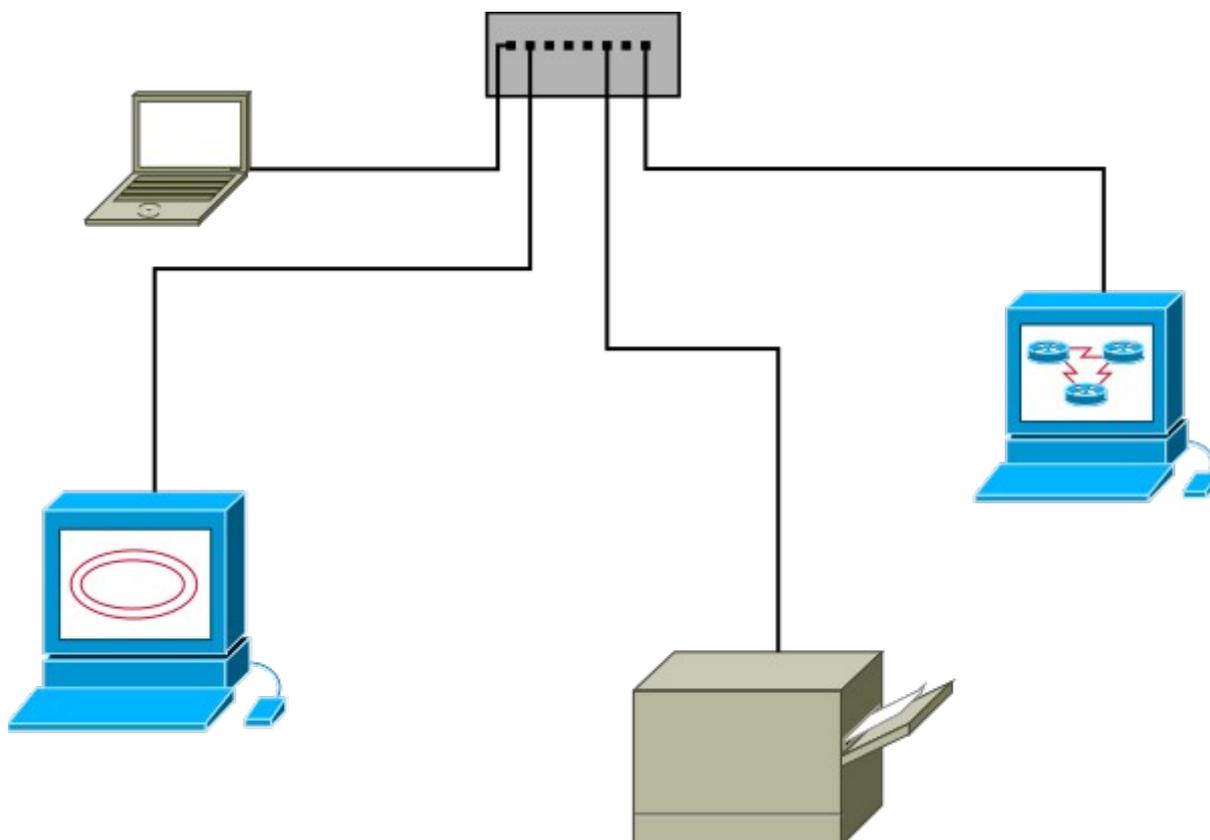


Рис. 22. Сетевая топология «звезда»

По топологии «звезда» организуются сети небольших рабочих групп или подразделений предприятий.

Топология «кольцо» (рис. 23) применяется для сетей, образующих инфраструктуру связи для территориально распределенных систем.

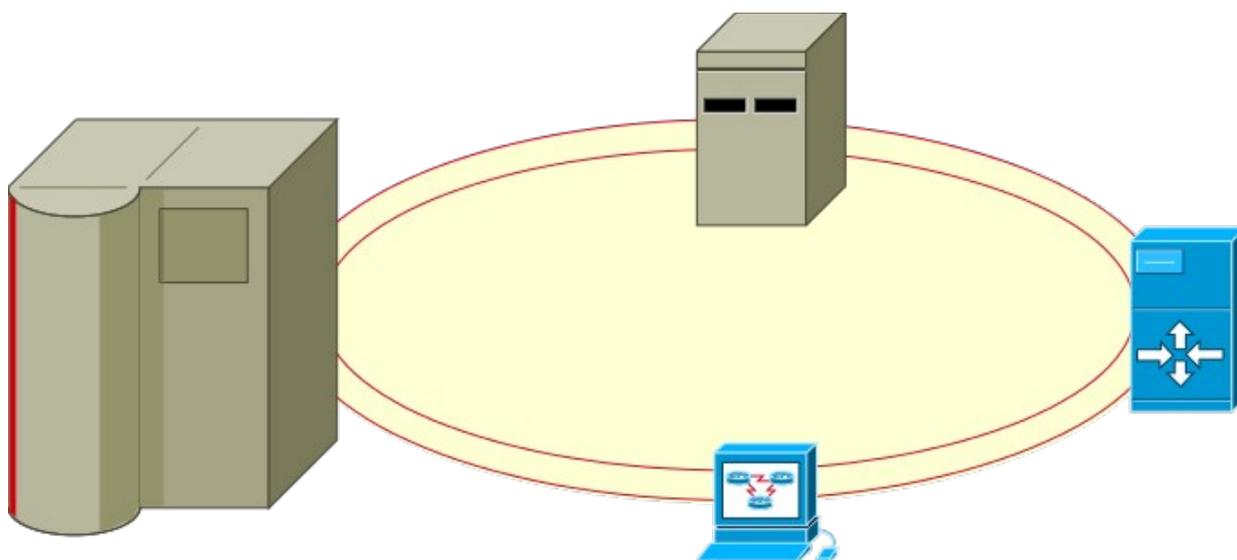


Рис. 23. Сетевая топология «кольцо»

В кольцевых сетях станции и серверы передают информацию в сеть только при получении специального сигнала – «маркера», который передается по кругу. Линии связи для надежности делают двойными (при этом обеспечивается *дуплексный режим* передачи данных).

Топология «дерево» (иерархическая звезда) (рис. 24) используется для организации сложных структурированных сетей. При этом ветви «дерева» образуют сети топологии «звезда».

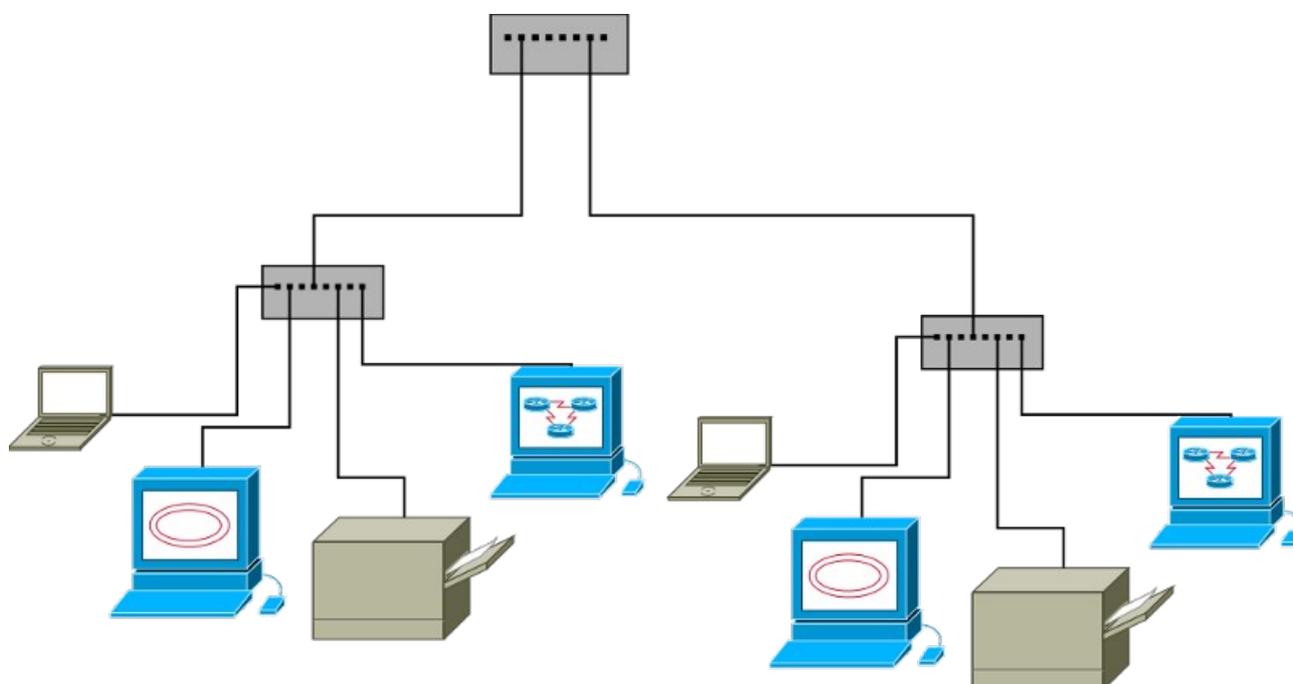


Рис. 24. Сетевая топология «дерево»

По такой схеме устроены все сети предприятий.

«Ячеистая» топология (mesh) (рис. 25) реализуется при соединении различных сетей разнообразными каналами связи. Использование нескольких каналов позволяет обеспечить балансировку нагрузки и повышенную надежность связи, даже если сами каналы являются ненадежными.

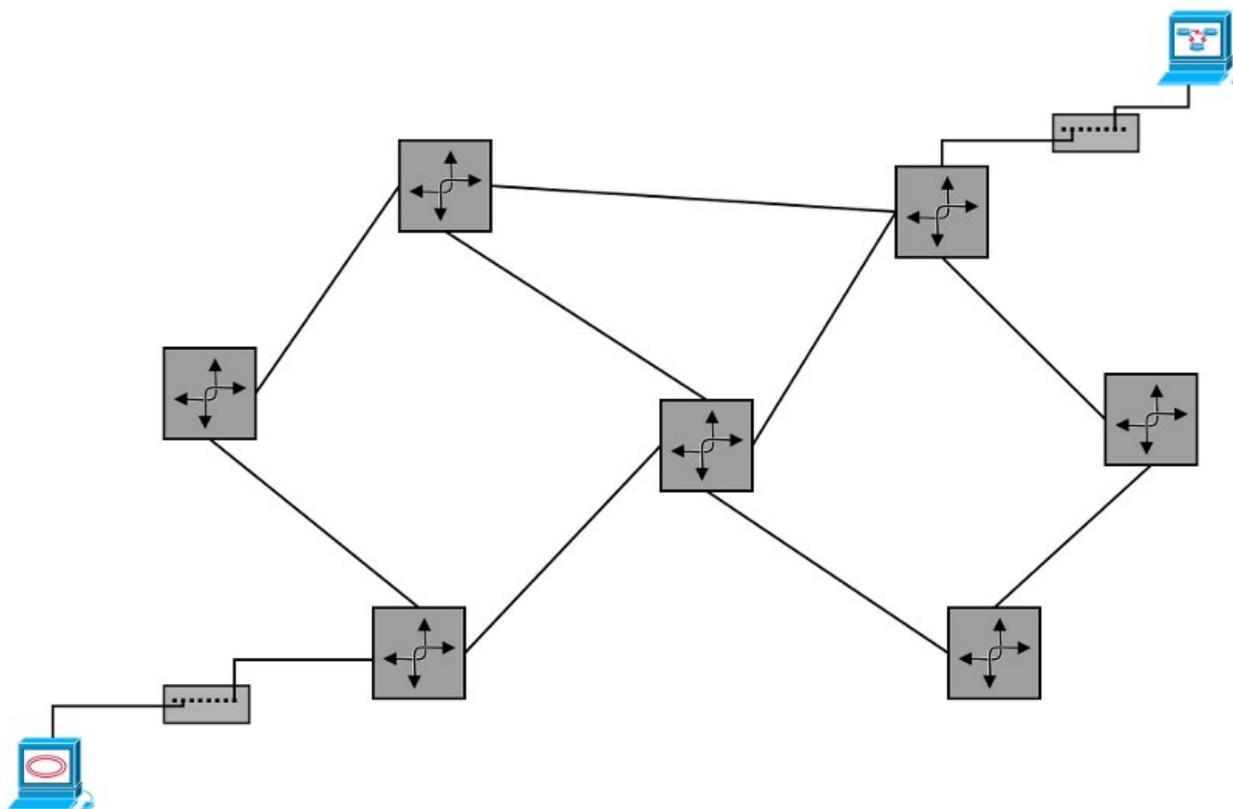


Рис. 25. «Ячеистая» сетевая топология

По такой схеме устроены глобальные компьютерные сети (Internet). Сигнал от одного узла к другому может идти по различным *маршрутам* в различные моменты времени.

## Серверы сети

### Понятие сервера

**Сервер** – обычный или специализированный компьютер, предназначенный для предоставления каких-либо ресурсов (см. раздел «Ресурсы вычислительных систем»). Поскольку сервер должен работать круглосуточно и при повышенной нагрузке, то аппаратные средства сервера создаются более надежными (и следовательно, более дорогими), а также предусматривается резервирование систем электропитания и внешней памяти.

С другой стороны, сервер — это специальный программный комплекс, обеспечивающий предоставление какого-либо ресурса средствами операционной системы.

Таким образом, можно сказать, что сервер — это специализированный программно-аппаратный комплекс, обеспечивающий предоставление одного или нескольких видов ресурсов.

## Виды серверов

- Файловый сервер
- Коммуникационный сервер
- Сервер доступа
- Сервер баз данных
- Сервер резервного копирования
- Медиа-сервер
- Почтовый сервер
- Сервер печати
- Сервер сканирования
- Терминальный сервер
- Web-сервер
- Сервер приложений
- Сервер точного времени
- и т.д..

## Авторизация пользователей сети

Для обеспечения **защиты информации** (в широком смысле) для пользователей сетей вводятся **права доступа** (привилегии) к каждому конкретному ресурсу (просматривать/ создавать/ изменять/ удалять/ записывать в). Для удобства организации пользователи объединяются в **группы**, для которых настраиваются **групповые привилегии** (политики). Пользователи автоматически *наследуют привилегии своей группы*, поэтому нет необходимости настраивать права для каждого пользователя, достаточно объявить, в какую группу он входит (если нужно, права пользователя могут быть настроены индивидуально).

**Авторизация** - это способ объявления вычислительной системе, что её ресурсами желает воспользоваться пользователь с определенными привилегиями. Для каждого пользователя создается **учетная запись** (account). Параметрами учетной записи являются **имя пользователя (login)**, **пароль**, **домашний каталог (home directory)** и **профиль** (набор настроек программного окружения).

## **Локальная авторизация**

Пользователь **идентифицируется** по имени и паролю **на конкретной станции сети** (узле). На этом узле размещается его домашний каталог и профиль. Доступ к ресурсам сети возникает в соответствии с разрешениями, зафиксированными групповой политикой и профилем. Если на другом узле сети нет учетной записи для этого пользователя, он не сможет пользоваться ресурсами такого узла.

## **Централизованная авторизация**

Все учетные записи и профили пользователей создаются на **сервере авторизации**. Пользователь может авторизоваться на любой станции сети, операционная система которой обращается к серверу за подтверждением наличия такого пользователя. Домашний каталог может создаваться либо на сервере, либо на конкретной станции (в зависимости от принятых политик). Таким образом, пользователь может работать с любого узла сети.

В Единой автоматизированной информационной системе таможенных органов (ЕАИС ТО) используется централизованная авторизация в доменной структуре в соответствии с организационной структурой таможенных органов — доменная система единой структуры каталогов (ДС ЕСК) ЕАИС ТО.

## **Адресация в компьютерных сетях**

В компьютерной сети компьютеры обмениваются информацией (пакетами) друг с другом. Чтобы передавать и принимать пакеты, нужно знать отправителя и получателя. Способ указания отправителя и получателя информации называется **адресацией**.

### **Можно выделить следующие виды адресации**

- физическая (практически всегда глобальная)
- логическая (локальная и глобальная).

**Физические адреса** являются машиночитаемыми (числовыми).

**Логические адреса** могут быть машиночитаемые (числовые) и человекочитаемые (символьные).

### **Физическая адресация**

**Физический адрес** – адрес физического интерфейса для **протоколов канального уровня** (Ethernet, FDDI, WiMAX ...). Теоретически является глобально уникальным.

Состоит из 48 битов (6 байтов), 12 цифр в 16-ричной системе счисления (например, *8a:44:c5:94:04:00*).

Устанавливается двумя способами:

- В постоянном запоминающем устройстве (ПЗУ) адаптера узла сети
- Программным путем при инициализации адаптера узла сети

### Локальные логические адреса

Логический локальный адрес присваивается компьютеру при настройке сетевого программного обеспечения. Этот адрес позволяет пользователям, работающим в сети, различать компьютеры.

В современных локальных сетях логический адрес представляет собой комбинацию букв и цифр и является **сетевым именем** компьютера (например, МКС-09).

Сетевое имя обычно связывается с конкретным рабочим местом (ролью рабочего места) и не зависит от компьютера, который установлен на этом рабочем месте. Таким образом, изменение аппаратных средств на рабочих местах пользователей локальной сети не влияет на работу других пользователей этой сети.

Для локальных сетей ПК в начале 90-х годов XX века IBM разработала протокол NetBEUI (NetBIOS Extended User Interface) для транспортного уровня сетевого стека **LAN Manager** (операционные системы IBM OS/2, потом Microsoft Windows for Workgroups и Microsoft WindowsNT). NetBEUI можно было эффективно использовать только в небольших локальных сетях (IBM разработала протокол NetBEUI для локальных сетей, содержащих до 200 рабочих станций). Протокол NetBEUI не маршрутизируемый, он не позволяет создавать глобальные сети, объединяя несколько локальных сетей.

Для протокола NetBEUI было введено понятие «Рабочая группа» (Workgroup). Сетевые имена должны быть уникальными в пределах рабочей группы. Первоначально ПК из различных рабочих групп не имели доступа друг к другу.

В настоящее время локальная логическая адресация поддерживается в «Службе доступа к файлам и принтерам сети Microsoft» (протокол NetBIOS over TCP/IP) и сервисами «Samba/CIFS» в POSIX-совместимых ОС. Рабочие группы являются «прозрачными» (ПК из различных рабочих групп имеют доступ друг к другу).

## Глобальная логическая адресация

В настоящее время существует единая система глобальной логической адресации, определяемой протоколом TCP/IP.

Адресация определяется на межсетевом уровне стека протоколов TCP/IP, параллельно существуют адреса IPv4 и IPv6.

IP-адрес – логический адрес, который записывается числами.

### Адреса IPv4

В версии IPv4 адрес состоит из 4-х чисел в десятичной системе, разделенных точками, например – **13.34.56.78**.

Каждое число, кроме первого, может принимать значения от 0 до 255, первое число не может быть 0. Таким образом, для IP-адреса в этом случае выделяется 4 байта.

Все возможные комбинации значений дают более 4 млрд. адресов.

$$2^8 \times 2^8 \times 2^8 \times 2^8 \Rightarrow 2^{32}$$

Однако существующие правила формирования IP-адресов ограничивают возможное количество адресов – получается менее 1 млрд.

Все возможные адреса разделяют на 4 класса адресов (сетей), которые отличаются друг от друга.

Классы сетей определяются:

- возможными значениями первого числа IP-адреса
- количеством чисел IP-адреса, выделенных для номера сети и номера компьютера.

Если обозначить буквой **N** «сетевую» часть IP-адреса, а буквой **C** «компьютерную» часть адреса, то можно составить определения классов.

- **Класс А:** 1-е число принимает значения в диапазоне от 1 до 126, структура адреса – **N.C.C.C**. Может быть 126 сетей класса А, но в каждой из них более 16 млн. адресов. Пример IP-адреса: **123.45.6.78**

- **Класс В:** 1-е число принимает значения от 128 до 191, структура адреса – **N.N.C.C**. Возможно  $63 \times 256 = 16128$  сетей класса В, а в каждой из них  $256 \times 256 = 65536$  адресов. Пример IP-адреса: **131.220.11.98**

- **Класс С:** 1-е число принимает значения от 192 до 223, структура адреса – **N.N.N.C**. Возможно  $31 \times 256 \times 256 = 2031616$  сетей класса С, но в каждой из них – максимум 256 адресов. Пример IP-адреса: **220.123.45.6**

## Особые адреса IPv4:

- **Класс D:** 1-е число принимает значения от 224, адреса этого класса не включаются в DNS
- **Адрес 127.0.0.1:** внутренний адрес сетевого адаптера, который присваивается ему автоматически при настройке программного обеспечения для работы по протоколу TCP/IP
- Группы адресов «частных сетей»:
  - Класс A: 10.0.0.0 – 10.255.255.255
  - Класс B: 172.16.0.0 – 172.31.0.0
  - Класс C: 192.168.0.0 – 192.168.255.0

Все эти адреса не назначаются компьютерам, реально работающим в Internet, и тем самым число возможных адресов еще больше ограничивается.

## Динамические адреса IPv4

Для каждого компьютера, работающего по протоколу TCP/IP, должен быть назначен IP-адрес. Однако это не означает, что адрес должен быть все время один и тот же.

**Статический адрес** (постоянный) назначается тем компьютерам, к которым должен быть постоянный доступ (серверам).

**Динамический адрес** (временный) можно назначать компьютерам, которые только пользуются какими-то возможностями (службами) сетей.

Для назначения статических адресов нужно настраивать ПО на каждом компьютере.

Для назначения динамических адресов создается специальный сервер адресов, который в соответствии с протоколом **DHCP** (Dynamic Host Configuration Protocol) выделяет адреса остальным компьютерам сети в момент загрузки ОС на этих компьютерах.

Использование DHCP является обоснованным в следующих случаях:

- В больших (корпоративных) сетях для упрощения управления компьютерами (не производится настройка ПО и установка IP-адресов для каждого компьютера сети)
- При сеансовом подключении к сети (dial-up) для всех видов модемов (экономия адресов). Пока клиент не подключился, ему адрес не нужен. Когда клиент отключился, адрес освобождается и может быть использован кем-то другим.
- В «домашних» сетях для упрощения подключения.

## Адресация IPv6

Поскольку адресация IPv4 ограничивает количество узлов в сети, предложено увеличить длину адреса с 4 байтов (32 бита) до 16 байтов (128 бит).

В новой системе адреса записываются числами в шестнадцатеричной системе группами до 4 чисел (до 16 бит), разделитель – двоеточие.

Существуют также полная и сокращенная формы записи.

- **Полная форма: 108В:0:0:0:8:800:200С:417А**
- **Сокращенная форма: 108В::8:800:200С:417А**

Последовательность «:» заменяет несколько нулевых групп.

В адресации IPv6 классы в понятии IPv4 отсутствуют.

Для обеспечения совместимости пространства адресов IPv6 и IPv4 приняты следующие соглашения:

- **127.0.0.1 => 0:0:0:0:0:0:0:1 (::1)** – «внутренний» адрес любого узла.
- **192.168.10.3 ==> 0:0:0:0:FFFF:192.168.10.3 (::FFFF:192.168.10.3)** – адрес пользовательского компьютера.

Процесс перехода на IPv6 требует обновления ПО всех узлов Интернет, формирующих инфраструктуру, поэтому проходит медленно (эволюционным путём).

## Глобальная логическая адресация DNS

Адреса IP являются машиночитаемыми, но не человекочитаемыми (особенно IPv6).

Поэтому была создана ещё одна система адресации (символьная) и обеспечено сопоставление символьных имен и числовых адресов.

Специальное ПО, которое занимается этим сопоставлением, называется «служба доменных имён» – DNS.

Все пространство Internet делится на области (домены, зоны) по принадлежности или по территории (стране).

Каждая зона имеет символьное имя. Имя для территориального домена (страны) – 2 символа, для домена по принадлежности – 3-4 символа. Это домены (зоны) первого уровня.

Внутри каждой зоны первого уровня происходит такое же деление (рис. 26), но нет ограничений на длину имени.

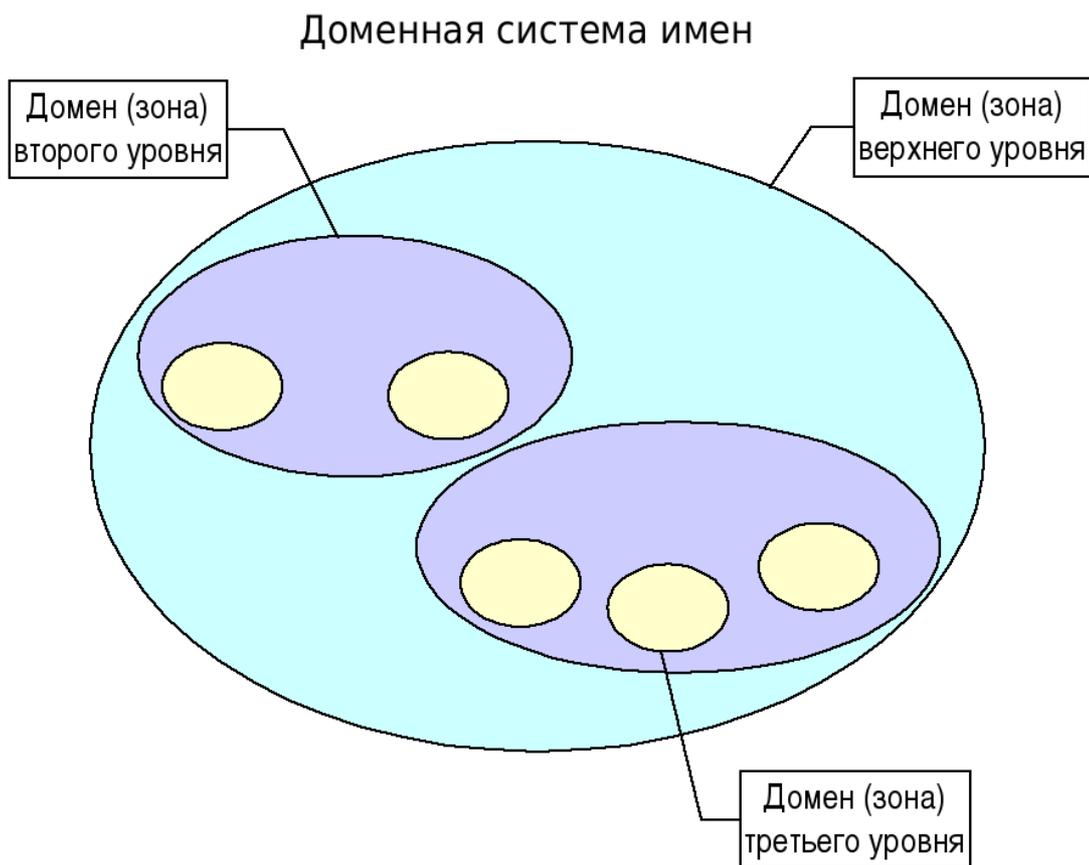


Рис. 26. Принцип построения доменной системы имен

### Полное доменное имя

Полное доменное имя строится из последовательности имен зон в порядке вложенности, разделенных точками, например **vot.nas.tam.net**.

Самая левая часть – имя конкретного компьютера (узла) – vot. Самая правая часть – имя домена верхнего уровня – net

В промежутке – имена доменов по уровням вложенности.

Один и тот же узел может иметь несколько доменных имен (например: **pochta.ru = fromru.com = pisem.net = mailru.com**), а может при одном доменном имени иметь несколько IP-адресов (если он сервер Internet).

В 2010 г. некоммерческая организация ICANN (Интернет-корпорация по управлению доменными именами и цифровыми адресами в Интернете) разрешила четырём странам использовать в именах доменов национальные кодировки.

Первые страны, которые получили право (и возможность) создавать домены в национальных кодировках — Саудовская Аравия, Египет, Россия, ОАЭ.

В мае 2010 г. начала работать зона **рф**.

Применение имён доменов в национальных кодировках имеет явные достоинства, в частности, отсутствие необходимости переключения раскладок и расширение пространства «имён».

В то же время исключительно использование национальных алфавитов при именовании информационных ресурсов приводит к ещё большему снижению культуры пользователей, создает технологические проблемы перекодирования и обновления браузеров, почтовых программ и ПО серверов DNS, а также влечет за собой изоляцию сегментов глобальной сети друг от друга.

### **Адресация ресурсов и пользователей**

Для унификации адресации ресурсов и пользователей в глобальных вычислительных сетях введены два вида идентификаторов:

- **URL** – универсальный указатель ресурса.
- **URI** – универсальный идентификатор ресурса.

Варианты URL:

- <http://www.ifmo.ru> – протокол HTTP, адрес сайта
- <ftp://ftp.altlinux.ru> (равносильно <ftp://anonymous@ftp.altlinux.ru>) – протокол FTP, ресурс для «скачивания» файлов
- <mailto:amin@site.me> – служба электронной почты

Пользователи указываются по наличию учетной записи на каком-либо узле сети: [vova@mysite.com](mailto:vova@mysite.com) (адрес электронной почты).

### **Служба (сервис) WHOIS**

WHOIS – сетевой протокол прикладного уровня, базирующийся на протоколе TCP. Предназначен для получения регистрационных данных о владельцах доменных имён, IP-адресов и автономных систем.

Протокол используется для доступа к публичным серверам баз данных (БД) **регистраторов IP-адресов** и **регистраторов доменных имён**. Регистраторы доменных имен создают поисковые веб-формы, доступные пользователям на многих сайтах в Интернете.

### **Управление Интернетом**

В задачи управления Интернетом входят:

- Распределение адресов сетей
- Распределение доменных имен

Структура управления имеет иерархический характер:

- Международные организации (ICANN, IANA) — верхний уровень
- Региональные регистраторы (в России – RU.Center)
- Провайдеры Интернет
- Администраторы сетей в организациях.

В Единой автоматизированной информационной системе таможенных органов (ЕАИС ТО) используется адресация IPv4 и адресация в домене customs.ru с именами доменов третьего уровня, соответствующими элементам организационной структуры.

## Прикладной уровень TCP/IP

С каждым протоколом прикладного уровня (сервисом, службой) стека протоколов TCP/IP связан некоторый числовой идентификатор, который используется в заголовке пакета (вместе с IP-адресом узла) для адресации конкретного приложения (и конкретного, принадлежащего ему, сетевого соединения). Такой идентификатор называется «номер порта» или просто «порт». (*Говорят «служба X работает на порту N».*)

Наиболее часто используемые порты приведены в таблице ниже.

Номер порта	Служба, протокол
21	FTP – протокол передачи файлов (File Transfer Protocol)
22	SSH – защищённый протокол удалённого управления (Secure SHell)
23	TELNET – незащищённый протокол удалённого управления (используется крайне редко)
25	SMTP – протокол передачи почты почтовыми агентами (Simple Mail Transfer Protocol)
53	DNS – протокол обмена для службы доменных имён (Domain Name Service)
80	HTTP – протокол обмена между браузерами и web-серверами (Hyper Text Transfer Protocol)
110	POP3 – протокол взаимодействия почтового сервера и клиента (Post Office Protocol v3)
139	NetBIOS-SSN – протокол службы доступа к файлам и принтерам сети Microsoft (Samba)
143	IMAP – протокол взаимодействия почтового сервера и клиента (Internet Mail Access Protocol v2 и v4)

Номер порта	Служба, протокол
443	HTTPS – Защищённый HTTP
465	SMTPS – Защищённый SMTP
993	IMAPS – Защищённый IMAP
995	POP3S – Защищённый POP3

## Сервисы и серверы

В данном случае **сервер** – программный комплекс, обеспечивающий работу сетевого сервиса (т.е. обрабатывающий запросы на какой-либо порт).

Наиболее часто используемые сервисы и соответствующее серверное ПО приведены в таблице ниже.

Служба, протокол	ПО сервера	Условия распространения
<b>FTP</b> – протокол передачи файлов (File Transfer Protocol)	WuFTPd, vsftpd	Свободное ПО
	Microsoft Internet Information Service (IIS)	Несвободное ПО
<b>SSH</b> – защищённый протокол удалённого управления (Secure SHell)	OpenSSH	Свободное ПО
<b>SMTP</b> – протокол передачи почты почтовыми агентами (Simple Mail Transfer Protocol)	postfix, exim, sendmail	Свободное ПО
	Microsoft Exchange Server	Несвободное ПО
<b>POP3</b> – протокол взаимодействия почтового сервера и клиента (Post Office Protocol v3)	pop3d, dovecot	Свободное ПО
	Microsoft Exchange Server	Несвободное ПО
<b>IMAP</b> – протокол взаимодействия почтового сервера и клиента (Internet Mail Access Protocol v2 и v4)	Cyrus-IMAP, dovecot	Свободное ПО
	Microsoft Exchange Server	Несвободное ПО
<b>HTTP</b> – протокол обмена между браузерами и web-серверами (Hyper Text Transfer Protocol)	Apache Web Server, lighthttpd, nginx	Свободное ПО
	Microsoft Internet Information Service (IIS)	Несвободное ПО
<b>DNS</b> – протокол обмена для службы	BIND (named)	Свободное ПО

Служба, протокол	ПО сервера	Условия распространения
доменных имён (Domain Name Service)	Microsoft Internet Information Server (IIS)	Несвободное ПО
NetBIOS-SSN – протокол службы доступа к файлам и принтерам сети Microsoft	Samba	Свободное ПО

## Межсетевые экраны

**Межсетевой экран** (брандмауэр, firewall) – программное средство, работающее на *компьютере-шлюзе* или на *сервере доступа*, позволяющее запретить доступ к определённым портам (т.е. по определённым протоколам) шлюза или сети, находящейся за шлюзом. Может также разрешать доступ к определённым портам некоторых узлов за шлюзом при условии прохождения пакетов именно через этот шлюз («проброс портов»).

Для межсетевого экрана создаются **правила**, описывающие для каких адресов какие порты следует закрыть или открыть.

Пример правил:

- Запретить обращение к порту 23 данного узла с любых адресов («закрыть порт для Telnet»)
- Запретить обращение к порту 22 данного узла с любых адресов («закрыть порт для SSH»)
- Разрешить обращение к порту 22 данного узла с адресов внутренней сети 192.168.1.0/24

*Обычно правила работают по «логическому И», последующее уточняет предыдущее, однако для разных реализаций принципы создания правил могут отличаться.*

Часто МСЭ встраиваются в сетевые устройства или делаются в виде специализированных программно-аппаратных комплексов.

## Протоколы защищенной связи

Для защищенной связи в системах коммуникаций используются семейства протоколов SSL и IPSec.

**SSL (англ. Secure Sockets Layer – уровень защищённых сокетов)** – криптографический протокол, который обеспечивает безопасность связи. В SSL используется асимметричная криптография для обмена ключами и симметричное шифрование для сохранения конфиденциальности.

Описывается спецификациями **RFC 2246** (англ.) и **RFC 2246** (рус.), работает на **прикладном уровне** стека TCP/IP (HTTPS, IMAPS, POP3S).

Если клиент и сервер решили использовать SSL, клиент обращается с запросом на соединение, а сервер посылает клиенту запрос на сертификат. Клиент подтверждает свою подлинность с помощью сертификата, после чего происходит обмен ключами симметричного шифрования для установки безопасного соединения. Клиент и сервер используют сеансовые ключи для шифрования и расшифрования данных, которые они посылают друг другу.

В SSL **шифруются данные**, из которых потом на транспортном уровне формируются пакеты.

**IPsec** (сокращение от **IP Security**) – набор протоколов для обеспечения защиты данных, передаваемых по межсетевому протоколу IP. Позволяет осуществлять подтверждение подлинности (аутентификацию), проверку целостности и/или шифрование IP-пакетов. IPsec также включает в себя протоколы для защищённого обмена ключами в сети Интернет.

Описывается спецификациями **RFC 2401 ... RFC 2412**. Работает на **межсетевом уровне** стека TCP/IP. Для подтверждения взаимной подлинности участников обмена данными и обмена секретными ключами используется **Internet Security Association and Key Management Protocol (ISAKMP)**.

IPsec может функционировать в двух режимах: **транспортном и туннельном**.

В **транспортном режиме** шифруются (или подписываются) только данные IP-пакета (дейтаграмма), исходный заголовок сохраняется. Транспортный режим как правило используется для установления соединения между узлами сети.

В **туннельном режиме** шифруется весь исходный IP-пакет: данные, заголовок, маршрутная информация, а затем он вставляется в поле данных нового пакета, то есть происходит *инкапсуляция*. Туннельный режим может использоваться для подключения удалённых компьютеров к **виртуальной частной сети** или для организации безопасной передачи данных через открытые каналы связи (например, Интернет) между шлюзами для объединения разных частей виртуальной частной сети.

## **Магистральные сети коммуникаций**

**Магистральная сеть связи** – транспортная телекоммуникационная инфраструктура для предоставления услуг связи. Как правило, магистральная сеть связи выстраивается на собственных или

арендованных волоконно-оптических линиях с использованием *канального оборудования связи*.

В состав канального оборудования входят

- Маршрутизаторы
- Мультиплексоры SDH/PDH

**Мультиплексор** – устройство (или программа), позволяющая передавать по одной коммуникационной линии или каналу передачи одновременно несколько различных потоков данных. В оптических каналах связи это обеспечивается с помощью пучков света с разными длинами волн и **дифракционной решетки**.

Технология передачи данных по большому числу каналов с использованием одного оптического волокна называется **DWDM** (плотное спектральное уплотнение, *dense wavelength-division multiplexing*).

**SDH (Synchronous Digital Hierarchy)** – стандарт для транспорта трафика в точках обмена трафиком (потоками данных) при соединении магистральных и местных каналов связи (*Internet Exchange – IX*).

SDH – термин, используемый в Европе. В США и Японии такая технология называется **SONET (Synchronous Optical NETwork)**

Основная скорость передачи – 155,250 Мбит/с (STM-1). Более высокие скорости определяются как кратные STM-1: STM-4 - 622 Мбит/с, STM-16 - 2488,32 Мбит/с, STM-64 - 9953,28 Мбит/с. (*STM – Synchronous Transport Module*).

Сложные сети обычно имеют многоуровневую структуру. Первый уровень – оборудование доступа пользователей. Этот уровень состоит из оборудования «последней мили» и, как правило, из мультиплексоров STM-1. Оборудование «последней мили» отвечает за доведение сигнала пользователей до мультиплексоров первого уровня. В роли оборудования «последней мили» обычно выступают так называемые оптические модемы, по сути являющиеся конверторами электрического сигнала в оптический и обратно. Мультиплексоры первого уровня собирают каналы пользователей для дальнейшей транспортировки. Следующий уровень могут составлять мультиплексоры уровня STM-4 и STM-16.

Основные операторы магистральных телекоммуникационных сетей в России перечислены в таблице ниже.

N	Оператор	Сеть, тыс. км
1	Ростелеком	500
2	МегаФон+Синтерра	118

N	Оператор	Сеть, тыс. км
3	МТС	117
4	ВымпелКом	136,6
5	ТрансТелеКом	75
6	Старт Телеком	16
7	Orange Business Services	8,5
8	Раском	6
9	RetnNet	5,7
10	ТелиаСонера Интернэшнл Кэрриер Раша	2

## Оборудование для организации сетей

Оборудование для компьютерных сетей в соответствующем контексте называется также «коммутационное» или «коммуникационное» оборудование. Выделяются следующие виды оборудования:

- Концентраторы (практически не используются)
- Мосты
- Коммутаторы
- Маршрутизаторы (роутеры)
- Шлюзы

**Сетевой мост** – устройство **канального** уровня модели OSI/RM. Обеспечивает связь двух подсетей (*сегментов*) с одинаковой *системой логической адресации*. Сетевой мост выполняет следующие функции:

1. Мост передает пакеты из одного сегмента в другой, если адресат (*физический адрес адаптера назначения*) находится в другом сегменте сети. Если адресат находится в том же сегменте, что и отправитель пакета, мост не пропускает пакет в другие сегменты.
2. Мост позволяет объединить сегменты с различными *средами передачи данных* (например, сегмент «витой пары» и беспроводной сегмент).
3. Мост не передает в другие сегменты ошибочные пакеты (слишком длинные, слишком короткие, пакеты коллизий).

**Коммутатор** — устройство **канального** уровня модели OSI/RM. Обеспечивает связь двух и более подсетей (*сегментов*) с одинаковой *системой логической адресации* («Коммутатор уровня 2»).

Современные коммутаторы имеют выделенные **высокоскоростные порты**, позволяющие создавать *магистральный (опорный) сегмент* в сети предприятия (*backbone*).

## Управляемые коммутаторы

**Управляемые коммутаторы** чаще всего используются для **сегментации сети**. Различные группы узлов (*хостов*) объявляются принадлежащими различным **виртуальным локальным сетям (VLAN)**. Хосты из различных VLAN не имеют связи друг с другом. Один или несколько портов коммутатора объявляются **транковым (trunk, tagged port)** – входящим в несколько VLAN. Через этот порт хосты из разных VLAN могут получать связь с сервером или другими сетями.

Сегментация может осуществляться

- По портам коммутатора (коммутаторы уровня 2, Level 2 Switch)
- По протоколам сетевого уровня (узлы, работающие по разным протоколам, автоматически недоступны друг для друга)
- По данным аутентификации пользователя или узла (имя, пароль, ключи).

В стандарте **IEEE 802.1Q** определены дополнительные биты в сетевом пакете 2-го уровня (кадре) для указания номера VLAN.

Управление (в большинстве случаев) осуществляется при посредстве встроенного web-сервера (с помощью браузера).

**Коммутаторы уровня 3 (Level 3 Switch)** анализируют не физические адреса, а сетевые (как правило, адреса IP). Все остальное - аналогично Level 2 Switch.

**Коммутаторы уровня 4 (Level 4 Switch)** анализируют не сетевые адреса, а транспортные («порты») протоколов TCP и UDP). Это позволяет снять часть нагрузки с серверов и межсетевых экранов.

## Маршрутизаторы

**Маршрутизаторы (роутеры)** – устройства 3 и 4 уровней модели OSI/RM.

Виртуальные частные сети обеспечивают поддержку различных протоколов, в особенности на прикладном уровне.

Роутер соединяет несколько IP-сетей, обеспечивает распределение TCP-пакетов по линиям с наименьшей загрузкой.

Современные роутеры имеют дополнительные функции **ДНСР-сервера, межсетевого экрана и трансляции локальных адресов (NAT)**.

Управление и настройка (в большинстве случаев) осуществляется при посредстве встроенного web-сервера (с помощью браузера).

## **Шлюзы**

**Шлюз** – программно-аппаратный комплекс, работающий на прикладном уровне, обеспечивает связь сетей. Шлюз обеспечивает выполнение нескольких функций, а именно

1. маршрутизация
2. разграничение передачи пакетов различных протоколов (межсетевой экран)
3. трансляция пакетов протоколов HTTP и FTP через определенный интерфейс («прокси-сервер»)
4. сегментация (разделение сетей) с помощью нескольких сетевых интерфейсов.

В Единой автоматизированной информационной системе таможенных органов (ЕАИС ТО) ведомственная интегрированная телекоммуникационная сеть (ВИТС) построена на каналах связи, арендуемых у магистральных операторов с использованием управляемых коммутаторов и другого сетевого оборудования. Для организации связи с участниками ВЭД используются специально выделенные шлюзы и межсетевые экраны, образующие автоматизированную систему внешнего доступа (АСВД) ЕАИС ТО.

## **Задания для самостоятельной работы**

Подготовить письменное сообщение (эссе) по указанному преподавателем варианту заданий из приведенного ниже списка. При подготовке сообщения выписать термины и сокращения, дать их объяснение и расшифровку.

1. Открытые системы как технологическая необходимость. Свойства открытых систем. Открытые спецификации и открытые интерфейсы.
2. Модель OSI/RM. Стандарты, определяющие данную модель. Уровни модели.
3. Интерфейсы и протоколы в модели OSI/RM. Понятие межпрограммных интерфейсов.
4. Физический и канальный уровни модели OSI/RM. Реализация взаимодействия на физическом и канальном уровне. Физическая адресация.
5. Сеансовый и сетевой уровни модели OSI/RM. Логическая адресация. Варианты логических адресов.

6. Транспортный уровень модели OSI/RM. Понятие «пакета». Способы контроля целостности пакетов. Протоколы TCP и UDP.
7. Логическая адресация IPv4. Принципы организации. Классы сетей и маски адресов. Частные сети. Адресация IPv6. Преобразование адресов.
8. Уровень представления и прикладной уровень модель OSI/RM. Понятие сетевого сервиса (службы). Порты протокола TCP/IP.
9. Физические принципы передачи электромагнитных сигналов. Понятия полного внутреннего отражения, волновода, спектра сигнала, дисперсии и затухания.
10. Модуляция электромагнитных сигналов. Виды модуляции. Применимость различных видов модуляции при передаче сигналов различной природы. Широтно-импульсная модуляция. Назначение, виды и особенности модемов.
11. Стандарты Ethernet. Характеристики проводных и беспроводных каналов связи. Принцип работы протокола CSMA/CD. Достоинства и ограничения технологии Ethernet.
12. Маркерные стандарты компьютерных сетей (TokenRing, ArcNet, FDDI). Особенности протоколов передачи данных. Применимость стандартов в современных условиях. Производители оборудования.
13. Понятие контрольной суммы пакета. Алгоритмы вычисления контрольных сумм. Доказательства однозначности контрольной суммы.
14. Размеры пакетов Ethernet, UDP и TCP. Информация, добавляемая при формировании TCP-пакетов. Способы проверки целостности TCP-пакетов.
15. Понятие топологии сети. Варианты топологий и среды передачи данных. Современные топологии локальных сетей. Понятие «backbone».
16. Варианты адресации в локальных сетях. Логические адреса. Адресация в сетях Novell Netware и Microsoft Windows. Адресация NetBIOS. Правила создания сетевых имен. Ограничения локальной адресации.
17. Физические адреса. Формирование физических адресов. Физический адрес в сетевом пакете Ethernet. Концентраторы и коммутаторы.

18. Коммутация сетевых пакетов. Коммутаторы уровня 2, 3 и 4. Характеристики коммутаторов. Основные производители. Управляемые и неуправляемые коммутаторы.
19. Понятие сегмента сети. Сегментация сети с помощью коммутаторов. VLAN — назначение, способы создания, варианты использования.
20. Характеристики локальных сетей. Протяженность локальных сетей. Схемы локальных сетей. Понятие « сетевого моста ».
21. Понятие « глобальной вычислительной сети ». Топологии глобальных сетей. Каналы связи в глобальных сетях. Магистральные сети передачи данных.
22. Стек протоколов TCP/IP и семиуровневая модель OSI/RM. Соотношение уровней. Организационные принципы назначения IP-адресов.
23. Доменная система имен. Нормативные документы и регуляторы назначения имен доменов. Домены в национальных кодировках. Понятия URL и URI, варианты написания, использование.
24. Маршрутизация в глобальных сетях. Принципы маршрутизации. Роутеры: назначение, характеристики, использование.
25. Шлюзы и прокси-серверы. Характеристики, разновидности. Варианты использования
26. Сети ADSL и HDSL. Требуемое оборудование, организация передачи данных. Протокол X.25.

## **Беспроводные сети**

### **Семейство стандартов сетей Wi-Fi**

Wi-Fi описывается стандартами IEEE 802.11.

- 802.11a – несущая частота 5 ГГц, скорость до 54 Мб/с
- 802.11b – несущая частота 2,4 ГГц, скорость до 11 Мб/с
- 802.11g – несущая частота 2,4 ГГц, скорость до 54 Мб/с
- 802.11n – несущие частоты 2,4-2,5 или 5 ГГц, скорость до 600

Мб/с (Используется в маршрутизаторах (роутерах) корпоративного класса, работает с устройствами 802.11a/b/g)

На **физическом уровне** в беспроводных сетях используются радиоволны с широтно-импульсной модуляцией.

На **канальном уровне** в зависимости от варианта стандарта применяются различные варианты формирования сигналов в выбранном частотном диапазоне (модуляция, кодирование и пр.).

## **Режимы работы сетей Wi-Fi**

Для организации сетей Wi-Fi требуется следующее оборудование:

- Wi-Fi адаптер
- Точка доступа (Access Point)

Беспроводные соединения могут осуществляться в следующих режимах:

- Прямое соединение (Ad-hoc)
- Стандартный режим (клиент – точка доступа)
- Беспроводной мост (две точки доступа)

**При прямом соединении** (Ad-hoc) не используются коммутационные устройства. Адаптеры (клиенты) соединяются друг с другом на основе указания сетевого имени.

Выбор маршрута прохождения сигналов в пределах общей сети осуществляется динамически (по самому сильному «ответу»). Такие сети называют «самоорганизующимися».

Могут использоваться не только для Wi-Fi, но и для других беспроводных соединений. Например, при достаточно большом количестве пользователей мобильных телефонов сотовая сеть становится не нужна, сигнал может проходить и ретранслироваться через аппараты других пользователей.

Сети Ad-hoc становятся очень востребованы в ЧС, когда элементы инфраструктуры (ретрансляторы, точки доступа и пр.) лишаются питания.

Передачи по таким сетям очень трудно контролировать.

**В стандартном режиме** работы Wi-Fi точка доступа является DHCP-сервером (в противном случае точка доступа и беспроводные адаптеры должны иметь разные статические адреса). При этом одна точка доступа может поддерживать до 8 клиентов (имеет значение вариант протокола клиента и вариант протокола точки доступа).

Незашифрованная (открытая) точка доступа называется **Hot Spot**.

Важно, что варианты шифрования трафика и формирования пароля должны быть согласованы между точкой доступа и клиентом.

**Беспроводной мост** используется при невозможности организации проводного соединения. Расстояние между точками составляет от сотен

метров до километров. При этом обоим беспроводным адаптерам точек доступа назначается один и тот же статический адрес IP.

## **Беспроводные сети WiMAX**

Особенности WiMAX описаны в стандарте IEEE 802.16/

Изначально стандарт IEEE 802.16 описывал связь на частотах от 10 до 66 ГГц, поэтому для связи требовалась «прямая видимость».

Более поздняя модификация – IEEE 802.16a – перешла в частотный диапазон от 2 до 11 ГГц.

Реализации WiMAX (**Wireless LAN**) – 802.16d (фиксированный WiMAX) и 802.16e (мобильный WiMAX).

Для мобильного WiMAX требуется «сотовая» топология базовых станций.

Фиксированный WiMAX характеризуется следующими параметрами каналов:

- Частоты: 3,5 и 5 ГГц
- Дальность: от 25 до 80 км
- Скорость передачи: до 75 Мбит/с

Мобильный WiMAX характеризуется следующими параметрами каналов:

- Частоты: от 2,3 до 3,8 ГГц
- Дальность: от 1 до 5 км
- Скорость передачи: до 40 Мбит/с

## **Шифрование в беспроводных сетях**

Шифрование обеспечивает защиту информации от «прослушивания». В Wi-Fi используются более простые алгоритмы (WEP, DES, AES), в WiMAX – более сложный (AES с периодически изменяющимся ключом, AES-CCK).

## **Задания для самостоятельной работы**

Подготовить письменное сообщение (эссе) по указанному преподавателем варианту заданий из приведенного ниже списка. При подготовке сообщения выписать термины и сокращения, дать их объяснение и расшифровку.

1. Способы защиты каналов связи от «прослушивания». Шифрование: общие принципы и виды. Понятие криптостойкости.
2. Симметричное шифрование. Виды шифров. Алгоритмы. Достоинства и недостатки симметричного шифрования.
3. Асимметричное шифрование. Виды шифров. Алгоритмы. Достоинства и недостатки асимметричного шифрования.
4. Протоколы IPsec. Уровни модели OSI/RM, затрагиваемые IPsec. Используемые алгоритмы шифрования.
5. Протоколы SSL. Уровни модели OSI/RM, затрагиваемые SSL. Используемые алгоритмы шифрования.
6. Сети VPN. Варианты организации взаимодействия. Требуемые программные средства. Основные поставщики в России. Оценка стоимости организации VPN-канала.
7. Понятие «сертификат», «доверенный сертификат», «корневой сертификат». Удостоверяющий центр: назначение и использование.

## Пример оснащения таможни

В таблице ниже перечислены аппаратные средства серверов, используемых в одной из таможен Северо-Западного таможенного управления и роли этих серверов.

Роль	Аппаратные средства
Сервер ТТП	Hewlett Packard (HP) ProLiant BL460c G7
Контроллер домена	Hewlett Packard (HP) ProLiant DL360 G5
Сервер баз данных	Hewlett Packard (HP) ProLiant DL380 G5
Сервер сбора ТД	Hewlett Packard (HP) ProLiant DL380 G5
Сервер данных «Коносаменты»	Hewlett Packard (HP) ProLiant BL460c G7
Сервер АИСТ-М	Hewlett Packard (HP) ProLiant DL380 G4
Сервер Скат-ТК	IBM X3650 M3
Сервер СУР	Hewlett Packard (HP) ProLiant DL380 G6
Сервер СПС «Консультант+»	Hewlett Packard (HP) ProLiant ML370 G4
Почтовый сервер	Hewlett Packard (HP) ProLiant DL380 G5
Сервер резервного копирования	Hewlett Packard (HP) ProLiant DL380 G5
Сервер Kaspersky Internet Security	Hewlett Packard (HP) ProLiant DL160 G5
Прокси сервер (Internet)	Hewlett Packard (HP) ProLiant DL360 G6
Дисковый массив СХД	IBM DS3524

Роль	Аппаратные средства
Ленточный накопитель СХД	Hewlett Packard MSL 6000

В качестве системного программного обеспечения используются операционные системы семейства Microsoft Windows Server Enterprise, в качестве сервера баз данных Microsoft SQL Server на уровне таможни (для программных средств и их компонентов, эксплуатируемых только на уровне таможни). На уровне региональных таможенных управлений и ГЦОД ЦИТТУ ФТС России используется серверы баз данных Oracle Database.

В качестве сетевого (коммуникационного) оборудования в ЕАИС ТО, как правило, используется оборудование компании Cisco различных классов.

Более подробно ЕАИС ТО и состав ее компонентов обсуждается в курсе «Информационные таможенные технологии».

## Литература

1. Бройдо В.Л., Ильина О.П. Вычислительные системы, сети и телекоммуникации / В. Л. Бройдо, О. П. Ильина .— 4-е изд .— СПб.: Питер, 2011 .— 560 с.
2. Таненбаум Э. Архитектура компьютера. 5-е изд. – Питер, 2012 – 848 с.
3. Таненбаум Э. Современные операционные системы. 3-е изд. – Питер, 2013 – 1120 с.
4. Точки Р.Дж., Уидмер Н.С. Цифровые системы. Теория и практика, 8-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2004. – 1024 с.
5. Галатенко В.А. Основы информационной безопасности: учебное пособие / В. А. Галатенко; под ред. акад. РАН В. Б. Бетелина. — Изд. 4-е .— М.: Интернет-Университет информационных технологий (ИНТУИТ.РУ): БИНОМ. Лаборатория знаний, 2010 .— 205 с.
6. Догадин Н.Б. Архитектура компьютера : учебное пособие. - БИНОМ. Лаборатория знаний, 2012. – 271 с.
7. Авдеев В.А. Периферийные устройства: интерфейсы, схемотехника, программирование. — ДМК Пресс, 2012. – 850 с.
8. Шаньгин В.Ф. Защита информации в компьютерных системах и сетях. — ДМК Пресс, 2012. — 592 с.

9. Таненбаум Э., Уэзеролл Д. Компьютерные сети. 5-е изд. – СПб.: Питер, 2013 – 960 с.
10. Панасенко С. Алгоритмы шифрования. Специальный справочник. – БХВ-Петербург, 2009 – 576 с.
11. Парфенов П.С. История и методология информатики и вычислительной техники. – Санкт-Петербург: СПбГУ ИТМО, 2010. – 141 с.
12. Ключев А.О., Ковязина Д.Р., Петров Е.В., Платунов А.Е. Интерфейсы периферийных устройств. – Санкт-Петербург: СПбГУ ИТМО, 2010. – 294 с.
13. Кириллов В.В. Архитектура базовой ЭВМ. – Санкт-Петербург: СПбГУ ИТМО, 2011. – 144 с.

**Миссия университета** – генерация передовых знаний, внедрение инновационных разработок и подготовка элитных кадров, способных действовать в условиях быстро меняющегося мира и обеспечивать опережающее развитие науки, технологий и других областей для содействия решению актуальных задач.

---

## **КАФЕДРА ТАМОЖЕННОГО ДЕЛА И ЛОГИСТИКИ**

Кафедра таможенного дела и логистики (ТДиЛ) Института международного бизнеса и права государственного образовательного учреждения высшего профессионального образования «Санкт-Петербургского государственного университета информационных технологий, механики и оптики» была образована в 2007 году. Кафедра ТДиЛ – единственная в России, которая готовит специалистов таможенного дела по стандартам Всемирной Таможенной Организации (ВТО) и имеет соответствующую аккредитацию ВТО. Среди членов кафедры есть как работники высшей школы, так и действующие сотрудники Федеральной таможенной службы. Кафедра осуществляет подготовку специалистов в области таможенного дела и логистики в соответствии с потребностями отрасли по специальности 38.05.02 (036401) «Таможенное дело» со специализацией «Таможенный менеджмент».

Хахаев Иван Анатольевич

**Вычислительные машины, сети и системы  
телекоммуникаций в таможенном деле**

**Учебное пособие**

В авторской редакции

Редакционно-издательский отдел Университета ИТМО

Зав. РИО

Н.Ф. Гусарова

Подписано к печати 20.05.2015г.

Заказ № 3301

Тираж 50 экз.

Отпечатано на ризографе