Оглавление

1. Ранг матрицы

http://www.mathelp.spb.ru/book1/rank.htm
http://ru.solverbook.com/spravochnik/matricy/rang-matricy/
http://mozgan.ru/Math/TxtAdjacentMinorsMatrix

1 2 -1 -2
2 4 3 0
-1 -2 6 6

##
###

2. Методы решения систем линейных уравнений

http://ru.solverbook.com/spravochnik/reshenie-uravnenij/reshenie-sistem-linejnyx-uravnenij/

##
###

3. Конечно-мерные линейные пространства. Связь между базисами.

Линейным (векторным) пространством называется множество V произвольных элементов, называемых векторами,
в котором определены операции сложения векторов и умножения вектора на число,
т.е. любым двум векторам u и v поставлен в соответствие вектор u+v, называемый суммой векторов u и v,
любому вектору v и любому числу λ из поля действительных чисел R поставлен в соответствие вектор λv,
называемый произведением вектора v на число λ; так что выполняются следующие условия:

1. $u + v = v + u \ \forall u, v \in V$ (коммутативность сложения);
2. $u + (v + w) = (u + v) + w \ \forall u, v, w \in V$ (ассоциативность сложения);
3. существует такой элемент $o \in V$, называемый нулевым вектором, что $v + o = v \ \forall v \in V$;
4. для каждого вектора v существует такой вектор $(-v) \in V$, называемый противоположным вектору v, что $v + (-v) = o$;
5. $\lambda(u + v) = \lambda u + \lambda v \ \forall u, v \in V, \forall \lambda \in R$;
6. $(\lambda + \mu)v = \lambda v + \mu v \ \forall v \in V, \forall \lambda, \mu \in R$;
7. $\lambda(\mu v) = (\lambda\mu)v \ \forall v \in V, \forall \lambda, \mu \in R$;
8. $1 \cdot v = v \ \forall v \in V$.

Линейное пространство V называется конечномерным, если существует такой набор векторов

$$B = (b1, b2, b3, ..., bn), bi \in V, i = 1 .. n,$$

что любой вектор $X \in V$ может быть выражен в виде

$$X = \sigma1\ b1 + \sigma2\ b2 + ... + \sigma n\ bn,$$

для некоторых действительных чисел σ1, σ2, ..., σn.
Данное выражение называется линейной комбинацией.

Множество B называется базисом, если составляющие его вектора являются линейно независимыми,
иными словами, ни один из векторов базиса нельзя выразить через оставшиеся.

Чисто векторов в базисе - размерность пространства.

В любом конечно-мерном линейном пространстве, вообще говоря, бесконечно много базисов:
чтобы понять это, достаточно умножить все вектора на фиксированное действительное число ω, отличное от нуля,
и получить другой базис.

С другой стороны, если есть два базиса B1, B2, то любой вектор из B2, поскольку он является вектором из V,
выражается через базис B1. И наоборот.
Это и является сущностью связи между базисами в конечно-мерном линейном пространстве.

Можно, таким образом, составить систему уравнений, содержащую так называемую __матрицу перехода__.
Пусть B1 = (b1_1, b1_2, b1_3, ..., b1_n) - первый базис,
B2 = (b2_1, b2_2, b2_3, ..., b2_n) - второй базис.

Тогда, в силу определения базиса, любой вектор из B2 является линейной комбинацией векторов из B1:

$$
\begin{aligned}
b2\_1 &= \sigma1\_1\, b1\_1 + \sigma1\_2\, b1\_2 + \ldots + \sigma1\_n\, b1\_n \\
b2\_2 &= \sigma2\_1\, b1\_1 + \sigma2\_2\, b1\_2 + \ldots + \sigma2\_n\, b1\_n \\
&\vdots \\
b2\_n &= \sigma n\_1\, b1\_1 + \sigma n\_2\, b1\_2 + \ldots + \sigma n\_n\, b1\_n
\end{aligned}
$$

Данное уравнение можно записать на языке матриц следующим образом:

$$
\begin{pmatrix} b2\_1 \\ b2\_2 \\ \vdots \\ b2\_n \end{pmatrix} = \begin{pmatrix} \sigma1\_1 & \sigma1\_2 & \ldots & \sigma1\_n \\ \sigma1\_1 & \sigma1\_2 & \ldots & \sigma1\_n \\ \vdots & \vdots & \ddots & \vdots \\ \sigma1\_1 & \sigma1\_2 & \ldots & \sigma1\_n \end{pmatrix} \begin{pmatrix} b1\_1 \\ b1\_2 \\ \vdots \\ b1\_n \end{pmatrix}
$$

Матрица, фигурирующая здесь, называется матрицей перехода.
Матрица обратного перехода - обратная матрица к матрице перехода.

##
###

4. Китайская теорема об остатках. Приложения теории чисел.

Китайская теорема об остатках.

Пусть p = p1 · p2 · ... · pk, где pi — попарно взаимно простые числа, a1, a2, ..., ak — произвольный набор целых чисел.
Тогда система сравнений
x = a1 (mod p1)
x = a2 (mod p2)
.
.
.
x = ak (mod pk)
имеет единственное решение по модулю p.

Это решение вычисляется следующим образом:
x = a1 · M1 · M1^(-1) + ... + ak · Mk · Mk^(-1),
где Mi = (p1 · ... · pk) / pi, Mi^(-1) - обратный элемент к Mi по модулю pi.

Приложения теории чисел:

1. В алгоритме RSA вычисления производятся по модулю большого числа n, представимого в виде произведения двух больших простых чисел.
Теорема позволяет перейти к вычислениям по модулю этих простых делителей, которые по величине уже порядка корня из n, а значит имеют в два раза меньшую битовую длину.
2. На основе китайской теоремы об остатках реализуются некоторые схемы разделения секрета в криптографии,
к примеру, схема Асмута — Блума.
3. Из теоремы следует мультипликативность функции Эйлера φ(n).
4. На теореме основывается алгоритм Полига — Хеллмана нахождения дискретного логарифма за полиномиальное время для чисел,
имеющих специальный вид.

##
###

5. Прямая и плоскость в пространстве: уравнения, условия взаимных расположений двух плоскостей, двух прямых, прямой и плоскости.

Уравнения плоскостей:

http://mathhelpplanet.com/static.php?p=uravneniya-ploskosti-cherez-tochku-perpendikulyarno-vektoru
http://mathhelpplanet.com/static.php?p=uravneniya-ploskosti-komplanarnoi-dvum-nekollinyearnym-vektoram
http://mathhelpplanet.com/static.php?p=uravneniya-ploskosti-cherez-tri-tochki

Уравнения прямой:

http://mathhelpplanet.com/static.php?p=uravneniya-pryamyh-v-prostranstve

Взаимное расположение плоскостей:

http://mathhelpplanet.com/static.php?p=vzaimnoe-raspolozhenie-ploskostyei

Взаимное расположение прямых, прямой и плоскости:

http://mathhelpplanet.com/static.php?p=vzaimnoe-raspolozhenie-pryamyh-v-prostranstve

##
###

6. Аксиоматическое определение вероятности. Следствия из аксиом теории вероятностей.

https://en.wikipedia.org/wiki/Probability_axioms // Намного более проще по содержанию, чем русский вариант в википедии, а также многие другие ресурсы Интернета.
https://nsu.ru/mmf/tvims/chernova/tv/lec/node8.html#SECTION000400 // С водичкой, зато на русском :)

##
###

7. Нормальное распределение. Его характеристики и свойства. Стандартное нормальное распределение. Сходимость по распределению. Асимптотическая нормальность. Центральная предельная теорема.

https://nsu.ru/mmf/tvims/chernova/tv/lec/node27.html#SECTION0007503 // Нормальное распределение
https://nsu.ru/mmf/tvims/chernova/tv/lec/node29.html#SECTION000770 // Его характеристики и свойства
https://nsu.ru/mmf/tvims/chernova/tv/lec/node59.html#SECTION0001320 // Сходимость по распределению
https://nsu.ru/mmf/tvims/chernova/tv/lec/node60.html#SECTION0001330 // ЦПТ

##
###

8. Точечное и доверительное оценивание параметрических функций. Методы получения точечных оценок для неизвестных параметров распределений: метод моментов, максимального правдоподобия, метод квантилей.

https://www.matburo.ru/ex_ms.php?p1=msmm // метод моментов, 2-й пример
https://www.matburo.ru/ex_ms.php?p1=msmmp // максимального правдоподобия, 2-й пример
http://www.machinelearning.ru/wiki/index.php?title=Квантиль // метод квантилей

https://studfiles.net/preview/5350806/page:5/

##
###

9. Функции нескольких переменных. Непрерывность. Дифференцирование. Экстремум функций двух переменных.

---> http://www.math24.ru/функции-нескольких-переменных.html

http://mathprofi.ru/chastnye_proizvodnye_primery.html

##
###

10. Определенный интеграл. Классы интегрируемых функций. Замена переменных в определенном интеграле.

---> http://www.math24.ru/определенный-интеграл-и-формула-ньютона-лейбница.html

Классы интегрируемых функций:

* Непрерывные на сегменте [a, b] функции интегрируемы на этом сегменте.
* Ограниченная на сегменте [a, b] функция f(x),
имеющая лишь конечное число точек разрыва, интегрируема на этом сегменте.
* Монотонная на сегменте [a, b] функция f(x) интегрируема на этом сегменте.

http://mathprofi.ru/metod_zameny_peremennoi.html

##
###

11. Числовые и функциональные ряды. Необходимые и достаточные условия сходимости.

Бесконечные ряды ---> http://www.math24.ru/бесконечные-ряды.html
Признаки сходимости ---> http://www.math24.ru/сходимость-рядов-и-признаки-сравнения.html
Признак Даламбера, Признак Коши ---> http://www.math24.ru/признаки-даламбера-и-коши.html
Функциональные ряды --->
https://ru.wikiversity.org/wiki/Функциональные_последовательности_и_ряды

##
###

12. Степенные ряды. Абсолютная, условная и равномерная сходимость. Свойства равномерной сходимости рядов.

---> http://www.math24.ru/степенные-ряды.html

Равномерная сходимость, свойства --->
https://ru.wikiversity.org/wiki/Функциональные_последовательности_и_ряды

##
###

13. Линейные уравнения с постоянными коэффициентами. Однородные и неоднородные уравнения. Методы решения.

---> http://www.math24.ru/линейные-уравнения-первого-порядка.html
http://www.math24.ru/однородные-уравнения.html

http://mathprofi.ru/kak_reshit_neodnorodnoe_uravnenie_vtorogo_poryadka.html

##
###

14. Уравнения в полных дифференциалах. Интегрирующий множитель.

---> http://www.math24.ru/уравнения-в-полных-дифференциалах.html

http://www.math24.ru/использование-интегрирующего-множителя.html

##
###

15. Дробно-линейное отображение и его свойства. Изоморфизмы дробно-линейных отображений.

Дробно-линейным преобразованием называется преобразование, выражающееся в виде частного двух функций

$$w = (az + b)/(cz + d),$$

причём надо считать $ad - bc \neq 0$, так как в противном случае дробь,
стоящая в формуле, будет сократимой и будет равна просто постоянному числу.

Решая уравнение относительно z, получим формулу для обратного преобразования, которое тоже будет дробно-линейным:

$$z = (-dw + b)/(cw - a).$$

Всякой точке комплексной плоскости z будет соответствовать определённая точка плоскости w и наоборот,
т. е. преобразование $w = f(z)$ преобразует всю плоскость, включая бесконечно далёкую точку, саму в себя.
Этот факт следует из обратимости преобразования.

Свойства дробно-линейного преобразования:

1) При дробно-линейном преобразовании "окружность" преобразуется "окружность". Под "окружностью" здесь понимается прямая или окружность, поскольку прямую можно рассматривать как окружность,
проходящую через бесконечно далёкую точку.
2) Суперпозиция дробно-линейных отображений будет также дробно-линейным отображением.
3) Функция, обратная дробно-линейной, также будет дробно-линейной.
4) Каковы бы ни были три различные точки z1, z2, z3 и три различные точки w1, w2, w3, существует, и притом только одно, дробно-линейное отображение L, что L(zi) = wi, при i = 1, 2, 3.
Примером такого преобразования является:

$$[(z - z1) \cdot (z3 - z2)] / [(z - z2) \cdot (z3 - z1)] = [(w - w1) \cdot (w3 - w2)] / [(w - w2) \cdot (w3 - w1)]$$

Изоморфизмы дробно-линейных отображений.

На основании свойства 4 следует, что любой круг на комплексной плоскости
можно при помощи некоторого дробно-линейного преобразования преобразовать в любой другой круг.
Это следует из того, что любая окружность однозначно определена 3-мя своими точками (из школьной геометрии).

##
###

16. Вычеты. Вычисление интегралов с помощью вычетов.

---> http://mathhelpplanet.com/static.php?p=vychety-i-ikh-primeneniye
http://mathhelpplanet.com/static.php?p=vychisleniye-integralov-s-pomoshchyu-vychetov

http://www.allmath.ru/highermath/mathanalis/matan/matan12.htm

##
###

17. Булевы функции: основные тождества, СДНФ и СКНФ, полиномы Жегалкина, замкнутые классы T0, T1, S, L, M. Полная система булевых функций, базис, критерий полноты (формулировка).

Булевы функции, основные тождества, СДНФ и СКНФ, полиномы Жегалкина --->

https://neerc.ifmo.ru/wiki/index.php?title=Определение_булевой_функции

Тождества: https://docplayer.ru/42539075-Tozhdestva-bulevoy-algebry.html(5 страница)

Замкнутые классы, Полнота, Критерий --->

https://neerc.ifmo.ru/wiki/index.php?title=Полные_системы_функций._Теорема_Поста_о_полной_системе_функций

##
###

18. Выводимость формулы из гипотез в исчислении высказываний и исчислении предикатов. Метод резолюций для проверки выводимости формулы из гипотез.

Исчисление высказываний ---> http://mathhelpplanet.com/static.php?p=formalizovannoye-ischisleniye-vyskazyvaniy

Исчисление предикатов ---> http://mathhelpplanet.com/static.php?p=formalizovannoye-ischisleniye-predikatov

Метод резолюций ---> http://ipo.spb.ru/journal/content/931/Метод резолюций.pdf
(4 страница).

##
###

19. Функции, вычислимые и невычислимые по Тьюрингу. Тезис Черча-Тьюринга. Алгоритмически неразрешимые проблемы, примеры.

Функции, вычислимые и невычислимые по Тьюрингу --->
http://mathhelpplanet.com/static.php?p=razreshimost-i-perechislimost-mnozhestv

Тезис Черча-Тьюринга ---> https://ru.wikipedia.org/wiki/Тезис_Чёрча_—_Тьюринга

(Суть тезиса: Машина Тьюринга способна вычислить всё, что вычислимо в принципе).

Алгоритмически неразрешимые проблемы, примеры --->
http://mathhelpplanet.com/static.php?p=nerazreshimyye-algoritmicheskiye-problemy
(проблема о нумерации алгоритмов, задача об останове:
Даны описание процедуры и её начальные входные данные, требуется определить, завершится ли когда-либо выполнение процедуры с этими данными.
Альтернативой этому является то, что она работает всё время без остановки.
+ Последний параграф - Другие примеры алгоритмической неразрешимости).

##
###

20. Экстремальные задачи теории графов: минимальное остовное дерево, кратчайший путь между вершинами, задача коммивояжера. Точные и приближенные алгоритмы для их решения: алгоритм Дейкстры, «жадные» алгоритмы.

1) Минимальное остовное дерево.
Пусть дан связный (то есть из любой вершины есть путь в любую другую) неориентированный граф G = (V, E),
где V - множество вершин, E - множество рёбер, каждое из которых имеет вид (v, u), где v, u ∈ V.
Каждое ребро имеет вес, который выражается функцией w(v, u), -- такая функция выражает "стоимость" прохождения по ребру (v, u).
К примеру, в компьютерных сетях, это может быть битрейт данной двухточечной связи (канала, link).
Примем далее, что чем меньше "стоимость", тем лучше.
Минимальное остовное дерево - это ациклическое множество рёбер, содержащее все вершины и имеющее минимальный суммарный вес рёбер.
Поскольку множество рёбер ациклическое, оно является деревом по определению.
Задача нахождения минимального остовного дерева - это задача о нахождении такого множества.
Алгоритм построения (алгоритм Прима):

Искомый минимальный остов строится постепенно, добавлением в него рёбер по одному.

Изначально остов полагается состоящим из единственной вершины (её можно выбрать произвольно).

Затем выбирается ребро минимального веса, исходящее из этой вершины, и добавляется в минимальный остов.

После этого остов содержит уже две вершины, и теперь ищется и добавляется ребро минимального веса, имеющее один конец в одной из двух выбранных вершин,

а другой — наоборот, во всех остальных, кроме этих двух. И так далее, т.е. всякий раз ищется минимальное по весу ребро,

один конец которого — уже взятая в остов вершина, а другой конец — ещё не взятая, и это ребро добавляется в остов (если таких рёбер несколько, можно взять любое).

Этот процесс повторяется до тех пор, пока остов не станет содержать все вершины (или, что то же самое, n - 1 ребро).

В итоге будет построен остов, являющийся минимальным. Если граф был изначально не связен, то остов найден не будет (количество выбранных рёбер останется меньше n-1).

2) Кратчайший путь между вершинами.
Дан ориентированный или неориентированный взвешенный граф с n вершинами и m рёбрами.
Веса всех рёбер неотрицательны. Указана некоторая стартовая вершина s.
Требуется найти длины кратчайших путей из вершины s во все остальные вершины, а также предоставить способ вывода самих кратчайших путей.
Эта задача называется "задачей о кратчайших путях с единственным источником" (single-source shortest paths problem).
Алгоритм построения (алгоритм Дейкстры):
Заведём массив d[], в котором для каждой вершины v будем хранить текущую длину d[v] кратчайшего пути из s в v.
Изначально d[s] = 0, а для всех остальных вершин эта длина равна бесконечности (при реализации на компьютере обычно в качестве бесконечности выбирают просто достаточно большое число, заведомо большее возможной длины пути):

$$d[v] = \infty, \text{ при } v \neq s.$$

Кроме того, для каждой вершины v будем хранить, помечена она ещё или нет, т.е. заведём булевский массив u[]. Изначально все вершины не помечены, т.е.

$$u[v] = \text{false } \forall v.$$

Сам алгоритм Дейкстры состоит из n итераций. На очередной итерации выбирается вершина v с наименьшей величиной d[v] среди ещё не помеченных, т.е.:

d[v] = min d[p]
p: u[p]=false

(Понятно, что на первой итерации выбрана будет стартовая вершина s).
Выбранная таким образом вершина v отмечается помеченной.
Далее, на текущей итерации, из вершины v производятся релаксации: просматриваются все рёбра (v, to), исходящие из вершины v,
и для каждой такой вершины to алгоритм пытается улучшить значение d[to]. Пусть длина текущего ребра равна len,
тогда в виде кода релаксация выглядит как:

d[to] = min(d[to], d[v] + len).

На этом текущая итерация заканчивается, алгоритм переходит к следующей итерации (снова выбирается вершина с наименьшей величиной d, из неё производятся релаксации, и т.д.).
При этом в конце концов, после n итераций, все вершины графа станут помеченными, и алгоритм свою работу завершает.
Утверждается, что найденные значения d[v] и есть искомые длины кратчайших путей из s в v.

3) Задача коммивояжера.
Коммивояжер должен выйти из первого города, посетить по разу в неизвестном порядке города 2, 3, …, n и вернуться в первый город.
Расстояния между городами известны. В каком порядке следует обходить города, чтобы замкнутый путь (тур) коммивояжера был кратчайшим?
В терминах теории графов задача формулируется следующим образом:
Пусть дан полный (то есть все вершины графа соединены ребром) неориентированный граф G = (V, E) и весовая функция w(v, u).
Требуется найти гамильтонов цикл, то есть простой (не содержащий некоторой рёбро или вершину, за исключением начальной, несколько раз) цикл,
соединяющий все вершины графа.
Самый очевидный алгоритм решения задачи коммивояжера — жадный: из текущего города идти в ближайший из тех, куда ещё не ходил.

##
###

21. Комбинаторные операции: сочетания и размещения (с возвращением и без возвращения элементов). Комбинаторные принципы: сложение, умножение, дополнение, включение-исключение. Бином Ньютона. Полиномиальная формула.

Комбинаторные операции ---> https://nsu.ru/mmf/tvims/chernova/tv/lec/node3.html

Комбинаторные принципы ---> http://ya-znau.ru/znaniya/zn/80
https://neerc.ifmo.ru/wiki/index.php?title=Формула_включения-исключения

Бином Ньютона, полиномиальная формула ---> http://hijos.ru/izuchenie-matematiki/algebra-10-klass/20-binomialnaya-i-polinomialnaya-formuly/

С возвращением, с учетом порядка: n^k
С возвращением, без учета порядка: C^(n-1)_n+k-1
Без возвращения, с учетом порядка: n!/(n-k)!
Без возвращения, без учета порядка: n!/k!(n-k)!

##
###

22. Алфавитное кодирование: необходимое и достаточные условия однозначности декодирования. Теорема и алгоритм Маркова. Коды Хаффмана и Хэмминга.

Алфавитное кодирование --->
https://neerc.ifmo.ru/wiki/index.php?title=Кодирование_информации
Алфавитное кодирование - это кодирование, выполняющееся над отдельными символами исходного алфавита.

Достаточные условие однозначности декодирования - префиксный или суффиксный код.

Необходимое условие однозначности декодирования --->
https://neerc.ifmo.ru/wiki/index.php?title=Неравенство_Макмиллана

Теорема и алгоритм Маркова ---> http://kpolyakov.blogspot.com/2012/10/blog-post.html

Теорема Маркова:
Для любой схемы алфавитного кодирования Σ существует такое число N_Σ, что схема Σ является однозначно декодируемой тогда и только тогда, когда однозначно декодируемы все исходные данные,
длины которых не превосходят N_Σ.

Коды Хаффмана и Хэмминга ---> https://habr.com/post/140611/
https://habr.com/post/144200/

##
###

23. Конечные автоматы: задачи анализа и синтеза автоматов, автоматные функции и операции над ними (суперпозиция, введение обратной связи).

Синтез (построение по регулярному выражению детерминированного конечного автомата, его распознающего) --->
Анализ (построение регулярного выражения, которое распознает конечный автомат) --->
https://studfiles.net/preview/2674889/page:22/

Автоматные функции и операции над ними --->
https://studfiles.net/preview/4287769/page:15/
https://helpiks.org/5-24600.html(суперпозиция)
https://helpiks.org/5-24601.html (введение обратной связи)

##
###

24. Теорема Шеннона для канала с шумом.

Для любого действительного числа ε и для любого битрейта передачи R, меньшего чем пропускная способность канала C,
существует такая схема кодирования/декодирования сообщений,
при которой вероятность ошибки при передаче достаточно длинного сообщения меньше ε.
Если битрейт больше чем пропуская способность канала C, вероятность ошибки стремится к 1/2,
при длине сообщения, стремящегося к бесконечности.

Пропускная способность канала C может быть вычислена по формуле:

C = B log_2_(1 + S/N), где

B - ширина полосы пропускания канала;
S/N - отношение "сигнал/шум".

---> https://en.wikipedia.org/wiki/Noisy-channel_coding_theorem

##
###

25. Теорема Котельникова.

Если непрерывная функция x(t) не содержит частот, больших B Гц.,
то она полностью определяется своими значениями, взятыми в интервале 1/(2B) секунд.
Иными словами, для восстановления такой функции по дискретным отсчётам необходима частота дискретизации, не меньшая 2B Гц.

---> https://en.wikipedia.org/wiki/Nyquist%E2%80%93Shannon_sampling_theorem

##
###

26. Точные полиномиальные алгоритмы из теории расписаний, примеры NP-полных задач из теории расписаний.

---> http://www.mi-ras.ru/~scepin/1-sched.pdf

##
###

27. Приближенные полиномиальные алгоритмы для решения NP-трудных задач: задача о вершинном покрытии, задача об упаковке в контейнеры.

задача о вершинном покрытии --->
https://ru.wikipedia.org/wiki/Задача_о_вершинном_покрытии

https://habr.com/post/120328/

задача об упаковке в контейнеры --->
https://ru.wikipedia.org/wiki/Задача_об_упаковке_в_контейнеры

##
###

28. Понятие информации. Носители информации. Понятие сообщения. Формы сообщений. Передача сообщений. Способы измерения информации.

Информация – это:
- потребляемый всеми отраслями общества ресурс, имеющий для них такое же значение, как энергия или полезные ископаемые;
- совокупность научно-технических факторов, сведений, знаний о результатах развития науки и техники;
- знания, который человек получает из различных источников.

Знания бывают двух категорий:
- декларативные («я знаю, ЧТО…»);
- процедурные (определяющие процесс достижения некоторой цели «я знаю КАК…»).

Носитель информации – среда или физическое тело для передачи, хранения и воспроизведения информации. Основные характеристики носителей информации:
- информационная емкость;
- скорость обмена информацией;
- надежность;
- стоимость.

Информация передается в виде сообщений. Сообщение – это последовательность знаков или сигналов, которые содержат информацию. Одно и то же сообщение для разных людей – РАЗНАЯ информация. Отсюда любое сообщение имеет интерпретацию.
Формы сообщений бывают:
1) Устное сообщение - предоставление информации с помощью речи.
2) Письменное сообщение - представление информации в письменном виде.
3) Дискретное сообщение - сообщение, переданное с помощью дискретных сигналов.
4) Непрерывное сообщение - сообщение, которое можно задать непрерывной функцией.
3) Языковое сообщение - сообщение, переданное с помощью определенного языка.
Естественные - обмен информацией между людьми (язык жестов, писменный, профессиональный и т.д.)
Искуственнные - общение человека с компьютером, либо устройств между собой (радиосигналы, языки программирования и т.д.).

Информация передается в форме сообщений от некоторого источника информации к ее приемнику посредством канала связи между ними. Источник посылает передаваемое сообщение, которое кодируется в передаваемый сигнал. Этот сигнал посылается по каналу связи. В результате в приемнике появляется принимаемый сигнал, который декодируется и становится принимаемым сообщением.
Процесс: источник передачи -> канал связи -> приемник передачи. Все каналы связи делятся на:
- симплексные (передача информации только в одном направлении);
- полудуплексные (процесс передачи может идти в двух направлениях, но в какой-то конкретный момент только в одном);
- дуплексные (одновременно в двух направлениях).

Передача информации по каналам связи часто сопровождается воздействием помех, вызывающих искажение и потерю информации.

Для измерения количества информации существует два способа:
- вероятностный – результат зависит от содержания сообщения, а не от его объема (сообщение, которое уменьшает неопределенность знаний в 2 раза, несет 1 бит информации). Неопределенность – количество возможных результатов некоторого события;
- алфавитный способ – объем информации определен объемом сообщения. Алфавит – множество символов, которое используется для кодировки сообщения в некотором языке. Мощность алфавита – количество символов алфавита. Любой текст любым приемником будет восприниматься посимвольно, последовательно.

##
###

29. Понятие информационного процесса. Виды информационных процессов. Понятие информационных ресурсов, информационных систем. Эволюция информационных технологий. Классификация информационных систем.

Информационные процессы - процессы, связанные с поиском, хранением, передачей, обработкой и использованием информации.

Виды информационных процессов:
1) Хранение
2) Передача
3) Обработка

Информационные ресурсы – информация и инструменты управления этой информацией.
Информационная система - комплекс информационных ресурсов, технологии их получения и обработки, которые позволяют поддерживать информацию в актуальном и непротиворечивом состоянии.

Эволюция информационных технологий:
1) Первый этап (до конца 60-х годов)
Характеризуется проблемой обработки больших объемов данных в условиях ограниченных возможностей аппаратных средств. Характерные черты этого этапа:

программирование в машинных кодах, появление блок-схем, программирование в символьных процессах, разработка машинно-ориентированных языков и Ассемблера. Достижением в технологии программирования явилась разработка оптимизирующих трансляторов и появление первых управляющих программ реального времени и пакетного режима.
2) Второй этап (до конца 70-х годов)
Выпущены мини-ЭВМ и ЭВМ третьего поколения на больших интегральных схемах. Основным критерием создания информационных технологий стала экономия труда программиста. Цель - разработка инструментальных средств программирования. Появились операционные системы второго поколения, работающие в трех режимах: реального времени, разделения времени и в пакетном режиме. Разработаны языки высокого уровня, пакеты прикладных программ, системы управления базами данных, системы автоматизации проектирования, диалоговые средства общения с ЭВМ, новые технологии программирования (структурное и модульное), появились глобальные сети.
3) Третий этап (с начала 80-х годов)
Был сконструирован персональный компьютер. Информация становится ресурсом наравне с материалами, энергией, и капиталом. Появилась новая категория – информационные ресурсы. Изменился подход к созданию информационных систем - ориентация смещается в сторону индивидуального пользователя для поддержки принимаемых им решений.
4) Четвертый этап (90-е годы)
В этот период разрабатываются информационные технологии для автоматизации знаний. Цель – информатизация общества. Появились машины с параллельной обработкой данных; портативные ЭВМ, не уступающие по мощности большим; графические операционные системы; новые технологии: системы мультимедиа; гипертекст; объектно-ориентированные технологии. Телекоммуникации становятся средством общения между людьми. Созданы предпосылки формирования общего рынка знаний посредством дистанционного обучения, электронной памяти человечества по культуре, искусству, народонаселению, науке и т.д. Страны становятся зависимыми от источников информации, от уровня развития и эффективности использования средств передачи и переработки информации. Наступает этап информатизации общества.
5) Пятый этап (середина 90-ых - наше время)
Появление IP-протоколов для мобильных телефонов (VoIP и др.) распахнуло дверь для включения их в сеть интернет и развития электронного мобильного бизнеса. Критерий - доступ к информационным ресурсам каждому члену общества. Цель — глобализация общества. Появляются технологии проведения видеоконференций, управления знаниями и новациями, видеопочта, технологии для перепроектирования и модернизации устаревших систем. Происходит переход к автоматизации бизнес-процессов, происходящих в организациях. Информационные технологии проникают в приборы, устройства, во все сферы жизни человека.

Виды информационных систем:
1) Малые. Для них характерно:
- массовое использование;
- работа с небольшими объемами информации;
- небольшая цена;
- отсутствие средств модификации;
- использование настольных БД.
2) Средние. Для них характерно:
- возможность использования сети;
- обработка информации для нескольких рабочих мест;
- разделение функций между рабочими местами;
- присутствует штат обслуживающих сотрудников;

- включают в себя малые ИС.

3) Крупные. Для них характерно:

- использование большого разнообразия вычислительной техники и программного обеспечения;
- поддержка территориальной распределенности предприятия;
- включает в себя средства для аналитической обработки и поддержки принятия решений;
- включает в себя малые и средние ИС.

Классификация информационных систем:

1) По признаку структурированности задач
   - для структурированных задач
   - для частично структурированных или неструктурированных задач
2) По функциональному признаку и уровням управления
   - производственные системы
   - системы маркетинга
   - финансовые и учетные системы
   - системы кадров
   - прочие типы, выполняющие вспомогательные функции
3) По степени автоматизации
   - ручные
   - автоматические
   - автоматизированные
4) По характеру использования информации
   - информационно-поисковые системы
   - информационно-решающие системы
   - управляющие ИС
   - советующие ИС
5) По сфере применения
   - информационные системы организационного управления
   - ИС управления технологическими процессами
   - ИС автоматизированного проектирования
   - интегрированные (корпоративные) ИС

##
###

30. Стандартные требования при производстве ЭВМ. Стандартные методики измерения производительности ЭВМ. Альтернативные методики измерения производительности ЭВМ.

Стандартные требования:

1) Соотношение между стоимостью и производительностью компьютера
2) Соотношение между надежностью и отказоустойчивостью

Отказоустойчивость – это свойство вычислительных систем, при котором процесс выполнения программы не прекращается даже при нарушениях. Чем выше отказоустойчивость, тем дороже машина

3) Масштабируемость (аппаратная и программная)

Увеличение числа и объема ресурсов

4) Совместимость ПО

Необходимо обеспечить функционирование ПО на новой элементной базе и сохранить при этом интерфейс пользователя

Единицей измерения производительности компьютера является время: компьютер, выполняющий тот же объем работы за меньшее время является более быстрым. Время выполнения любой программы измеряется в секундах. Часто производительность измеряется как скорость появления некоторого числа событий в секунду, так что меньшее время подразумевает большую производительность.
Стандартные методики оценки:
1) Астрономическое время/время ответа/время выполнения/прошедшее время
Это задержка выполнения задания, включающая буквально все: работу процессора, обращения к диску, обращения к памяти, ввод/вывод и накладные расходы операционной системы. Однако при работе в мультипрограммном режиме во время ожидания ввода/вывода для одной программы, процессор может выполнять другую программу, и система не обязательно будет минимизировать время выполнения данной конкретной программы.
2) Время центрального процессора
- пользовательское время ЦП: время ответа, которое используется ЦП на выполнение данной программы (не включая все прочие затраты);
- системное время ЦП: время, необходимое процессору для выполнения функций ОС, связанных с данной программой.
3) Использование системы синхросигналов, вырабатываемых тактовым генератором (тактов)
Дискретные временные события называются тактами синхронизации или просто тактами. Разработчики компьютеров обычно говорят о периоде синхронизации, который определяется либо своей длительностью (например, 10 наносекунд), либо частотой (например, 100 МГц). Длительность периода синхронизации есть величина, обратная к частоте синхронизации.
Таким образом, время ЦП для некоторой программы может быть выражено двумя способами:
- количеством тактов синхронизации для данной программы, умноженным на длительность такта синхронизации;
- количеством тактов синхронизации для данной программы, деленным на частоту синхронизации.
4) Использование среднего количества тактов синхронизации на одну команду - CPI (clock cycles per instruction).

Альтернативные методики оценки:
а) MIPS (миллион операций в секунду). Количество операций за единицу времени.
Минусы: нельзя сравнивать машины с разными системами команд; характеристика не одна и та же для разных программ; уменьшение производительности при уменьшении количества программ.
б) MFLOPS (миллион операций с плавающей точкой в секунду). Используется для научных вычислений.
в) Наборы тестов:
- LINPACK. Ливерморские циклы: малый набор циклов – 14, большой набор – 24.
Цикл – часть программы, выполняющая определенные операции, например: программы для решения линейных алгебраических уравнений.
- SPEC. CINT92 – обработка целочисленных значений; CFP92 – обработка вещественных значений.
Используется отношение времени выполнения теста на машине, деленное на время выполнения на эталонной машине. В качестве эталонной машины используется

VAX11/780. Из всех отношений вычисляется среднее геометрическое, которое выдается за величину SPECINT (для целочисленных) и SPECFP (для вещественных). Минусы: эти тесты рассчитаны на выполнение в однопрограммном режиме. Не могут оценить реальную производительность. Для многопрограммного режима нужно знать пропускную способность. Запускается сразу несколько подобных тестов. Средние геометрические для группы многозадачного режима рассчитаны на научное применение.
- TPC. Оценка производительности систем бизнес класса (банковская сфера): TPC-A, TPC-B, TPC-C

##
###

31. Понятие типа данных. Концепция типа данных. Пример характеристики типа данных.

Каждая константа, переменная, выражение или функция бывают определённого типа. Этот тип существенным образом характеризует множество значений, к которому принадлежит константа, которые может принимать переменная или выражение или которые может вырабатывать функция.

Основные свойства типа данных:

1) Любой тип данных определяет множество значений, к которому принадлежит константа, которые может принимать переменная (или выражение),
или вырабатывать операция (или функция).
2) Тип значения, задаваемого константой, переменной или выражением, можно определить по их виду или описанию без необходимости выполнять какие-либо вычисления.
3) Каждая операция или функция требует аргументов фиксированного типа и выдает результат фиксированного типа.
Если операция допускает аргументы нескольких типов (например, '+' используется как для сложения целых, так и для сложения вещественных чисел), то тип результата можно определить по специальным правилам языка.
4) Каждый тип данных содержит множество допустимых операций, выполняемыми над значениями этого типа.
5) Новые типы данных можно строить на основе уже существующих. Значения, принадлежащие составному типу данных, как правило,
представляют собой совокупности значений компонент, принадлежащих к определённым ранее типам компонент. Такие составные типы данных называются
__структурированными__.
Если имеется только один тип компонент, т.е. все компоненты принадлежат одному типу, то он называется __базовым__.
6) Число различных значений, принадлежащих типу T, называется кардинальном числом T. Кардинальное число определяет размер памяти,
нужной для размещения переменной x типа T.

Поскольку типы компонент могут также быть составными, можно построить целую иерархию структур, но конечные компоненты структуры,
разумеется, должны быть атомарными. Следовательно, система нотаций должна допускать описание и простых, неструктуированных типов.
Самый простой метод описания простого типа - это перечисление значений этого типа.

Например, в программе, связанной с плоскими геометрическими фигурами, может описываться простой тип, называемый фигурой,
значения которого задаются идентификаторами `прямоугольник`, `квадрат`, `эллипс`, `круг`.
Тип данных, указываемый таким способом, называется __перечислением__.
Но кроме типов, задаваемых программистом, нужно иметь некоторые стандартные типы, которые называются предопределёнными.
Они обычно включают числа и логические переменные. Если значения некоторого типа упорядочены, то такой тип называется упорядоченным или __скалярным__.

Пример характеристики типа данных (с точки зрения языка C).

ТИП: unsigned long

МНОЖЕСТВО ЗНАЧЕНИЙ: [0, 2^64 - 1]

ДОПУСТИМЫЕ ОПЕРАЦИИ: '+',
'-',
'*',
'/' (целочисленное деление),
'%' (взятие остатка),
'++' (инкремент),
'--' (декремент),
'&' (побитное логическое `и`),
'|' (побитное логическое `или`),
'^' (побитное исключающее логическое `или`),
'~' (инверсия битов).

ЗАНИМАЕМАЯ ПАМЯТЬ: 8 байт

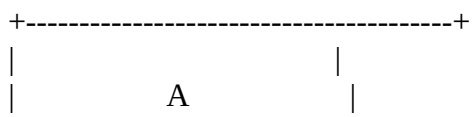
##
###

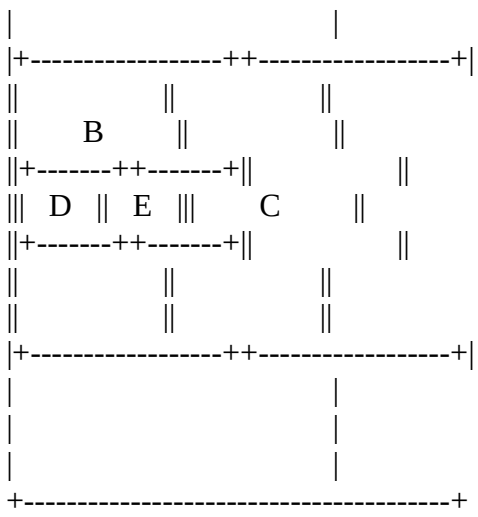
32. Понятие дерева. Способы изображения деревьев. Способы представления деревьев. Обход дерева. Основные характеристики сбалансированных деревьев: идеально-сбалансированное дерево, АВЛ-дерево, красно-черное дерево, дерево случайного поиска, B-дерево.

__Древовидная структура__ с базовым типом T - это либо:
1) пустая структура; либо
2) узел типа T, с которым связано конечное число древовидных структур с базовым типом T, называемых поддеревьями.

Способы изображения деревьев.

1) Вложенные множества.



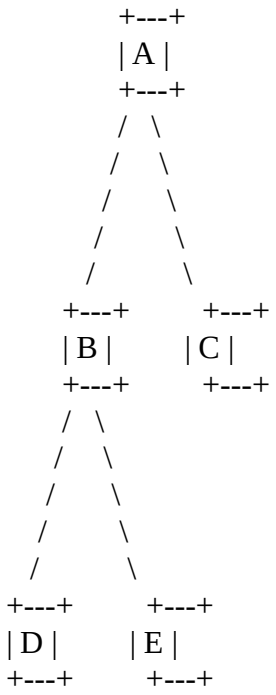


2) Вложенные скобки.

(A (B (D, E), C))

3) Ломанная последовательность.

4) Граф.



Способы представления деревьев.

1) Узел имеет фиксированный тип. Узлы соединяются при помощи указателей.
Отсутствие поддерева обозначается нулевым указателем: NULL;

```
struct node {
	void *data_ptr;
	struct node *left, *right;
} *root;
```

2) Представление дерева в виде массива структур. Способ, применимый в языках, где отсутствует поддержка динамической памяти.
Всё дерево размещается в специально выделенном массиве структур. Указатели на левое и правое поддеревья заменены на индексы в массиве.
Отсутствие поддерева обозначается специальным значением индекса, к примеру -1.

```
#define TREE_SIZE 5
struct node {
	void *data_ptr;
	int left_idx, right_idx;
} tree[TREE_SIZE];
```

Обход дерева.

1) Прямой обход (NLR -- Node - Left Subtree - Right Subtree). Обработать узел, посетить левое поддерево, посетить правое поддерево.

```
#define NULL ((void *) 0)

struct node {
	  void *data_ptr;
	  struct node *left, *right;
} *root;

void NLR(struct node *ptr, void (*handler) (struct node *))
{
	if (ptr != NULL) {
		handler(ptr);
		NLR(ptr->left, handler);
		NLR(ptr->right, handler);
	}
}
```

2) Центрированный (LNR -- Left Subtree - Node - Right Subtree). Посетить левое поддерево, обработать узел, посетить правое поддерево.

```
void LNR(struct node *ptr, void (*handler) (struct node *))
{
	if (ptr != NULL) {
		LNR(ptr->left, handler);
		handler(ptr);
		LNR(ptr->right, handler);
	}
}
```

3) Обратный (LRN -- Left Subtree - Right Subtree - Node). Посетить левое поддерево, посетить правое поддерево, обработать узел.

```
void LRN(struct node *ptr, void (*handler) (struct node *))
{
	if (ptr != NULL) {
		LRN(ptr->left, handler);
		LRN(ptr->right, handler);
		handler(ptr);
	}
}
```

Основные характеристики сбалансированных деревьев.

1) Идеально-сбалансированное дерево.
Дерево __идеально-сбалансировано__, если для каждого его узла количества узлов в левом и правом поддереве различаются не более чем на 1.

Правило построения идеально-сбалансированного бинарного дерева:

1) Взять один узел в качестве корня.
2) Построить левое поддерево с nl = n div 2 узлами тем же способом.
3) Построить правое поддерево с nr = n - nl - 1 узлами тем же способом.

2) АВЛ-дерево.

Сбалансированное дерево, по Адельсону-Вельскому и Ландису, - это дерево, для каждого узла которого высота его двух поддеревьев различается не более чем на 1.

Со сбалансированными деревьями можно выполнять следующие операции за O(log n) единицу времени даже в худшем случае:

* Найти узел с данным ключом.
* Включить узел с данным ключом.
* Удалить узел с данным ключом.

3) Красно-черное дерево.

Красно-черное дерево - бинарное дерево поиска с одним дополнительным битом цвета в каждом узле.
Цвет узла может быть либо красным, либо чёрным. В соответствии с накладываемыми на узлы дерева ограничениями
ни один простой путь от корня в красно-чёрном дереве не отличается от другого по длине более чем в два раза,
так что красно-черные деревья являются приближенно сбалансированными.

Каждый узел дерева содержит атрибуты color, key, left, right и p (parent). Если не существует дочернего или родительского узла
по отношению к данному, соответствующий указатель принимает значение NULL.
Данные указатели NULL рассматриваются как указатели на внешние узлы (листья) бинарного дерева поиска.

При этом все "нормальные" узлы, содержащие поле ключа, становятся внутренними узлами дерева.

Бинарное дерево поиска является красно-черным деревом, если оно удовлетворяет следующим свойствам:

* Каждый узел является либо красным, либо черным.
* Корень дерева является черным узлом.
* Каждый лист дерева (NULL) является черным узлом.
* Если узел красный, то оба его дочерних узла черные.
* Для каждого узла все простые пути от него до листьев, являющихся потомками данного узла,
содержат одно и то же количество черных узлов.

Красно-черное дерево с n внутренними узлами имеет высоту, не превышающую 2lg(n + 1), где lg - логарифм по основанию 2.

4) Дерево случайного поиска.

При вводе элемента в дерево случайного поиска этому элементу присваивается приоритет -- вещественное число с равномерным распределением в диапазоне [0, 1].
Приоритеты элементов в дереве случайного поиска определяют их положение в дереве в соответствии с правилом:
приоритет каждого элемента в дереве не должен быть более приоритета любого из его последователей.
Правило двоичного дерева поиска также остается справедливым: для каждого элемента X элементы в левом поддереве X будут меньше, чем в X, а в правом поддереве - больше, чем X.

5) B-дерево.

B-дерево представляют собой естественное обощение бинарных деревьев поиска.
Если внутренний узел X B-дерева содержит X.n ключей, то у него X.n + 1 дочерних узлов.
Ключи в узле X используются как разделители диапазона ключей, с которыми имеет дело данный узел, на X.n + 1 поддиапазонов,
каждый из которых относится к одному из дочерних узлов X. При поиске ключа в B-дереве мы выбираем один из X.n + 1 дочерних узлов путём сравнения искомого значения с X.n + 1 ключами, хранящимися в узле X.

Определение.

B-дерево T представляет собой корневое дерево (корень которого T.root), обладающее следующими свойствами:

1) Каждый узел X содержит следующие атрибуты:
* X.n -- количество ключей, хранящихся в настоящий момент в узле X.
* Собственно X.n ключей -- X.key_1, X.key_2, ..., X.key_n -- хранящихся в неубывающем порядке, так что X.key_1 <= X.key_2 <= ... <= X.key_n.
* Логическое значение X.leaf, равное TRUE, если X представляет собой лист, и FALSE, если X является внутренним узлом.

2) Кроме того, каждый внутренний узел содержит X.n + 1 указателей X.c_1, X.c_2, ..., X.c_(X.n + 1) на дочерние узлы.
У листьев дочерних узлов нет, так что значения их атрибутов c_i не определены.

3) Ключи X.key_i разделяют поддиапазоны ключей, хранящихся в поддеревьях:
Если k_i является произвольным ключом, хранящимся в поддереве с корнем X.c_i, то
k_1 <= X.key_1 <= k_2 <= X.key_2 <= ... <= X.key_(X.n) <= k_(X.n + 1)

4) Все листья расположены на одной и той же глубине, которая равна высоте дерева h.

5) Имеется нижняя и верхняя границы количества ключей, которые могут содержаться в узле.
Эти границы могут быть выражены с помощью одного фиксированного целого числа t >= 2, называемого минимальной степенью B-дерева.
* Каждый узел, кроме корневого, должен содержать как минимум t - 1 ключей. Каждый внутренний узел, не являющийся корневым,
имеет, таким образом, как минимум t дочерних узлов. Если дерево не является пустым, корень должен содержать как минимум один ключ.
* Каждый узел содержит не более 2t - 1 ключей. Таким образом, внутренний узел имеет не более 2t дочерних узлов.
Мы говорим, что узел заполнен, если он содержит ровно 2t - 1 ключей.

##
###

33. Понятие сортировки. Параметры оценки алгоритмов сортировки. Классификация сортировок. Характеристики внутренних методов сортировки. Дополнительные факторы, учитываемые при сортировке. Хеширование. Рехеширование.

Пусть дана конечная последовательность a_1, a_2, ..., a_n.
Под сортировкой подразумевают такую перестановку элементов этой последовательности
a_k_1, a_k_2, ..., a_k_n,
что при заданной функции упорядочения f справедливо отношение
f(a_k_1) <= f(a_k_2) <= ... <= f(a_k_n).
Обычно функция упорядочения не вычисляется по какому-то специальному правилу, а содержится в каждом элементе в виде явной компоненты (поля). Её значение называется ключом элемента.

Параметры оценки алгоритмов сортировки.

* Время сортировки – характеристика быстродействия алгоритма.
* Память – характеристика дополнительной памяти, требуемой алгоритмом сортировки. Дополнительная память - это любая память, требуемая помимо хранения исходного массива данных, а также текста программы.
* Устойчивость – алгоритм сортировки не меняет взаимного расположения элементов с равными ключами.
* Естественность поведения – параметр, которой указывает на эффективность метода при обработке уже отсортированных, или частично отсортированных данных. Алгоритм ведет

себя естественно, если учитывает эту характеристику входной последовательности и работает лучше.

Классификация сортировок.

* Внутренняя сортировка – это алгоритм сортировки, который в процессе упорядочивания данных использует только оперативную память (ОЗУ) компьютера.
То есть оперативной памяти достаточно для помещения в нее сортируемого массива данных с произвольным доступом к любой ячейке и собственно для выполнения алгоритма.
Внутренняя сортировка применяется во всех случаях, за исключением однопроходного считывания данных и однопроходной записи отсортированных данных.
В зависимости от конкретного алгоритма и его реализации данные могут сортироваться в той же области памяти, либо использовать дополнительную оперативную память.
* Внешняя сортировка – это алгоритм сортировки, который при проведении упорядочивания данных использует внешнюю память, как правило, жесткие диски.
Внешняя сортировка разработана для обработки больших списков данных, которые не помещаются в оперативную память.
Обращение к различным носителям накладывает некоторые дополнительные ограничения на данный алгоритм: доступ к носителю осуществляется последовательным образом, то есть в каждый момент времени можно считать или записать только элемент, следующий за текущим; объем данных не позволяет им разместиться в ОЗУ.

Характеристики внутренних методов сортировки.

Основное требование к методам сортировки массивов (внутренние методы) - экономное использование памяти.
Это означает, что переупорядочение элементов нужно выполнять на том же месте.
Методы сортировки, перемещающие данные в том же массиве, характеризуются следующими параметрами:
* C - число необходимых сравнений ключей.
* M - число пересылок элементов.
Эти числа определяются некоторыми функциями от числа n сортируемых элементов.
Хорошие алгоритмы сортировок требуют порядка n * log n сравнений.

Методы, сортирующие элементы массива на месте, можно разбить на три основных класса в зависимости от лежащего в их основе приёма:
* Сортировка включениями (вставками).
* Сортировка выбором.
* Сортировка обменом.

Дополнительные факторы, учитываемые при сортировке.

* Размер сортируемой последовательности (умещается ли вся последовательность в оперативную память?).
* Характеристики ключа (ключ - простой/составной, является ли ключ машинным словом и т. д.).
* Распределение ключей (имеется ли какое-то частичное упорядочение, имеются ли дубликаты ключей).
* Длина записи (записи большей длины целесообразно отделить от ключей; ключи ссылаются на данные при помощи указателей).

Хеширование. Рехеширование.

Хеширование - процесс вычисления по ключу K элемента X индекса элемента в массиве, равного i = h(K),
где h - некоторая хеш-функция.
Функция h отображает совокупность ключей U на ячейки хеш-таблицы T[0 .. m - 1]:

h: U -> {0, 1, ..., m - 1},

где размер m хеш-таблицы обычно гораздо меньше значения |U| - число возможных ключей.

При размещении следующих элементов (рехешировании) возможно появление коллизий - ситуаций,
когда различные элементы X_1, X_2, имеющие различные ключи K_1, K_2 соответственно, отображаются
в одну и ту же ячейку хеш-таблицы.

Методы разрешения коллизий:

1) Разрешение коллизий с помощью цепочек. Элементы с одинаковыми хешами вставляются в связный список,
ассоциированный с ячейкой хеш-таблицы.
2) Разрешение коллизий выбором ближайшей свободной ячейки. Элементы с одинаковыми хешами вставляются в различные ячейки хеш-таблицы.
В случае возникновения коллизии, с некоторым шагом анализируются последующие ячейки хеш-таблицы, выбирается первая свободная.

##
###

34. Понятие графа. Способы изображения графов. Способы представления графов. Обход графа. Алгоритм нахождения кратчайшего пути в графе. Алгоритм нахождения множества достижимых вершин в графе.

Граф - совокупность двух множеств V и E: G = (V, E) - где V - непустое множество вершин:

V = {V_1, V_2, ..., V_n}

E - множество рёбер:

E = {E_1, E_2, ..., E_k}

При этом, для неориентированного графа, E_i - неупорядоченная пара (V_i', V_i"), где V_i', V_i" - смежные вершины.
Граф изображается при помощи дуг и точек - получается геометрическая модель указанных выше двух множеств.

Способы представления графов.

1) Матрица смежности.
Матрица смежности - матрица, где столбцы и строки соответствуют вершинами графа.
На пересечении i-й строки и j-го столбца стоит
0 - отсутствует ребро (V_i, V_j);

1 - если V_i и V_j смежны.
Иными словами M[i, j] показывает, есть ли связи между двумя вершинами.

2) Матрица инцидентности.
Таблица, где строки соответствуют вершинам графа, а столбцы соответствуют рёбрам графа.
В ячейку матрицы на пересечении строки i со столбцом j записывается:
0 - j-е ребро не связано с i-й вершиной;
1 - j-е ребро связано с i-й вершиной.

3) Список смежности.
Список, где каждой вершине графа соответствует строка, в которой хранится список смежных вершин.
Такая структура данных не является таблицей в обычном понимании, а представляет собой многоуровневый список.

4) Список рёбер.
Список, где каждому ребру графа соответствует строка, в которой хранятся две вершины, являющиеся концами данного ребра.

Обход графа.

1) Обход в ширину.

Выбирается исходная вершина, обозначаемая далее s.
Все вершины графа, за исключением s, помечаются как непосещённые. Вершина s помечается посещённой.

Создается пустая очередь Q. В Q помещается вершина s.

В цикле, пока Q не пуста, выполняются следующие операции:
1) извлекается первая в очереди вершина u;
2) для каждой вершины v, смежной с u, проверяется, помечена ли она как посещённая;
3) если вершина v не была посещена, то она помечается посещённой и вставляется в очередь Q;
4) действия 2, 3 повторяются со следующей смежной с u вершиной;
5) если Q не пуста, то происходит возврат к пункту 1;
6) иначе алгоритм завершён.

2) Обход в глубину.

процедура посещение(G: неориентированный граф, u: вершина):
пометить u как посещённую;
для каждой вершины v, смежной с u и ещё не посещённой, рекурсивно вызвать процедуру посещение(G, v);
конец процедуры.

Все вершины графа G помечаются как непосещённые.
Выбирается исходная вершина, обозначаемая далее s.
Запускается процедура посещение(G, s).

Алгоритм нахождения кратчайшего пути в графе.

Алгоритм Дейкстры --->
https://neerc.ifmo.ru/wiki/index.php?title=%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%94%D0%B5%D0%B9%D0%BA%D1%81%D1%82%D1%80%D1%8B
https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%94%D0%B5%D0%B9%D0%BA%D1%81%D1%82%D1%80%D1%8B

Алгоритм нахождения множества достижимых вершин в графе.

Любой обход в графе из текущей вершины вернёт множество достижимых вершин.

##
###

35. Жизненный цикл программного обеспечения. Программы с большой и малой жизнью. Этапы разработки программ по ГОСТ ЕСПД, по Майерсу. Технологии макетирования. Модель водопада. Экстремальное программирование.

Жизненный цикл программного обеспечения:
1) Идея. Формулировка задачи, которую приносит заказчик.
2) Технико-экономическое обоснование
3) Техническое задание. Точная формулировка задачи, после прочтения которой должно быть понятно, что требуется, и не должно возникать вопросов по условиям.
4) Алгоритмы и структуры данных в машинно-НЕзависимой форме. Составить алгоритм решения поставленной задачи и решить, как будут представляться данные независимо от компьютера, среды и ЯП.
5) Алгоритмы и структуры данных в машинно-зависимой форме. Перевод алгоритма на некоторый ЯП.
6) Тестирование и отладка, поиск ошибок и их исправление.
7) Испытанная программа. Часто разработка останавливается на этом этапе.
8) Документирование программы
9) Программный продукт. Программа, которую любой программист может эксплуатировать, модифицировать и сопровождать.

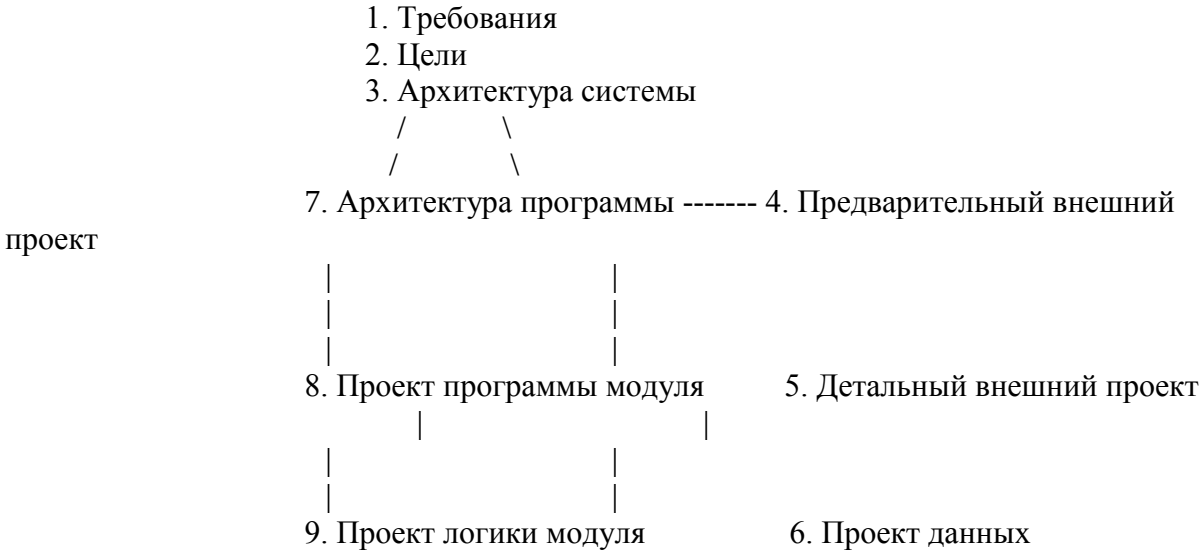
Требования к программному продукту:
1) Для всех данных, которые будут использоваться в программе, определить область доступных значений. Сообщить пользователю, если от него требуется ввод каких-либо данных.
2) Обобщить все алгоритмы, которые будут использоваться в программе. Алгоритмы, которые есть в программе, должны обрабатывать все возможные случаи.
3) Документация

Программы с большой и малой жизнью:
1) Программы с малой жизнью
Создаются небольшими коллективами (иногда одиночками). Часто предназначаются для научных целей или инженерных областей. Срок жизни 2-3 года.

Изначально не включают в себя средства тиражирования. Не содержат средств модификации в процессе эксплуатации.
2) Программы с большой жизнью
Создаются большими коллективами (от 100 человек). Применяются для регулярной обработки информации.
Включают в себя средства тиражирования. Наличие средств модификации в процессе эксплуатации.

Этапы разработки программ по ГОСТ ЕСПД:
ГОСТ ЕСПД (единая система программной документации) - документ, регламентирующий основные этапы разработки программы.
1) Техническое задание - документ, регламентирующий, каким образом будет разработана программа.
2) Техническое предложение - документ с различными вариантами решения поставленной задачи, делается оценка вариантов, выбирается лучший.
3) Эскизный проект - программа, отражающая принципиальные принятые решения по поводу задачи, выполняет лишь часть всех функций.
4) Технический проект - программа, отражающая окончательное решение задачи в полном объеме. Часто работа заканчивается уже на этом этапе.
5) Рабочий проект - программа, которая имеет документацию и пригодна для тиражирования.

Этапы разработки программ по Майерсу:



1) Требования - предварительная формулировка задачи заказчика.
2) Цели - окончательная формулировка задачи заказчика.
3) Архитектура системы - разложение задачи на подзадачи и определение связей между ними.
4) Предварительный внешний проект - описание взаимодействия программы и пользователя.
5) Детальный внешний проект - добавления описания всех сообщений и команд.
6) Проект данных - описание всех видов и наборов даных, которые будут использоваться для работы данной программы.
7) Архитектура программы - разложение подзадачи на выполняемые функции и определение связей между ними.
8) Проект программы модуля - алгоритм решения в машинно-НЕзависимой форме.

9) Проект логики модуля - алгоритм решения в машинно-зависимой форме.

Технологии макетирования:
Программа создается как последовательность прототипов.
Прототип - программа, которая решает все типичные задачи заказчика и которую можно быстро разработать.
1) Демонстрационный прототип - программа, которая демонстрирует заказчику способ решения его задач и жизнеспособность методов, выбранных для решения этих задач.
2) Исследующий прототип - программа, которая решает все задачи, но неустойчива в работе.
3) Действующий прототип - программа, которая надежно решает все задачи, но для решения некоторых требует слишком много ресурсов.
4) Промышленная система - программа, которая надежно решает все задачи и при этом требует минимума ресурсов. Часто работа заканчивается на этом этапе.
5) Коммерческая система - программа, пригодная для тиражирования.

Модель водопада:
В модели водопада каждая из процессных областей представляет собой отдельную фазу проекта. Фазы выполняются строго последовательно, т.е. анализ и дизайн начинаются после завершения разработки требований, началу реализации предшествует завершение дизайна и т.д.
1) Разработка требований - сбор бизнес-требований заказчика и их преобразование в функциональные требования к программному продукту.
2) Анализ и дизайн - разработка модели предметной области, проектирование схемы базы данных, объектной модели, пользовательского интерфейса и т.п.
3) Реализация - создание продукта по спецификациям, разработанным на предыдущем этапе.
4) Тестирование - включает проверку соответствия функциональности программного продукта потребностям пользователей, а также поиск дефектов в реализации.
5) Развертывание - обучение пользователей, инсталляция системы, перевод в промышленную эксплуатацию.

Экстремальное программирование:
Применяется для ускорения разработки программ.
1) Формулировка требований
2) Постановка целей
3) "Игра с заказчиком". Заказчик выбирает из перечисленных функций наиболее важные, после чего они реализуются
4) Тестирование
5) Если реализованы не все функции, перейти к шагу 3

##
###

36. Принятие решений при разработке программ. Формальное обоснование принятых решений. Вариантный сектор, вариантная сеть

Вариантный сектор - методика фиксации и обоснования проектных решений.
Есть некоторый вопрос (цель), для которого существует несколько возможных решений.
Первое, что нужно сделать - перечислить все возможные варианты.

Далее необходимо перечислить свойства вариантов. Они могут быть как положительными, так и отрицательными, но формулировка должна быть четкой, чтобы было понятно положительное это свойство или отрицательное, т.е. должно быть понятно, к чему мы стремимся. Например, свойство "уровень зарплаты" сформулировано неправильно. Правильная формулировка: "Высокий уровень зарплаты".
Третий шаг - составление таблицы.
В первый столбец записываются варианты решений. В первой строке перечисляются критерии, которые будут использоваться в качестве критериев оценки. Далее заполняется матрица. Указывается, до какой степени проявляется каждое свойство у каждой альтернативы по 10-балльной шкале. В вектор записывается степерь важности (желаемости) каждого критерия от -10 до 10. Далее матрица умножается на вектор, в результате чего получается рейтинг альтернатив.
Порядок заполнения: сначала проявление свойств, затем заполнение степеней желаемости, после этого умножение матрицы на вектор.

	критерий 1	критерий 2	критерий 3	рейтинг
важность	2	8	6	
решение 1	3	4	1	44
решение 2	5	7	6	102
решение 3	1	2	3	36

Вариантный сектор — одна проблема, какой-то один вопрос. Реальных проблем будет большое количество и нужно их некоторым образом организовать. Совокупность упорядоченных вариантных секторов – вариантный каркас проекта.
Проект разрабатывается в некоторых условиях, ограничениях. Кроме ограничений в условия могут попасть цели проекта. Для того чтобы всё это объединить, вводится понятие «вариантный каркас проекта».
1) Заголовок проекта
2) Условия (цели, подцели, ограничения)
3) Темы, подтемы — иерархия тем. В каждой подтеме на нижнем уровне — вариантный сектор. Для нумерации тем используется буквы, для нумерации секторов — цифры.

Периодически возникают несколько проектов, близких друг к другу. Поэтому хотелось бы сохранить наработки по прежнему проекту и добавлять туда новые материалы. Часть вопросов возникает в зависимости от принятых ранее вариантов. Если попытаться объединить вместе сектора из нескольких проектов, возникает необходимость отметить, какой сектор для какого проекта. Появляется конструкция, которую именуют вариантной сетью.

##
###

37. Порядок сборки программы. Методы тестирования программ. Методы отладки программ.

1) Препроцессирование - преобразование исходного кода программы для дальнейшего компилирования (макроподстановки, обработка директив препроцессора: вставка файлов, условное компилирование)
2) Компиляция - преобразование кода, полученного на прошлом этапе, в ассемблерный код. Ассемблерный код — это мнемоническая запись машинного кода.
3) Ассемблирование - преобразование ассемблерного кода в машинный код, сохраняя его в объектом файле. Объектный файл — это файл, содержащий машинный код с неразрешёнными внешними зависимостями.
4) Компоновка - разрешение внешних зависимостей, достигаемое как слитием воедино отдельных объектных файлов, так и связыванием со статическими библиотеками. Существует двух типов: статическая компоновка (получение исполняемого файла, готового к выполнению на CPU; в таком файле полностью разрешены все внешние зависимости),
динамическая компоновка (разрешение зависимостей при запуске программы).

Тестирование - процесс исполнения программы с целью обнаружения ошибок.
Методы тестирования программ:
1) Тестирование с точки зрения "черного ящика" - выяснение обстоятельств, в которых поведение программы не соответствует её спецификации;
внутренняя структура программы при этом не учитывается.
2) Тестирование с точки зрения "белого ящика" - получение тестовых данных путем анализа логики программы.
3) Ручное тестирование - чтение и визуальная проверка программы группой лиц (3-4 человека, один из них автор, обмен мнениями в конце, цель - нахождение ошибок).
4) Пошаговое тестирование - подход к тестированию, при котором каждый модуль (подпрограмма) для тестирования подключается к набору уже протестированных модулей - модули тестируются НЕизолированно друг от друга.
5) Нисходящее тестирование - разновидность пошагового тестирования, при которой тестирование начинается с верхнего, головного модуля программы;
каждый новый модуль должен вызываться одним из уже протестированных модулей.
6) Восходящее тестирование - разновидность пошагового тестирования, при которой тестирование начинается с терминальных модулей (модулей, не вызывающих другие);
каждый новый модуль для тестирования должен вызывать один из уже протестированных модулей.

Отладка - процесс, осуществляемый после удачного теста в два этапа:
- определение природы и местонахождения ошибки в программе;
- исправление ошибки.

Методы отладки программ:
1) Метод "грубой силы" - метод, при котором ошибка выявляется при помощи средств, позволяющих отследить состояние программы в большом числе точек (например, с использованием операторов вывода).
2) Метод индукции - метод, позволяющий выявить ошибку на основе анализа данных, при которых она была обнаружена.
3) Метод дедукции - метод, позволяющий на основании некоторых общих теорий или предпосылок, используя операции исключения и уточнения, обнаружить местонахождение ошибки.
4) Прослеживание логики в обратном порядке - отладка начинается в точке, где был зафиксирован некорректный результат (в операторе вывода) и идет в обратном порядке до тех пор, пока не выявлена ошибка; состояние при каждом операторе вычисляется на основании предшествующего состояния.

5) Метод тестирования - метод, когда ошибку локализуют за счет тестов, подобранных определенным образом.

##
###

38. Парадигмы языков программирования, разные подходы. Критерии оценки языков программирования. Представление основных объектов данных в императивных языках. Механизмы типизации.

1) Синтезирующее программирование - ручное, автоматическое или автоматизированное манипулирование данными о задаче с целью получения алгоритма её решения.
* Императивной программирование (Pascal, C). Определить всё, что требуется, создать алгоритм, выразить его средствами языка программирования.
* Функциональное программирование (Lisp). Программа, представленная в виде суперпозиции функций.
* Логическое программирование (Prolog). Программа как набор фактов и правил.
* Параллельное программирование. Написание программ, которые исполняются на нескольких CPU.

2) Сборочное программирование - построение программы из уже существующих и корректных фрагментов.
* Модульное программирование. Любая программа - совокупность модулей.
В простейшем случае программа - один единственный модуль.
* Компонентное программирование. В основе лежит бинарный объект.
Бинарный объект имеет чёткий интерфейс, доступный из нескольких языков программирования.
Этот объект можно использовать из различных языков программирования,
при условии, что у них реализация соответствующего интерфейса к объекту.

3) Конкретизирующее программирование - создание программ из специальных универсальных заготовок.
* Объектно-ориентированное программирование. Любая программа представляет собой совокупность взаимодействующих объектов.
Каждый объект характеризуется данными, а также методами. И данные, и методы могут быть скрыты от внешней сущности (инкапсуляция).
Объект - экземпляр класса. Класс - описание нового типа данных.
* Шаблонно-ориентированное программирование (Clarion, элементы встречаются в C).
Возможность использования шаблонов для взаимодействия с пользователем.

Критерии оценки языков программирования:

0) Назначение!
Различные языки программирования имеют различное назначение:
Java, к примеру, никак не годится на написание ядра операционной системы.

1) Целевая платформа.
Пример: сравнение Java и C.
Если программа написана на C и должна работать на машинах с Windows® и Linux®,
потребуются компиляторы для платформ и два разных исполняемых файла.

В случае с Java сгенерированного байт-кода будет достаточно для выполнения программы на любом компьютере,
на котором установлена виртуальная Java-машина.

2) Гибкость языка.
Гибкость языка определяется тем, насколько легко можно добавлять к существующей программе новые функциональные возможности.
Это может быть добавление нового набора функций или использование существующей библиотеки для добавления новой функциональности.

3) Время исполнения проекта.
Время исполнения – это время, необходимое для создания рабочей версии программы, т.е. версии,
готовой для работы в производственных условиях и выполняющей предусмотренные функции.

4) Производительность.
Каждая программа и платформа имеет определенный предел производительности,
и на эту производительность влияет используемый при разработке язык.

5) Поддержка и сообщество.
Язык программирования, как и хорошая программа, должен опираться на твердую поддержку сообщества.
Язык с активным форумом скорее всего будет популярнее замечательного языка, помощь по которому трудно найти.

Представление основных объектов данных в императивных языках (на примере C).

Скалярные типы данных - знаковые и беззнаковые целые числа различной разрядности, перечисления, указатели.

1 байт - signed/unsigned char;
2 байта - signed/unsigned short;
4 байта - signed/unsigned int;
4/8 байт (в зависимости от архитектуры CPU) - signed/unsigned long;

```
enum _type {
	first = 1;
	second;
	<...>
	final;
};
```

enum определяет, по сути, именованные константы типа int, обрабатываемые на этапе компиляции.

void *ptr -- нетипизированный указатель; может быть присвоен любому типизированному указателю.
<базовый тип> *ptr -- типизированный указатель.
Указатель всегда занимает в памяти размер машинного слова (32/64 бита для 32/64-х разрядных машин),
поскольку, по сути, является адресом ячейки памяти.

Структуированные типы - структуры, объединения, битовые поля, массив.

Структура:

```
struct _type {
	<поля структуры>
};
```

Структура - набор полей, имеющих общее смысловое значение (каждое поле является атрибутом некоторого объекта),
структура в памяти хранится в виде последовательного размещения её полей, при этом допустимы отступы между полями,
чтобы каждое поле было выровнено по естественной границе.

Битовое поле - частный случай структуры, в котором одно или несколько полей представляют отдельные биты (а не байты).
Описывается в виде <базовый целочисленный тип> name:<число бит>;

Объединение - набор полей, характеризующих одну и ту же область памяти.
Объединение в памяти занимает столько байт, сколько занимает самый крупный её член.
Позволяет по-разному работать с одной и той же памятью.

```
union _type {
	<поля объединения>
};
```

Массив - набор однотипных данных, хранящихся в памяти последовательно.

<базовый тип> _array[<число элементов>];

Типизация бывает двух видов:

1) Статическая (C, Pascal) - тип имеет сама переменная, характерно для компилируемых языков.
Соответствие типов строго проверяется на этапе компиляции.
2) Динамическая (Bash, Python) - тип имеет значение, но не переменная.
Любая переменная может принимать значения произвольных типов.

##
###

39. Структурное программирование. Основные структуры управления. Теорема структурирования. Преобразование Ашкрофта-Манны.

Структурное программирование - парадигма программирования, стремящаяся достичь ясность и качество кода,
а также уменьшить время разработки программ путём использования структурных элементов выбора (if/then/else),
повторения (while, for), последовательного выполнения, а также процедур.

Основные структуры управления:

1) Последовательное выполнение. Операторы выполняются последовательно;
новый оператор не выполняется до тех пор, пока не выполнен предыдущий.
2) Условие. Проверяется некоторое условие. Если оно истинно, выполнение идёт по ветке "то".
В противном случае выполнение продолжается по ветке "иначе".
3) Цикл. Тело цикла повторяется до тех пор, пока условие истинно.

Теорема структурирования гласит, что любой алгоритм может быть представлен с использованием
перечисленных выше структур управления (без оператора goto).

Преобразование Ашкрофта-Манны.

В основе метода лежит введение переменной-состояния.
Каждый блок неструктуированной программы помечается некоторым целым числом.
Начальное значение переменной-состояния - номер начального блока в исходном алгоритме.
Все блоки исходного алгоритма помещаются в цикл с условием.
Выход из цикла происходит по достижении переменной-состоянием значения последнего блока.
Вместо передачи управления происходит присваивание переменной-состоянию значения соответствующего блока.
Таким образом, оператор goto преобразуется в итерации цикла, а также присваивания единственной целочисленной переменной.

##
###

40. Понятие формальных языков и грамматик. Иерархия по Хомскому.

__Формальный язык__ -- множество цепочек конечной длины в некотором алфавите A.

__Грамматика__ -- четверка G = (N, A, P, S), где
1. N -- конечное множество нетерминальных символов, или нетерминалов
(иногда называемых вспомогательными символами, синтаксическими переменными или понятиями);
2. A -- не пересекающееся с N конечное множество терминальных символов, или терминалов
(проще говоря, алфавит);
3. P -- конечное подмножество множества
(N ∪ A)* N (N ∪ A)* x (N ∪ A)*
Элемент (a, b) множества P называется правилом (или продукцией) и записывается в виде
a -> b;
4. S -- выделенный символ из N, называемый начальным (или исходным) символом.

Примером грамматики служит четверка G1 = ({X, S}, {0, 1}, P, S), где P состоит из правил (e -- "пустой" символ):
S -> 0X1

0X -> 00X1
X -> e

Нетерминальными символами являются X и S, а терминальными - 0 и 1.

Грамматика определяет язык рекурсивным образом. Рекурсивность проявляется в задании особого рода цепочек,
называемых выводимыми цепочками грамматики G = (N, A, P, S):

1. S - выводимая цепочка.
2. Если abc - выводимая цепочка и b -> d содержится в P, то adc - тоже выводимая цепочка.

Выводимая цепочка грамматики G, не содержащая нетерминальных символов, называется терминальной цепочкой, порождённой грамматикой G.

Язык, порождаемый грамматикой G, - это множество терминальных цепочек, порождаемых грамматикой G.

Иерархия по Хомскому.

Грамматики можно классифицировать по виду их правил.

Грамматика G:

1) праволинейная, если каждое правило из P имеет вид
X -> tY или X -> t, где X, Y ∈ N, t ∈ A*;
2) контекстно-свободной, если каждое правило из P имеет вид
X -> α, где X ∈ N, α ∈ (N ∪ A)*
3) контекстно-зависимой (или неукорачивающей), если каждое правило из P имеет вид
α -> β, где |α| <= |β|
4) грамматикой общего вида (без ограничений) -- во всех остальных случаях.

##
###

41. Автоматные грамматики. Конечные автоматы. Теорема Клини. Понятие регулярного выражения. Эквивалентность регулярных выражений и автоматных грамматик.

Автоматная (праволинейная) грамматика -- грамматика G = (N, A, P, S), правила вывода которой имеют вид:
X -> tY или X -> t, где X, Y ∈ N, t ∈ A*;

Недетерминированный конечный автомат -- это пятёрка M = (Q, A, δ, q0, F), где

1. Q -- конечное множество состояний;
2. A -- конечное множество допустимых входных символов (алфавит);
3. δ -- отображение множества Q x A в множество Γ(Q), определяющее поведение управляющего устройства;

Γ(Q) - означает, что для каждой пары (состояние, входной символ) существует несколько состояний,
в которые может перейти автомат; функцию δ иногда называют функцией переходов;

4. q0 ∈ Q -- начальное состояние управляющего устройства;

5. F ⊆ Q -- множество заключительных состояний.

Детерминированный конечный автомат -- автомат, у которого, для каждого состояния q и для каждого входного символа a,
множество значений функции δ(q, a) содержит не более одного состояния.

Теорема Клини гласит, что
класс языков, определяемых недетерминированными конечными автоматами,
совпадает с классом языков, определяемых детерминированными конечными автоматами;
иными словами, для любого языка, распознаваемого недетерминированным конечным автоматом,
можно построить детерминированный конечный автомат, распрознающий данный язык.

Регулярные выражения в алфавите A и регулярные множества, которые они обозначают, определяются рекурсивно следующим образом:
1. ∅ -- регулярное выражение, обозначающее множество ∅;
2. e -- регулярное выражение, обозначающее множество {e};
данное регулярное выражение обозначает пустую строку;
3. если x ∈ A, то x -- регулярное выражение, обозначающее регулярное множество {x};
4. если p, q - регулярные выражения, обозначающее регулярные множества P и Q соответственно, то
(p + q) -- регулярное выражение, обозначающее P ∪ Q;
(pq) -- регулярное выражение, обозначающее P x Q (декартово произведение множеств);
(p)* -- регулярное выражение, обозначающее P*
(степени множества P, основанные на декартовом произведении);
5. ничто другое не является регулярным выражением.

Эквивалентность регулярных выражений и автоматных грамматик.

Утверждения
1. L - регулярное множество,
2. L - язык, порождённый праволинейной грамматикой,
3. L - язык, порождённый конечным детерминированным автоматом,
4. L - язык, порождённый конечным недетерминированным автоматом,
5. L обозначается регулярным выражением
эквивалетны.

##
###

42. Контекстно-свободные грамматики. Учет самовложения в алгоритмах распознавания. Метод рекурсивного спуска при анализе грамматики. LL-грамматики. Синтаксические диаграммы для описания КС-грамматик.

Контекстно-свободные грамматики -- грамматики, правила вывода которых имеют вид:
X -> α, где X ∈ N, α ∈ (N ∪ A)*
Иными словами, на вид правых частей правил не накладывается никаких ограничений, а левая часть каждого правила - единственный нетерминал.
С помощью контекстно-свободных грамматик задают синтаксис языков программирования.

Самовложение в КС-грамматиках.
Если в грамматике G есть нетерминал X, для которого X -> αXβ,
то есть из X нетривиально выводится цепочка αXβ, где α, β - непустые цепочки терминалов и нетерминалов,
то говорят, что такая грамматика содержит самовложение.

Метод рекурсивного спуска при анализе грамматики.

Рекурсивный спуск — это эффективный и простой нисходящий алгоритм распознавания. Его суть в следующем. Для каждого нетерминала грамматики (понятия, конструкции языка) записывается отдельная распознающая процедура.
При этом соблюдаются следующие соглашения:
1. Перед началом работы процедуры текущим является первый символ анализируемого понятия.
2. В процессе работы процедура считывает все символы входной цепочки,
относящиеся к данному нетерминалу (выводимые из данного нетерминала) или сообщает об ошибке.
Если правила для данного нетерминала содержат в правых частях другие нетерминалы, то процедура обращается к распознающим процедурам этих нетерминалов для анализа соответствующих частей входной цепочки.
3. По окончании работы процедуры текущим становится первый символ, следующий во входной цепочке за данной конструкцией языка
(символами, выводимыми из данного нетерминала).

LL-грамматики.

LL(k)-грамматикой называется КС-грамматика, в которой выбор правила в ходе левостороннего вывода
однозначно определяется не более чем k очередными символами входной цепочки, считываемой слева направо.

Синтаксические диаграммы для описания КС-грамматик - это направленный граф с одним входным ребром и одним выходным ребром
и помеченными вершинами. Цепочка пометок при вершинах на любом пути от входного ребра к выходному - это цепочка языка,
задаваемого синтаксической диаграммой.

##
###

43. Структура компилятора. Основные функции лексического, синтаксического и контекстного анализаторов. Таблицы компиляции. Этапы генерации кода. Понятие о виртуальных машинах. Самокомпиляция и раскрутка.

Структура компилятора.

Исходная программа, написанная на некотором языке программирования, есть не что иное, как цепочка знаков.

Компилятор в конечном итоге превращает эту цепочку знаков в цепочку битов - объектный код.
В этом процессе часто можно выделить подпроцессы со следующими названиями:
1. Лексический анализ.
2. Работа с таблицами.
3. Синтаксический анализ, или разбор.
4. Контекстный анализ.
4. Генерация кода, или трансляция в промежуточный код (например, язык ассемблера).
5. Оптимизация кода.
6. Генерация объектного кода (например, ассемблирование).

Функции лексического анализатора.

Работа лексического анализатора состоит в том, чтобы сгруппировать определённые терминальные символы
в единые синтаксические объекты, называемые лексемами.
Какие объекты считаются лексемами зависит от определения языка программирования.

Лексема - это цепочка терминальных символов, с который связывается лексическая структура,
состоящая из пары вида (тип лексемы, некоторые данные). Первой компонентной пары является синтаксическая категория,
такая как "константа" или "идентификатор", а вторая - указатель: в ней указывается адрес ячейки, хранящей информацию об этой конкретной лексеме.

Таким образом, лексический анализатор - это транслятор, входом которого служит цепочка символов,
представляющая исходную программу, а выходом - последовательность лексем.
Этот выход образует вход синтаксического анализатора.

Функции синтаксического анализатора.

Синтаксический анализ, или разбор, - это процесс, в котором исследуется цепочка лексем и устанавливается,
удовлетворяет ли она структурным условиям, явно сформулированным в определении синтаксиса языка.

Выходом анализатора служит дерево, которое представляет синтаксическую структуру, присущую исходной программе.

Функции контекстного анализатора.

Задачей контекстного анализа является установление свойств объектов и их использования.
Наиболее часто решаемой задачей является определение существования объекта и соответствия его использования контексту, что осуществляется с помощью анализа типа объекта.
Под контекстом здесь понимается вся совокупность свойств текущей точки программы, например множество доступных объектов, тип выражения и т. д.

Таблицы компиляции.

* Таблица имён - таблица, в которой хранится информация об идентификаторах программы (именах функций, переменных, их областей видимости и т. п.).
* Таблица управления - таблица, в которой хранится информация о передаче управления в программе (вызовы функций, циклы, условия).

Этапы генерации кода.

Задача генератора кода - построение для программы на исходном языке эквивалетной машинной программы.
Обычно, в качестве входа для генератора служит некоторое промежуточное представление программы.

Генерация кода включает ряд специфических, относительно независимых подзадач:
* распределение памяти (в частности, распределение регистров);
* выбор машинных инструкций;
* генерацию объектного (или загрузочного) модуля.

Понятие о виртуальных машинах.

Виртуальная машина - система, эмулирующая аппаратное обеспечение некоторой платформы
и исполняющая программы для целевой платформы на хосте.
Виртуальная машина исполняет некоторый машинно-независимый код (например, байт-код)
или машинный код реального процессора. Пример виртуальной машины: виртуальная машина Java.

Самокомпиляция и раскрутка.

Раскрутка компилятора (англ. bootstrapping — от boot и strap) — метод создания транслятора для некоторого языка программирования,
при котором транслятор пишется на том же языке программирования, для трансляции которого создаётся;
создание транслятором исполняемых файлов из исходного кода самого транслятора.
Используется для переноса трансляторов на новые платформы.

##
###

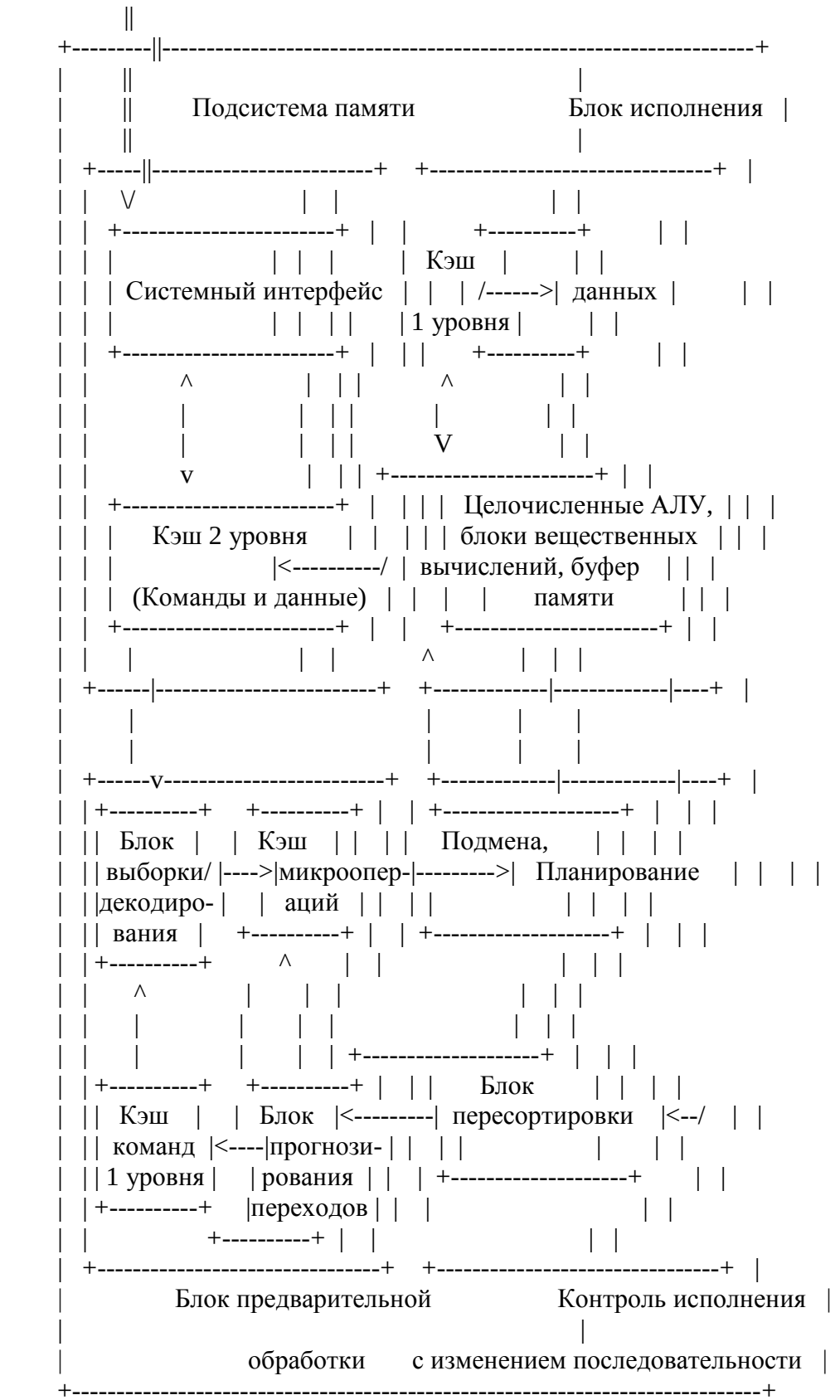
44. Процессоры компании Intel. Архитектура процессоров IA-32. Микроархитектура процессоров Intel.

==========
==========

На примере Sandy Bridge (core i7).

Микроархитектура Sandy Bridge.

/\ К общему кэшу 3-го уровня (Symmetric Multi-Processing, SMP)



Core i7 состоит из четырех основных блоков:

* подсистемы памяти;
* блока предварительной обработки;
* блока контроля исполнения с изменением последовательности;
* блока исполнения.

Подсистема памяти.

Каждый процессор Core i7 содержит подсистему памяти с объединённым кэшем второго уровня (L2), а также логикой доступа к кэшу 3 уровня (L3). Все процессоры совместно используют общий кэш 3 уровня (SMP) - это "последняя остановка", после которой обращение выходит за пределы микросхемы центрального процессора и отправляется по шине к внешней памяти.
Объем кэшей L2 в Core i7 составляет 256 Кбайт; они представляют собой 8-входовую (8-way) ассоциативную кэш-памяти с 64-байтовыми строками (cacheline). Размер общего кэша L3 лежит в диапазоне от 1 до 20 Мбайт. Независимо от размера кэш L3 представляет собой 12-входовый ассоциативный кэш с 64-байтовыми строками. Если запрос к кэшу третьего уровня не приносит результата, он передается в оперативную памяти по шине DDR.

С кэшем 1 уровня связаны два блока предварительной выборки, не показанные на рисунке выше.
Эти блоки пытаются перенести данные из основной памяти в L1 еще до того, как эти данные были запрошены. Один блок осуществляет предварительную выборку следующего блока памяти при обнаружении последовательного "потока" памяти, передаваемого процессору. Второй, более сложный блок предварительной выборки отслеживает последовательность адресов операций чтения/записи конкретной программы. Если операции осуществляются с постоянным шагом, блок заранее выбирает следующий элемент, к которому, скорее всего, обратится программа.

Блок предварительной обработки.

Подсистема памяти связана как с блоком предварительной обработки, так и с кэшем данных L1.
Блок предварительной обработки отвечает за выборку команд из подсистемы памяти, декодирование их в микрооперации по типу RISC и сохранение в двух кэшах команд. Все команды после выборки помещаются в кэш команд L1. Размер кэша L1 составляет 32 Кбайт, он представляет собой 8-входовую ассоциативную кэш-память с 64-байтовыми строками (cacheline). В ходе выборки из кэша L1 команды попадают в декодеры, определяющие последовательность микроопераций, используемых для реализации команды в конвейере исполнения.

Механизм декодирования связывает устаревший набор команд CISC и современные RISC-команды.

Декодированные микрооперации передаются в кэш микроопераций, называемые кэшем команд L0.
Кэш микроопераций напоминает традиционный кэш команд, но в нем достаточно места для
хранения последовательностей микрокоманд, генерируемых отдельными командами. Поскольку кэшируются не исходные команды, а декодированные микрооперации, необходимость
в повторном декодировании при последующих исполнениях команды отпадает.

Прогнозирование переходов также выполняется в блоке предварительной обработки. Блок прогнозирование должен "угадать", когда ход выполнения программы отклонится от строго последовательной выборки, причем он должен сделать это задолго до исполнения
команд перехода. Блок прогнозирования переходов отслеживает результаты предыдущих переходов и использует эту информацию для новых прогнозов. Детали реализации
блока прогнозирования переходов держатся в секрете.

Планировщик (контроль исполнения с изменением последовательности).

Команды передаются из кэша микроопераций планировщику команд в порядке, определяемом
программой, но при их исполнении возможно отсутпление от этого порядка. Обнаружив микрооперацию, которую нельзя исполнить, планировщик удерживает её, одновременно продолжая
обрабатывать поток команд - запускаются все последующие команды, которые не требуют обращения
к занятым ресурсам (регистрам, функциональным блокам и т. д.). Здесь же выполняется подмена
регистров, благодаря чему WAR- и WAW- взаимозависимые команды могут исполняться без задержки.

ПРИМЕЧАНИЕ: https://en.wikipedia.org/wiki/Hazard_(computer_architecture)

Write after write (WAW) - инструкция, следующая после текущей, пытается записать операнд,
прежде чем он записан текущей инструкцией.

Write after read (WAR) - инструкция, следующая после текущей, пытается записать операнд,
прежде чем он считан текущей инструкцией.

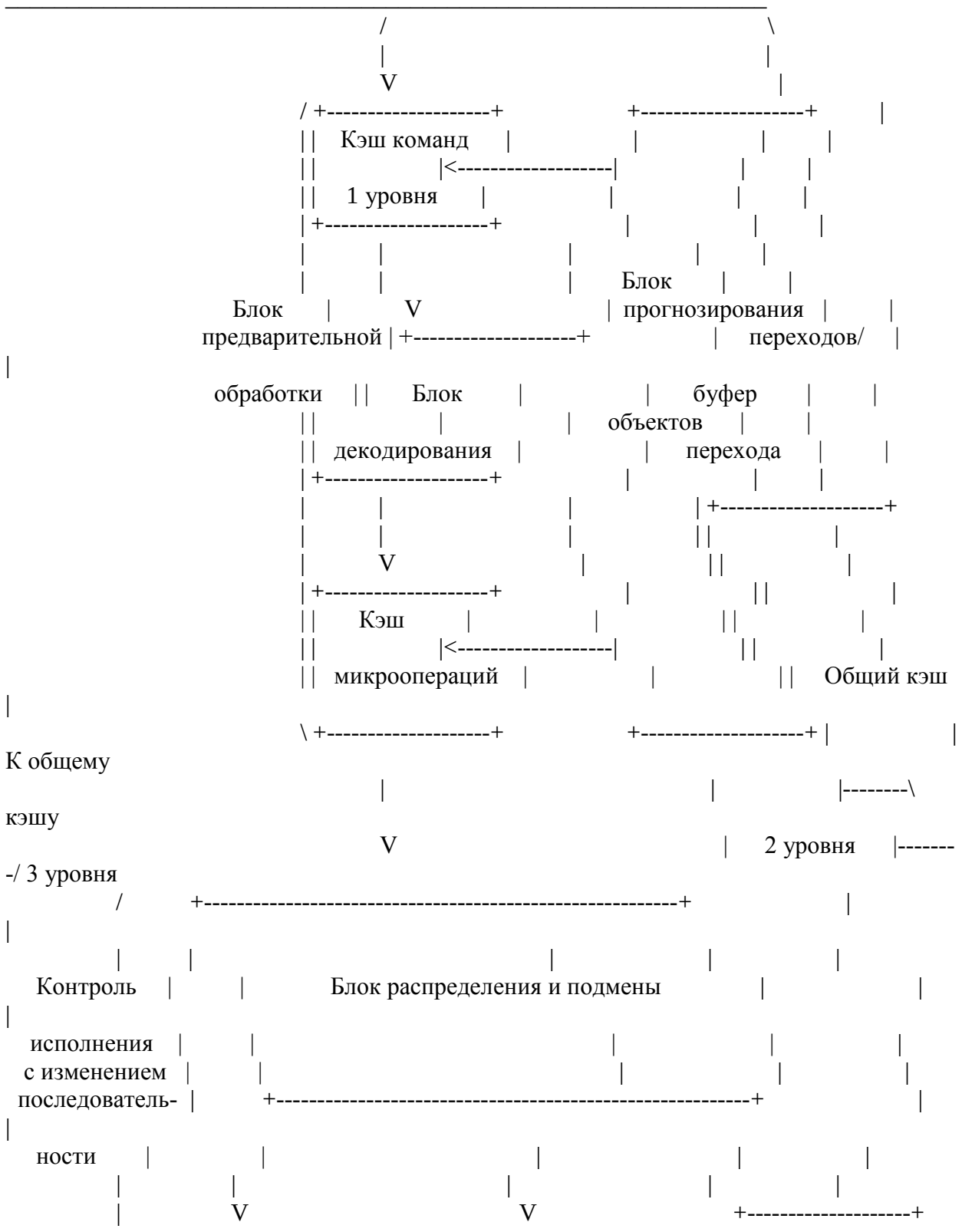
Хотя очередность выдачи команд может отличаться от предусмотренной в программе, требование точности прерываний архитектуры Core i7 гласит, что результаты выполнения ISA-команд
(Instruction Set Architecture) должны становиться видимыми программе без отступления от заданной
программой последовательности. За реализацию этого требования отвечает блок пересортировок.

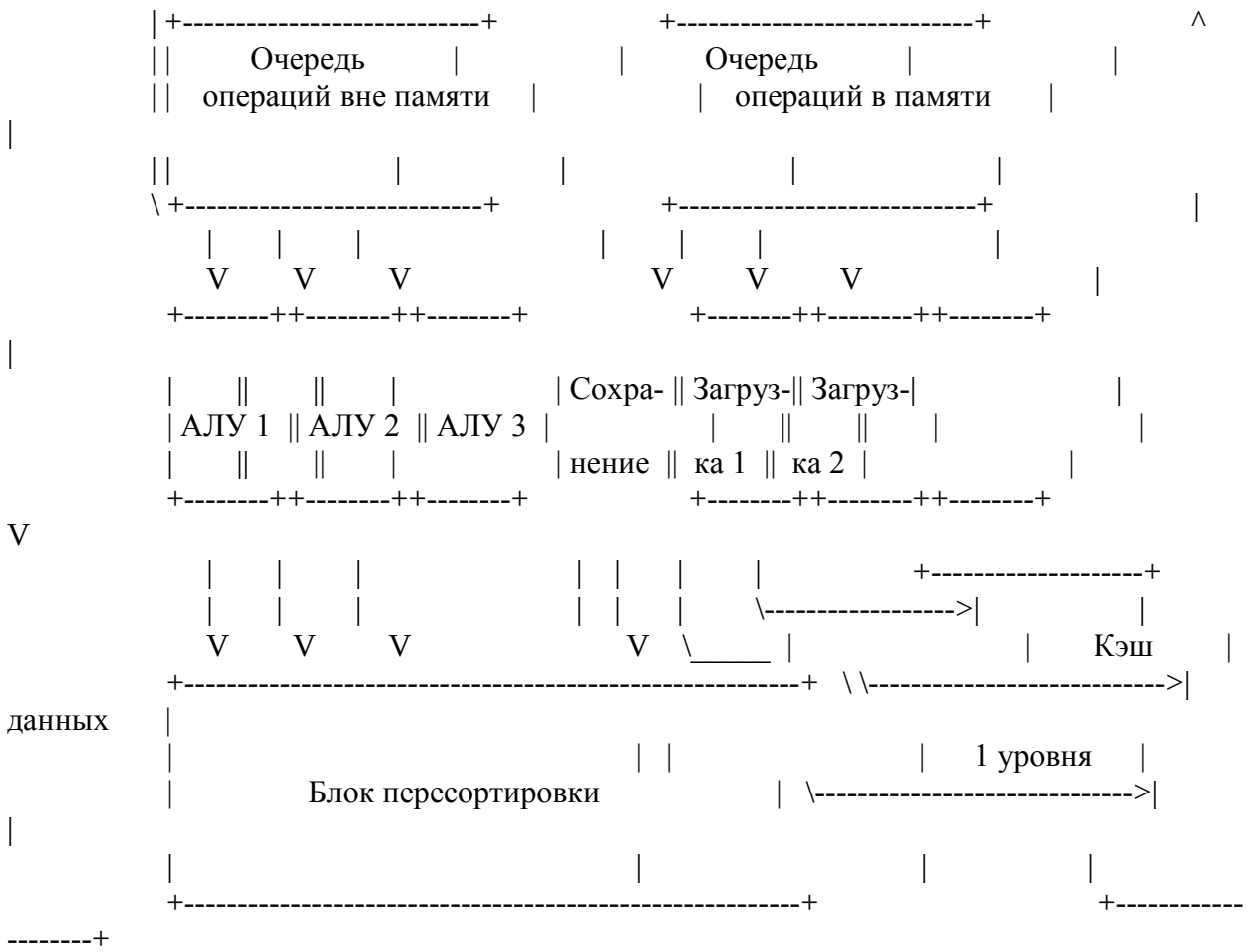
Блок исполнения.

Блоки исполнения непосредственно осуществляют целочисленные операции, операции с плавающей точкой и специализированные команды. Существуют несколько блоков исполнения,

работающих параллельно. Данные они получают из регистрового файла и кэша данных первого
уровня.

Конвейер Sandy Bridge у Core i7.





На рисунке выше приведена упрощенная схема микроархитектуры Sandy Bridge, в том числе её конвейер.
В верхней части схемы находится блок предварительной обработки, ответственный за выборку команд
из памяти и их подготовку к исполнению. Этот блок получает новые команды x86 из кэша команд
первого уровня. Они декодируются в микрооперации и помещаются в кэш микроопераций, содержащий
приблизительно 1,5К микроопераций.

Если блок декодирования сталкивается с условным переходом, он обращается за информацией к блоку
прогнозирования переходов. Этот блок содержит историю переходов, осуществлявшихся в прошлом,
и на основании накопленных данных предполагает, будет ли выполнен условный переход, когда
он в следующий раз встретится в программе. Здесь используются проприетарные алгоритмы Intel.

Если команда перехода отсутствует в таблице, применяется статическое прогнозирование. При этом подразумевается, что обратный переход, во-первых, является частью цикла, во-вторых,
по умолчанию предполагается, что он будет выполнен. Точность статического прогноза в этом

случае высока. Прямой переход считается входящим в структуру оператора if и не выполняемым
по умолчанию. Точность статического прогноза в случае прямых переходов значительно ниже,
чем в случае обратных.

Для выбранной ветви целевой адрес определяется по содержимому буфера объектов перехода.
В буфере объектов перехода хранится целевой адрес перехода при последнем выполнении.
Обычно этот адрес правилен (он всегда правилен для переходов с постоянным смещением).
Косвенные переходы осуществляются по разным адресам и их прогнозирование по данным буфера объектов перехода будет ошибочным.

Второй компонент конвейера - логика исполнения с изменением последовательности - получает данные из кэша микроопераций.
При поступлении их блока предварительной обработки каждой последующей микрооперации (за цикл их поступает три)
блок распределения и подмены регистрирует её в таблице, состоящей из 168 записей и называемой буфером переупорядочивания команд.
В этом буфере хранятся данные о состоянии микроопераций, вплоть до пересортировки её результатов.
Затем блок распределения и подмены проводит проверку на предмет доступности ресурсов, необходимых для выполнения микрооперации.
Если ресурсы свободны, микрооперация устанавливается в одну из очередей планировщика.
Для микроопераций, исполняемых в памяти и вне памяти, предусмотрены отдельные очереди.
Если исполнение микрооперации в данный момент невозможно, она откладывается, однако обработка последующих микроопераций продолжается;
таким образом, микрооперации часто исполняются вне их исходной последовательности.
Этот принцип позволяет поддерживать загрузку
всех функциональных блоков на максимально высоком уровне. В каждый отдельно взятый момент могут одновременно обрабатываться до 154 команд,
причем 64 из них могут загружаться из памяти, а 36 - сохраняться в памяти.

Иногда микрооперации простаивают. Это происходит в тех случаях, когда к одному и тому же регистру для чтения или записи пытаются
обратиться несколько микроопераций; соответственно, одной из них это удается, а остальным - нет.
Такие конфликты называются WAR и WAW взаимозависимостями. Подмена целевого регистра позволяет записать результаты исполнения
микрооперации в один из 160 временных регистров, а значит, выполнить эту микрооперацию немедленно.
Если же временные регистры недоступны или микрооперация попадает в ситуацию RAW (read after write) взаимозависимости (обойти которую нельзя),
планировщик указывает характер возникшей проблемы в виде записи в буфере переупорядочивания команд.
Впоследствии, после освобождения всех необходимых ресурсов, микрооперация устанавливается в одну из очередей на исполнение.

Очереди планировщика помещают готовые к исполнению операции в один из шести функциональных блоков:
1. АЛУ 1 и блок умножения с плавающей точкой;
2. АЛУ 2 и блок сложения/вычитания с плавающей точкой;
3. АЛУ 3, блок обработки переходов и сравнений с плавающей точкой;
4. Команды сохранения;
5. Команды загрузки 1;
6. Команды загрузки 2.

Три целочисленных АЛУ не одинаковы. АЛУ 1 выполняет любые арифметические и логические операции, умножения и деления.
АЛУ 2 способно выполнять только арифметические и логические операции. АЛУ 3 выполняет арифметические и логические операции,
а также разрешение переходов. Не идентичны и два блока исполнения операций с плавающей точкой.
Первый поддерживает арифметические операции с плавающей точкой, включая умножение, а второй способен
выполнять только сложение и вычитание с плавающей точкой, а также перемещения.

АЛУ и блоки исполнения операции с плавающей точкой получают данные от двух регистровых файлов емкостью по 128 записей.
Один из этих файлов отводится для целых чисел, другой - для чисел с плавающей точкой. В них содержатся все операнды,
необходимые для исполнения команд; кроме того, они играют роль хранилища результатов. В силу подмены регистров,
восемь из них содержат регистры, доступные на уровне архитектуры команд (EAX, EBX, ECX, EDX и т. д.),
однако расположение "реальных" значений в каждом конкретном случае зависит от изменений в отображении, происходящих в ходе исполнения.

Кэш данных первого уровня тесно связан с внутренней конвейерной подсистемой Sandy Bridge. В этом кэше емкостью 32 Кбайта могут храниться целые числа,
числа с плавающей точкой и другие типы данных. В отличие от кэша микроопераций, эти данные никоим образом не декодируются.
Функция кэша данных сводится к хранению копий байтов, находящихся в памяти. Что касается его характеристик, то кэш данных первого уровня
представляет собой 8-входовую ассоциативную кэш-память с емкостью строки 64 байта. Он поддерживает сквозную запись;
иными словами, при изменении строки кэша она незамедлительно копируется обратно в кэш второго уровня (write-through).
В течение цикла кэш данных первого уровня может выполнить две операции чтения и одну операцию записи.
Для реализации множественных обращений используются банки, то есть кэш делится на несколько внутренних кэшей (8 в случае Sandy Bridge).
Если все три обращения относятся к разным банкам, они могут выполняться одновременно; в противном случае одно из обращений к конфликтующим банкам простаивает.
Если затребованное слово не удается обнаружить в кэше первого уровня, отправляется запрос в кэш второго уровня;
последний в такой ситуации либо отвечает сразу, либо обращается к общему кэшу третьего уровня, после чего отвечает.

В любой момент в состоянии исполнения могут находиться до десяти запросов, направленных из кэша первого уровня в кэш второго уровня.

Так как микрооперации исполняются вне исходной последовательности, сохранение в кэше первого уровня возможно только после пересортировки результатов всех команд, предшествующих команде сохранения. Такую пересортировку результатов с их трассировкой (отслеживанием того, где они находятся) выполняет блок пересортировки. В случае прерывания прекрашается обработка всех команд, ещё не прошедших пересортировку результатов;
таким образом, обеспечивается соблюдение требования, согласно которому при прерывании должны быть завершены все команды до определённой точки в программе (точность прерываний).

Если команда сохранения прошла пересортировку результатов, но предшествующие команды ещё обрабатываются, из-за невозможности обновления кэша первого уровня результаты их исполнения передаются в буфер незавершенных команд. В этом буфере можно единовременно разместить до 36 команд сохранения.
Если одна из последующих команд загрузки попытается считать сохраненные данные, она из буфера незавершенных команд будет перенаправлена непосредственно к команде, которая в этот момент ещё не помещена в кэш данных первого уровня. Этот процесс называется перенаправлением для загрузки (store-to-load forwarding).

##
###

45. Процессоры Intel в реальном режиме: регистры процессора, управление памятью и программами, данные и способы адресации, система команд, система прерываний.

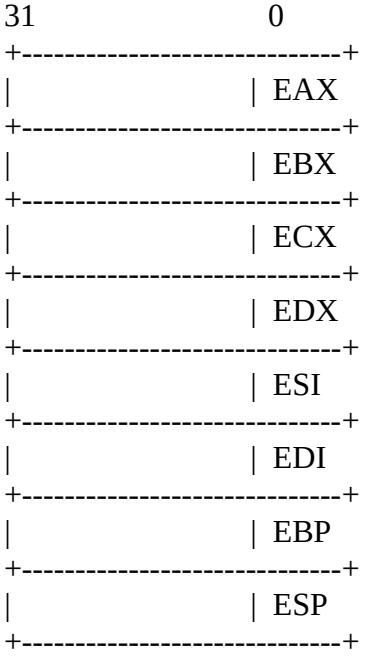
Любой Intel CPU входит в режим реальной адресации памяти (real mode) после сброса (reset).

Модель памяти реального режима Intel CPU.

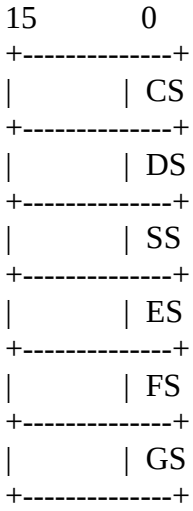
Программа видит память как набор независимых адресных пространств, называемых сегментами.
Код, данные и стек обычно располагаются в различных сегментах. Чтобы адресовать байт в сегменте,
программа применяет логический адрес, состоящий из сегментного селектора и смещения. Сегментный селектор
определяет сегмент, к которому нужно обратиться. Смещение - байт в адресном пространстве данного сегмента.
В реальном режиме используются сегменты, состоящие из 64 Кбайта каждый. Физический адрес = Сегментный селектор << 4 + Смещение.
CPU не предоставляет какой-либо защиты памяти. Любая программа способна адресовать любой сегмент в пределах 20-битного адресного
пространства, в частности, любая программа способна обращаться к MMIO (взаимодействие с устройствами) и править вектора прерываний.
В реальном режиме размер адреса и операнда по умолчанию равен 16 битам. Префикс "переопределение размера адреса" (67h) может быть использован в реальном режиме,

чтобы включить 32-х битную адресацию. Тем не менее, максимально допустимый 32-х битный линейный адрес (сумма сегментного селектора и смещения) по прежнему равен 0x000FFFFFh.
Префикс "переопределение размера операнда" (66H) позволяет использовать в вычислениях 32-х битные регистры (EAX, EBX, ...), а загружать из памяти двойные слова (4 байта).

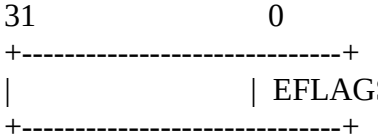
Регистры общего назначения.



Сегментные регистры.



Слово состояния программы и управляющий регистр.



Указатель команд.

* EAX - аккумулятор для операндов и результатов операций;
* EBX - указатель на данные в сегменте DS;
* ECX - счётчик для операций со строками и циклов;
* EDX - указатель I/O;
* ESI - указатель на данные в сегменте DS; указатель на источник для строковых инструкций;
* EDI - указатель на данные в сегменте ES; указатель на приёмник для строковых инструкций;
* ESP - указатель на вершину стека (в сегменте SS);
* EBP - указатель на данные, хранящиеся в стеке (в сегменте SS).

Сегментные регистры (CS, DS, SS, ES, FS и GS) содержат 16-битные селекторы сегментов. Сегментный
селектор - это специальный указатель, определяющий сегмент в памяти. Для доступа к конкретному сегменту,
сегментный селектор для этого сегмента должен присутствовать в соответствующем сегментном регистре.
В реальном режиме сегментный селектор равен линейному адресу сегмента, делённому на 16.
Каждый из сегментных регистров связан с одним из трёх типов памяти: код, данные и стек.
К примеру, регистр CS содержит сегментный селектор для сегмента кода, где хранятся инструкции,
на данный момент исполняемые CPU. CPU извлекает инструкции из сегмента кода, используя логический адрес,
состоящий из сегментного селектора CS и указателя команд EIP.

Регистры DS, ES, FS и GS указывают на 4 сегмента данных, для их использования в инструкции требуется
специальный префикс переопределения сегмента.

Регистр SS указавает на сегмент стека, все операции со стеком (push*, pop*) используют SS для
нахождения сегмента стека.

32-х битный регистр EFLAGS содержит набор статусных флагов, управляющих флагов, а также системные флаги.

Способы адресации:

1) Непосредственные операнды -- кодирование данных (констант) непосредственно в самой инструкции;
2) Регистровые операнды -- источник или приёмник данных может быть регистром.
3) Операнды в памяти -- данные извлекаются из оперативной памяти (а по факту, из кэша :))

Адрес значения, хранящегося в памяти, формируется следующим образом:

Сегментный селектор -- указывается явно или неявно. Наиболее распространённый способ указания сегментного
селектора - это загрузить его в сегментный регистр и затем позволить процессору обращаться к регистру неявно,
в зависимости от типа исполняемой инструкции.
Правила выбора сегмента по умолчанию:

Тип обращения (к памяти)	Используемые регистр	Используемый сегмент	Когда выбирается?
Инструкции	CS	Сегмент кода	Любая выборка инструкций
Стек	SS	Сегмент стека	Все инструкции push* и pop*
			Любые обращения к памяти, использующие регистры ESP и EBP в качестве базы
Данные	DS	Сегмент данных	Любые обращения к данным, за исключением данных, относящихся к стеку или строке-приёмнику
Строки-приёмники	ES	Сегмент данных, указываемый сегментным регистром ES	Приёмники строковых инструкций

Сегментный регистр, используемый для выбора сегмента, можно изменить, указав специальный префикс "переопределения сегмента".
Следующие правила выбора сегмента по умолчанию не могут быть переопределены:
* Извлечение инструкций должно быть осуществлено из сегмента кода.
* Строки-приёмники могут быть адресованы в строковых операциях только из сегмента ES.
* Операции со стеком (push* и pop*) могут адресовать только сегмент стека SS.

Смещение -- состоит из следующих компонент:
* Сдвиг - 8-, 16-, или 32-х битное значение.
* База - значение, хранящееся в регистре общего назначения.
* Индекс - значение, хранящееся в регистре общего назначения.
* Масштаб - значение, равное 1, 2, 4 или 8, которое умножается на величину "Индекса".

Смещение, получающееся в результате сложения этих компонент, называется эффективным адресом.
Каждый из данных компонент может иметь либо положительное либо отрицательное (дополнение до 2) значение,
за исключением "Масштаба".
Следующая схема описывает способы, которыми компоненты могут быть объединены для формирования
эффективного адреса:

СМЕЩЕНИЕ=БАЗА+(ИНДЕКС*МАСШТАБ)+СДВИГ

Система команд.

Система команд x86, исполняющихся в реальном режиме, включает в себя следующие категории:
* инструкции общего назначения;
* FPU инструкции.

Инструкции общего назначения осуществляют базовое перемещение данных, арифметические и логические
операции, инструкции перехода, а также строковые операции. Они работают с данными, хранящимися
в памяти, в регистрах общего назначения и в регистре EFLAGS. Также они работают с адресной информацией,
хранящейся в памяти, регистрах общего назначения, а также сегментных регистрах.
Данная группа инструкций включает в себя:
* перемещение данных,
* двоичную целочисленную арифметику,
* двоично-десятичную арифметику,
* логические операции,
* операции сдвига и вращения,
* операции над битами и байтами,
* управление программным потоком,
* строковые операции,
* управление флагами (регистр EFLAGS),
* операции над сегментными регистрами,
* ввод/вывод (IO ports),
* прочие подгруппы.

FPU инструкции.

X87 FPU инструкции исполняются вещественным сопроцессором. Данные инструкции работают с вещественными числами, целыми числами,
а также числами, представленными в двоично-десятичном формате.
Сюда входят такие инструкции как:
* перемещение данных,
* загрузка констант,
* управляющие FPU инструкции,
* арифметические инструкции.

Система прерываний.

Прерывание - асинхронное событие, обычно генерируемое устройством ввода/вывода.
Исключение - синхронное событие, генерируемое процессором в случае обнаружения некоторого условия.

Процессор реагирует на прерывания и исключения одинаково. Когда прерывание или исключение обнаружено,
процессор приостанавливает выполнение текущей программы или задачи и переключается на специализированную
процедуру-обработчика. Процессор обращается к обработчику через запись в специальной таблице, называемой
"таблицей дескрипторов прерываний" (таблица векторов прерываний в реальном режиме). Всего возможно
256 различных прерываний, с номерами от 0 до 255. Некоторые из этих номер зарезервированы архитектурой,
другие же доступны для использования для внешних устройств.

##
###

46. Процессоры Intel в защищенном режиме: регистры процессора, управление памятью, поддержка многозадачности и защита памяти

Базовая модель исполнения аналогична реальному режиму, за следующими исключениями:
* Размер операнда и адреса, по умолчанию, равен 32 битам. Появляется возможность адресации до 4 Гб памяти (включая MMIO).
* Уровни привилегий исполняющегося кода: 0 - ядро операционной системы, 3 - пользовательские программы, 1, 2 - промежуточные уровни.
* Селекторы сегментов теперь являются, по сути, индексами в таблицах дескрипторов; младшие три бита имеют особую семантику
(обозначают конкретную таблицу, где искать дескриптор сегмента - локальную или глобальную таблицу дескрипторов, а также уровень привилегий - запрашиваемый или реальный).
Дескрипторы сегментов теперь, помимо базового адреса, содержат информацию о размере сегмента, а также о допустимом уровне привилегий.
Уровни привилегий определяют, какие инструкции может выполнять текущая задача, а также какие действия ей доступны в системе (например, может ли она отключать прерывания).

* Страничная адресация и поддержка виртуальной памяти - все обращения к памяти проходят через MMU, который прозрачно для программного обеспечения осуществляет подмену
адреса.
* Таблица векторов прерываний теперь заменена на таблицу дескрипторов прерываний. Адрес таблицы дескрипторов прерываний хранится в специальном регистре CPU. Дескриптор прерывания, помимо адреса обработчика, содержит также служебную информацию (как например, с каким уровнем привилегий можно вызывать обработчик прерывания по инструкции int).

##
###

47. Аппаратно-программная модель процессоров IA-64 и Intel64: регистры процессора, управление памятью и программами, данные и способы адресации, система команд.

https://en.wikipedia.org/wiki/X86-64

Рассмотрим теперь файлы регистров IA-64. В их число входят: 128 регистров общего назначения GR; 128 регистров с плавающей запятой FR; 64 регистра предикатов PR; 8 регистров перехода BR; 128 прикладных регистра AR; не менее 4 регистров идентификатора процессора CPUID; счетчик команд IP, указывающий на адрес связки, содержащей исполняемую команду; регистр маркера текущего окна CFM, описывающий окно стека регистров и др.
Регистры CPUID являются 64-разрядными. В CPUID-регистрах 0 и 1 лежит информация о производителе, в регистре 2 находится серийный номер процессора, а в регистре 3 задается тип процессора (семейство, модель, версия архитектуры и т.п.) и число CPUID-регистров. Разряды регистра 4 указывают на поддержку конкретных особенностей IA-64, т.е. тех, которые реализованы в данном процессоре.
Прикладные регистры AR0-AR127 - специализированные (в основном 64-разрядные) регистры, применяемые в IA-64 и IA-32. AR0-7 называются регистрами ядра; запись в них привилегирована, но они доступны на чтение в любом приложении и используются для передачи приложению сообщений от операционной системы.

Все рассматриваемые команды можно подразделить на: команды работы со стеком регистров (например, alloc); целочисленные команды; команды сравнения и работы с предикатами; команды доступа в память; команды перехода; мультимедийные команды; команды пересылок между регистрами; "разные" (операции над строками и подсчет числа единиц в слове); команды работы с плавающей запятой.

##
###

48. Аппаратно-программная модель процессоров ARM: регистры процессора, управление памятью и программами, данные и способы адресации, система команд.

https://wiki.osdev.org/ARM_Overview

##
###

49. Операционные системы: подходы к определению операционной системы как вида программного обеспечения,

функции операционных систем, архитектурные типы, современные тенденции в развитии операционных систем.

==========
==========

__Операционная система__ - вид программного обеспечения, управляющего работой вычислительной системы, а также
расширяющего возможности ЭВМ.

ПОДХОДЫ
1) ОС как расширенная машина.

ОС предоставляет сервисы приложениям пользовательского уровня:
- Системные вызовы.
Системные вызовы - это, по сути, механизм, с помощью которого пользовательские приложения запрашивают
выполнение некоторого действия от ОС. Типичными примерами системных вызовов являются порождение дочернего процесса,
открытие/закрытие нового файла, запись/чтение в/из файл(а).
- Абстракции (файлы, сокеты) для доступа к устройствам ввода/вывода.
Ключевой особенностью является единообразие интерфейса независимо от конкретного оборудования
(сетевая карта, жёсткий диск). В данном случая ядро (kernel) операционной системы преобразует
системные вызовы в обращения к конкретным драйверам периферийного оборудования. За счёт этого
упрощается разработка программного обеспечения.
- Виртуализация. Концепция "процесса" как программы, исполняющейся на виртуальном процессоре, позволяет предоставить программе
видение, будто она единолично исполняется на процессоре и в её распоряжении находится всё оперативная память.
Здесь важно отметить, что без аппаратной поддержки (MMU, прерывания) данные возможности не были бы доступны.
Таким образом, ОС расширяет возможности компьютера, является как бы "новым уровнем" в архитектуре компьютера.

2) ОС как менеджер ресурсов.

В данном подходе считается, что основная задача ОС - упорядоченное и управляемое распределение ресурсов:
процессора;
оперативной памяти;
устройств ввода-вывода
-- между различными программами, претендующими на их использование.

Функции операционных систем:

1) Распределение ресурсов программам -- "процессам": процессорное время, память, устройства ввода/вывода.
2) Управление устройствами компьютера: обслуживание прерываний, конфигурирование оборудование (plug & play).
3) Предоставление сервиса пользовательским приложениям -- выполнение системных вызовов.
4) Виртуализация -- поддержка виртуализации (примеры: VirtualBox, Qemu/KVM)
5) Безопасность -- управление доступом, системы разграничения доступа пользователей, уровни привилегий.
6) Упрощение разработки программного обеспечения -- предоставление библиотек (shared objects), упрощение загрузки программ, облегчение переносимости кода.

Архитектурные типы ("Зоопарк"):

* ОС мейнфреймов
* ОС серверов
* ОС персональных компьютеров
* ОС встраиваемых систем
* ОС реального времени

Тенденции:

* Поддержка большего числа архитектур CPU.
* Поддержка большего количества периферийных устройств.
* Оптимизация кода: внедрение большого числа микро- и архитектурно-зависимых оптимизаций.
* Сохранение обратной совместимости с пользовательским ПО (сохранение бинарного интерфейса системных вызовов).
* Hardening (усиление безопасности, фикс уязвимостей, внедрение новых мер безопасности).
* Сетевые возможности: поддержка многих сетевых карт, реализация известных сетевых протоколов канального, сетевого, транспортного уровней, добавление новых возможностей по конфигурированию сети (пример, nl80211 -- конфигурирование Wi-Fi, netlink -- замена интерфейса ioctl'ов для сетевых модулей).
* Поддержка аппаратной виртуализации (KVM -- поддержка Intel VT-x).

##
###

50. Управление процессами и потоками: представление процессов и потоков в операционных системах, дисциплины планирования процессов, взаимодействие процессов, проблема тупиков.

__Процесс__ - абстракция, описывающая исполняемую на компьютере программу. Процесс является единицей управления ресурсами: иными словами, ОС планирует ресурсы ЭВМ на уровне процессов (за исключением, CPU). Каждый процесс характеризуется своей собственной виртуальной памятью, а также своими ресурсами ввода/вывода (открытые файлы, используемые сокеты).

__Поток__ - исполняемая единица. Любой поток принадлежит некоторому процессу, исполняется в его контексте. При этом каждый процесс может иметь несколько потоков -- таким образом реализуется многопоточность приложений. ОС распределяет ресурсы CPU на уровне потоков. Каждый поток при этом характеризуется своим собственным контекстом исполнения (регистры, стек).

__Планирование__ - определение следующего выполняющегося потока (а следовательно, процесса) на CPU.

Алгоритмы (дисциплины) планирования.

Планирование в пакетных (=неинтерактивных) системах
1) FIFO. Запускаемые процессы помещаются в очередь (FIFO). Следующим исполняемым процессом на CPU выбирается тот, кто располагается самым первым в очереди. При использоваии данного алгоритма центральный процессор выделяется процессам в порядке поступления их запросов. При этом процесс исполняется на CPU до тех пор, пока не будет заблокирован по причине выполнения операции ввода/вывода или пока не завершится.
2) Самое короткое задание первым. Следующим исполняемым на cpu заданием будет процесс, требующий самое короткое время для своего завершения. Эффективен, если задания доступны планировщику одновременно. В случае последовательного поступления заданий превращается в алгоритм FIFO.
3) Приоритет наименьшему времени выполнения. Следующим исполняемым на cpu заданием будет процесс, оставшееся время исполнения которого минимально.

Планирование в интерактивных системах:
4) Циклическое планирование. Каждому процессу выделяется квант времени, в течение которого он может исполняться. По истечении этого кванта времени, процесс прерывается, и ресурсы CPU
переходят другому процессу. В случае если процесс перешел в заблокированное состояние по вводу/выводу или завершился, то планирование осуществляется именно в этот момент. По исчерпании кванта времени,
прерванный процесс помещается в конец FIFO. Следующим исполняемым процессом выбирается тот, кто находится в начале FIFO.
5) Приоритетное планирование. Аналогичен алгоритму циклического планирование, за исключением того, что каждому процессу присваивается приоритет. Следующим исполняемым на CPU процессом выбирается тот,
который находится в состоянии готовности (не заблокирован вводом/выводом) и имеет наивысший приоритет. В качестве реализации может быть использовано несколько очередей (FIFO).
Процессы разного приоритета помещаются в разные очереди.
6) Гарантированное планирование. Учитываются 3 характеристики:
t - время, в течение которого процесс исполнялся на CPU,
T - время, прошедшее с момента запуска процесса,
n - количество исполняющихся процессов.
ОС исходит из идеи, что каждый процесс имеет право на время CPU в размере t' = T/n (каждому процессу выделяется одинаковое количество процессорного времени).
Для каждого процесса подсчитываются величины d = t/t' = (t * n) / T. Следующим исполняемым процессом выбирается тот, у которого величина d наименьшая:
такой процесс "недополучил" больше процессорного времени, чем его конкуренты.
7) Лотерейное планирование. Следующий процесс выбирается при помощи датчика случайных чисел. Возможна настройка вероятности "быть выбранным", путём задания численных приоритетов и их учёта.

Взаимодействие процессов.

Одним из свойством процесса является его "изолированность" от других процессов: таким образом достигается иллюзия единоличного владения ресурсами компьютера.
Тем не менее, возникает необходимость обмена данными между различными процессами.
Существуют следующие механизмы межпроцессорного взаимодействия (на примере ОС Linux):

1) Использование разделяемых областей памяти. Нескольким процессам отображается в виртуальное адресное пространство одна и та же физическая страница оперативной памяти.
За счёт этого появляется возможность взаимодействия процессов через выделенную область памяти.
Использование разделяемой памяти доступно в ОС Linux через системный вызов mmap с флагом MAP_SHARED.
2) Сигналы. Сигналы UNIX (signals) являются ограниченным, но полезным способом межпроцессорной коммуникации. По сути, сигнал - это аналог прерывания для CPU.
При поступлении сигнала вызывается специальный обработчик процесса, который выполняет некоторые действия. Одни сигналы имеют чёткую семантику (как например, SIGALRM - сигнал от таймера),
семантика других же определяется самим процессом (как например, SIGUSR1, SIGUSR2).
3) UNIX-сокеты. Механизм работы данного способа межпроцессорного взаимодействия аналогичен обычным TCP/UDP сокетам. Один процесс (сервер) прослушивает определённый сокет,
идентифицируемый путём в VirtualFS (VFS). Другой процесс (клиент) "подключается" к этому сокету и посылает через него сообщения определённого формата.
Процесс-сервер получает данное сообщение и выполняет соответствующие действия.
Подобный способ межпроцессорного взаимодействия использует большинство демонов (daemons) ОС.
4) Канал (pipe). Канал - механизм межпроцессорного взаимодействия между процессом-родителем и дочерним процессом. Процесс-родитель записывает данные в один из концов канала, идентифицируемый
файловым дескриптором на запись (fd). Дочерний процесс же считывает эти данные с другого конца (файловый дескриптор на чтение). Обычно создаваемые файловые дескрипторы заменяют stdin или stdout процесса,
поэтому межпроцессорное взаимодействие эффективно выглядит как получение пользовательского ввода/вывода.
5) Коммуникация через файлы. Обычные файлы VFS тоже могут являться способом взаимодействия процессов. При этом коммуницировать можно как через содержимое файла, так и через сам факт наличия файла.
Последнее используется в так называемых lock или pid файлах, которые свидетельствуют о наличии определённого процесса или о выполнении определённого действия в системе.
6) Сетевые сокеты TCP/UDP как способ взаимодействия удалённых процессов (исполняющихся на различных машинах).

Проблема тупиков.

При использовании разделяемых областей памяти (или вообще любых разделяемых ресурсов, к примеру, принтера) появляется необходимость синхронизации процессов: иными словами, требуется некоторая сериализация выполняемых над разделяемых ресурсом операций так, чтобы исключить одновременное его использование.

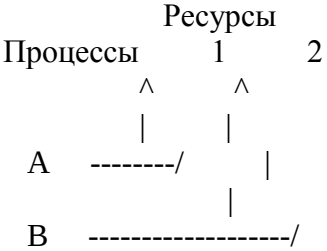
Участок кода программы, работающий с разделяемым ресурсом (например, памятью), называется __критической секцией__.

Для безопасного использования разделяемого ресурса используются дополнительные переменные, семантика которых аналогична "светофору": одно значение переменной, к примеру 0, говорит о том, что ресурс в настоящие момент не используется; другое значение, к примеру любое ненулевое значение, говорит о том, что ресурс в настоящее время занят.

Необходимым условием возможности использования переменных для синхронизации является аппаратная поддержка атомарной операции __"test-and-set"__.
Данная операция выполняется по следующему алгоритму _атомарно_:
1) В переменную записывается ненулевое значение (например, 1).
2) Предыдущее значение переменной сохраняется в специальном регистре.
Программное обеспечение при этом имеет возможность узнать, успешно ли была взята блокировка, по значению, сохранённому в регистре.

Для сериализации исполняемых операций используются следующие примитивы синхронизации:
1) Спинлок (spinlock) -- процесс пытается циклически выполнить операцию "test-and-set" до тех пор, пока не преуспеет.
2) Мьютекс (mutex) -- процесс прерывается операционной системой в случае неудачного выполнения операции "test-and-set".
В случае успеха процесс продолжает свою работу.
3) Семафор (semaphore) -- счётчик. Для его реализации часто задействуется атомарная операция compare-and-add, помещающее в переменную не новое ненулевое значение, а сумму предыдущего и указанного в коде операции.
По сути подсчитывает оставшееся число разделяемого ресурса. В случае его отсутствия (счётчик = 0) процесс также прерывается. Мьютекс иначе называется бинарным семафором.

__Проблема тупиков__ возникает при наличии нескольких процессов и нескольких разделяемых ресурсов.
Пример тупика:



Стрелочками нарисованы взятые разделяемые ресурсы 1, 2 каждым из процессов A и B. После этого, если процесс A попытается взять ресурс 2, а процесс B - ресурс 1, возникнет ситуация тупика (__deadlock__).

Условия возникновения тупиков:
1) Условие взаимного исключения. Каждый ресурс либо выделен в данный момент только одному процессу, либо доступен.
2) Условие удержания и ожидания. Процессы, удерживающие в данный момент ранее выделенные им ресурсы, могут запрашивать новые ресурсы.

3) Условие невыгружаемости. Ранее выделенные ресурсы не могут быть принудительно отобраны у процесса. Они должны быть явным образом
высвобождены тем процессом, который их удерживает.
4) Условие циклического ожидания. Должна существовать кольцевая последовательность из двух и более процессов, каждый из которых ожидает высвобождения ресурса, удерживаемого следующим членом последовательности.

Для возникновения тупика должны соблюдаться все четыре условия.

##
###

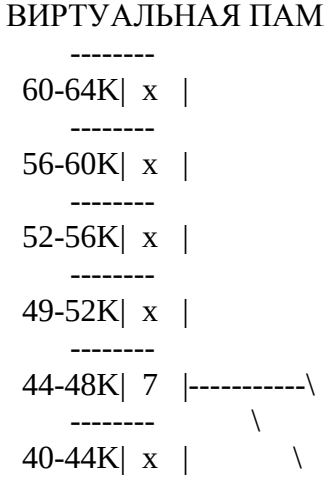
51. Управление оперативной памятью: управление физической и виртуальной памятью, реализация свопинга.

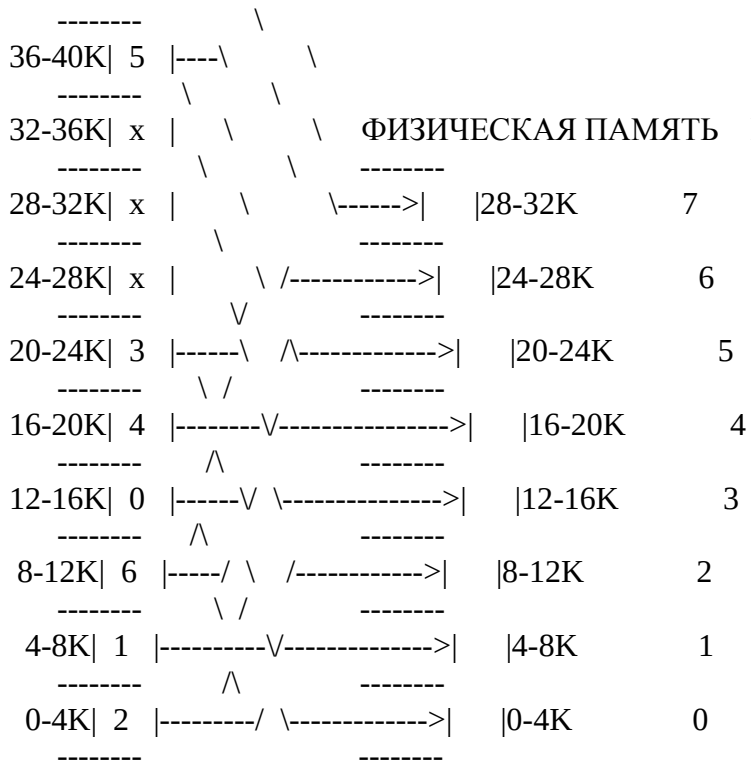
Современные CPU имеют одним из компонент диспетчер памяти (MMU, Memory Management Unit).
Задача MMU -- преобразовывать поступающие на входе виртуальные адреса в физические, реально выставляемые на адресных линиях системной шины.
Данный аппаратный компонент является основой построения виртуальной памяти.
Программное управление диспетчером памяти осуществляется через специальные регистры CPU, а также через структуры данных, находящиеся в оперативной памяти (такие структуры данных описывают правила преобразования виртуального адреса в физический).

Виртуальная память:

Виртуальное адресное пространство состоит из блоков фиксированного размера (обычно 4 Кб), называемых страницами.
Соответствующие блоки в физической памяти называются страничными блоками.
Страницы и страничные блоки имеют, как правило, одинаковые размеры. Перенос информации между оперативной памятью и диском (при свопинге) всегда осуществляется целыми страницами.

Пример организации виртуальной памяти (надо просто понять суть по диаграмме, учить дословно необязательно :)):





На диаграмме выше приняты следующие обозначения: диапазон, помеченный 0-4K означает, что виртуальные или физические адреса этой страницы составляют от 0 до 4095. Диапазон 4-8K ссылается на адреса от 4096 до 8191 включительно и так далее. Каждая страница содержит строго 4096 адресов, которые начинаются с чисел, кратных 4096, и заканчиваются числама на единицу меньше чисел, кратных 4096.
К примеру, когда программа обращается к памяти по адресу 0, диспетчеру памяти поступает на вход виртуальный адрес 0.
Диспетчер памяти вычисляет, что адрес относится к нулевой виртуальной странице. Данной странице соответствует физический блок с PFN=2.
Соответственно, на адресные линии выставляет значение 8192. Оперативная память не знает о существовании диспетчера и видит только запрос на чтение/запись по адресу 8192, который и выполняет. Таким образом, диспетчер памяти эффективно справляется с отображением всех виртуальных адресов в диапазоне [0, 4095] на физические адреса [8192, 12 287]. По аналогии происходят трансляции и в других виртуальных страницах.

Важно отметить, что во всех случаях отображение имеет характер один-к-одному (биекция).

Сама по себе возможность отображения 16 виртуальных страниц на 8 страничных блоков за счёт соответствующей настройки таблиц диспетчера
не решает проблемы превышения объема виртуальной памяти над объемом физической памяти. Поскольку в нашем распоряжении только 8 физических страничных блоков, то на физическую память могут отображаться только 8 виртуальных страниц. Остальные страницы, помеченные на рисунке крестиками, в число отображаемых не попадают. Реальное оборудование отслеживает присутствие конкретных страниц в физической памяти за счёт __бита присутствия-отсутствия__.

В случае если программа обращается к странице, на данный момент не отображенной в оперативную память, диспетчер памяти генерирует исключение "отсутствие страницы" (page fault). Операционная система выбирает редко используемый страничный блок и

сбрасывает его содержимое на диск. Затем она извлекает с диска запрошенную страницу, помещает её в только что освободившийся страничный блок, вносит изменения в таблицы трансляций и заново запускает прерванную инструкцию.

Алгоритмы замещения страниц.

В случае возникновения `page fault`-исключения, ядро операционной системы должно выбрать страницу, которую требуется поместить на диск.
Есть различные подходы к тому, какую именно страницу необходимо выбрать:

1) NRU (Not Recently Used, исключение давно не использовавшейся страницы). Данный алгоритм реализуется на основе двух битов дескрипторов страниц, управляемых как аппаратно, так и программно:
* R - устанавливается при каждом обращении к странице;
* M - устанавливается при изменении страницы.

При запуске процесса оба страничных бита по умолчанию равны 0.
Периодически (например, по прерыванию от таймера) бит R сбрасывается, чтобы отличить те страницы,
к которым в последнее время были обращения от тех, к которым обращений не было.
При возникновении ошибки отсутствия страницы ядро операционной системы просматривает все страницы и на основе текущих значений
битов M, R делит их на четыре класса:
* Класс 0: в последнее время не было ни обращений, ни модификаций.
* Класс 1: обращений в последне время не было, но страница модифицирована.
* Класс 2: в последнее время были обращения, но модификаций не было.
* Класс 3: были и обращения, и модификации в последнее время.

Страницы класса 1 появляются в том случае, если у страниц класса 3 бит R сбрасывается по прерыванию от таймера.
Эти прерывания не сбрасывают бит M, поскольку содержащаяся в нём информация необходима для того, чтобы узнать,
нужно ли перезаписывать страницу на диске. Сброс бита R без бита M приводит к возникновению страниц класса 1.
Алгоритм NRU удаляет произвольную страницу, относящуюся к самому низкому непустому классу.
В основе алгоритма заложена идея, что лучше удалить модифицированную страницу, к которой не было обращений по крайней мере
за последний такт системных часов, чем удалить интенсивно используемую страницу.
Алгоритм NRU позволяет достичь приемлемой производительности.
2) FIFO. Операционная система ведёт список всех физических страниц. Недавно загруженные с диска страницы поступают в конец списка.
При возникновении исключения отсутствия страницы удаляется страница, находящаяся в голове списка.
3) Алгоритм "Второй Шанс". Модификация алгоритма FIFO, учитывающая бит R. У страницы, находящейся в голове списка, проверяется бит R.
Если он равен 1, то бит сбрасывается, а сама страница помещается в конец списка, но не удаляется. Если он равен 0, то действия аналогичны алгоритму FIFO.
Если страница не удалена, то проверяется следующая по очередности страница. С ней происходят те же самые проверки и операции.
Вырожденный случай -- все страницы были использованы. В таком случае по алгоритму у каждой страницы будет сброшен бит R, а суммарный эффект будет аналогичен алгоритму FIFO.

Эффективная реализация данного алгоритма -- манипулирования одними указателями без физического перемещения дескрипторов страниц.
4) LRU (Least Recently Used, исключение наименее востребованной страницы). Для каждой страницы ведётся специальнй счётчик.
При возникновении прерывания от таймера операционная система сканирует все страницы и добавляет к счётчику каждой страницы текущее значение R.
После этого значение R также сбрасывается. При возникновении ошибки отсутствия страницы, удаляется страница, имеющая наименьшее значение счётчика.
5) "Рабочий набор".
Данный метод основывается на том наблюдении, что большинство программ обращается к адресному пространству неравномерно.
Набор страниц, используемых процессом в настоящий момент, называется __рабочим_набором__.
На практике приближённо определить рабочий набор можно по критерию: рабочий набор - страницы, к которым были обращения за последние N мс
(время виртуальное, равное интервалу времени, в течение которого задача исполняется на CPU).
При возникновении ошибки отсутствия страницы выбирается страница, не принадлежащая рабочему набору.

##
###

52. Управление устройствами ввода/вывода: система прерываний, системы драйверов внешних устройств.

CPU взаимодействует с периферийным оборудованием, выставляя на адресных линиях системной шины специальные адреса, воспринимаемые контроллером внешнего устройства.
Данный метод взаимодействия называется отображенным в память вводом/выводом (MMIO, Memory Mapped Input/Output).
Таким образом, используя привычные инструкции пересылки из памяти/в памяти MOV, LW/SW, можно считывать различные данные от устройства,
включая его состояние.

Методы ввода/вывода:
1) Активное ожидание/Опрос. Перед каждой пересылкой данных к контроллеру внешнего устройство операционная система считывает со специального регистра устройства его состояние, которое можно охарактеризовать как готово/не готово.
На псевдокоде данный метод можно описать как:

volatile u32 *data_register = (volatile u32 *) <Адрес регистра, куда записываются данные>;
volatile u32 *status_register = (volatile u32 *) <Адрес регистра, откуда можно считать состояние устройства>;
while (*status_register != READY);
*data_register = <Записываемые данные>;

2) Ввод/вывод, управляемый прерываниями. Вместо того чтобы ожидать устройство на готовность, контроллер данного устройства программируется операционной системой таким образом, чтобы инициировать прерывание CPU по завершении операции -- готовности устройства. Таким образом, циклы CPU не расходуются впустую. При

наступлении прерывания CPU останавливает выполнение текущей задачи и переходит к выполнению специальной процедуры, называемой обработчиком прерывания.

На псевдокоде данный метод можно описать как:

Начальный запрос устройства:

```
volatile u32 *data_register   = (volatile u32 *) <Адрес регистра, куда записываются данные>;
volatile u32 *status_register = (volatile u32 *) <Адрес регистра, откуда можно считать состояние устройства>;
volatile u32 *interrupt_register = (volatile u32 *) <Адрес регистра, управляемого генерацией устройством прерываний CPU>;
*interrupt_register = ENABLE_INTERRUPTS;    /* Включаем прерывания на контроллере внешнего устройства */
enable_cpu_interrupts();                    /* Включаем прерывания на CPU */
while (*status_register != READY);
*data_register = <Записываемые данные>;
```

Обработчик прерывания:

```
volatile u32 *data_register   = (volatile u32 *) <Адрес регистра, куда записываются данные>;
acknowledge_interrupt();                    /* Посылка контроллеру прерывания информации о завершении обработки прерывания */
return_from_interrupt();
```

3) Ввод/вывод с использованием DMA.

CPU конфигурирует контроллер DMA, либо принадлежащий самому устройству, либо выступающий в качестве самостоятельного устройства, таким образом, чтобы:
 * был осуществлен трансфер заданного буфера оперативной памяти в устройство;
 * по окончании трансфера было инициировано прерывание CPU.

На псевдокоде данный метод выглядит следующим образом:

Начальный запрос устройства:

```
set_up_DMA_controller();
```

Обработчик прерывания:

```
acknowledge_interrupt();
return_from_interrupt();
```

Драйвера внешних устройств.

__Драйвер__ - программа, управляющая работой внешнего устройство через взаимодействие с его контроллером.
В своей работе драйвер использует описанные выше методы ввода/вывода, а также MMIO.

Драйвера входят в состав ядра операционной системы и имеют с ним чётко обозначенный интерфейс, зависящий от подсистемы ядра, в которую входит драйвер (сетевая подсистема, подсистема PCI и т. п.).

Любой драйвер имеет в своем составе функцию инициализации, в которой:
 * происходит проверка присутствия нижележающего оборудования с помощью запросов через MMIO;
 * происходит идентификация оборудования (через чтение специализированных регистров устройства);
 * в случае если оборудование присутствует и поддерживается драйвером, происходит обращение к ядру операционной системы
 для установки обработчика прерывания от устройства.
 * драйвер регистрирует себя в подсистеме ядра; таким образом, он становится доступен пользовательскому пространству.

##
###

53. Управление файловыми системами: организация дискового пространства, современные файловые системы.

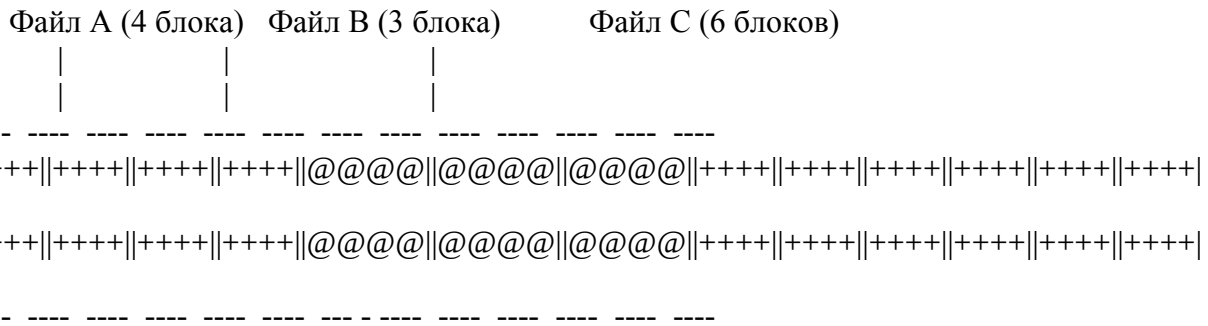
__Файловая_система__ - способ хранения файлов на долговременном носителе.
__Файл__ - именованная последовательность байт, хранящаяся на долговременном носителе.

Организация дискового пространства.

Цель организации дискового пространства - эффективное отслеживание соответствие файлов блоков на диске.

Расположение файлов:

1) Непрерывное размещение. Файлы располагаются на долговременном носителе информации в виде непрерывной последовательности блоков.
Таким образом, при размере блока 512 байт, файл размером 10 Кб займет на диске 20 последовательных блоков.



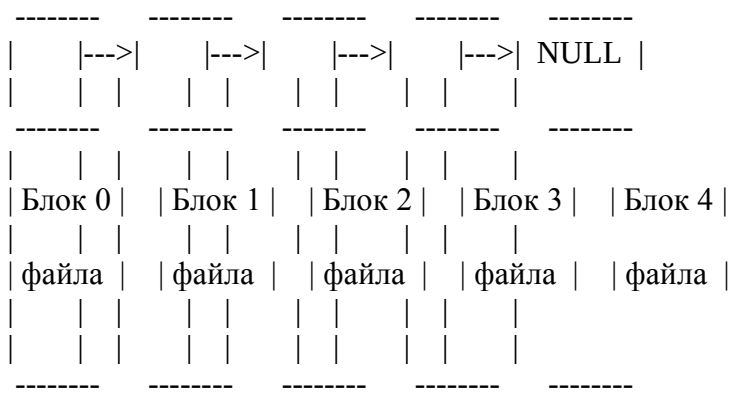
Пример организации хранилища показан на диаграмме выше. Здесь следует заметить, что каждый следующий файл записывается сразу после предыдущего.

Следует заметить также, что каждый новый файл начинается от границы нового блока. Поэтому, если фактически файл А занимает 3,5 блоков, то пространство будет потеряно впустую (фрагментация).

При удалении блоки, занимаемые файлов, помечаются как свободные. Перемещение файлов не производится, поскольку может занимать значительное время. Фрагментация неизбежна.

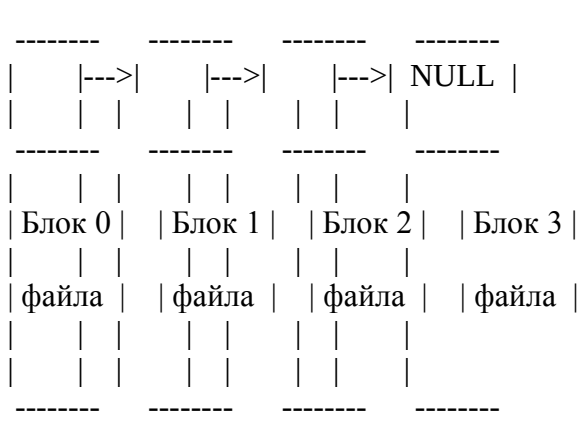
2) Размещение с использованием связанного списка. Представление каждого файла в виде связанного списка дисковых блоков.
При этом первое слово каждого блока используется в качестве указателя на следующий дисковый блок, а вся остальная часть блока предназначена для хранения данных.

ФАЙЛ А.



Физический блок 4 7 2 10 12

ФАЙЛ В.



Физический блок 6 3 11 14

В отличие от непрерывного хранения файлов, в этом методе может быть использован каждый дисковый блок.

При этом потери дискового пространства на фрагментацию отсутствуют (за исключением внутренней фрагментации в последнем блоке).
Кроме того, достаточно, чтобы в записи каталога хранился только дисковый адрес первого блока. Всю остальную информацию можно найти начиная с этого блока. В то же время доступ к файлам, хранящимся подобным образом осуществляется медленее, чем к последовательно хранимым файлам.

3) Размещение с использованием связанного списка, использующего таблицу в памяти (FAT, File Allocation Table).
Развитие второго метода, при котором указатели хранятся на долговременном носителе информации в виде массива, где индекс элемента равен номеру физического блока. При инициализации файловой системы данная таблица загружается в оперативную память, за счёт чего можно быстро вычислить требуемый физический блок.

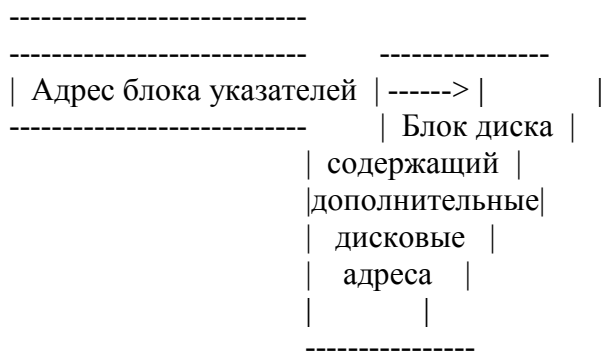
Физический блок		
0		
1		
2	10	
3	11	
4	7	<---- ФАЙЛ А НАЧИНАЕТСЯ ЗДЕСЬ.
5		
6	3	<---- ФАЙЛ В НАЧИНАЕТСЯ ЗДЕСЬ.
7	2	
8		
9		
10	12	
11	14	

```
  ----------
12 |  -1   |
  ----------
  ----------
13 |       |
  ----------
  ----------
14 |  -1   |
  ----------
  ----------
15 |       | <---- НЕИСПОЛЬЗУЕМЫЙ БЛОК.
  ----------
```

Существенным недостатком данного метода является прямая пропорциональность дополнительно требуемой памяти объему памяти долговременного носителя информации. При больших объемах памяти долговременного носителя информации данный метод неприменим.

3) I-узлы (inodes). С каждым файлов связывается i-узел. При этом i-узлы всех файлов не обязательно должны хранится в оперативной памяти одновременно.
I-узел (inode) - специальная структура данных, содержащая атрибуты файлов, а также дисковые адреса его блоков.
Схема i-узла:

```
  ---------------------------
  |                         |
  |     Атрибуты файла      |
  |                         |
  ---------------------------
  ---------------------------
  |     Адрес блока 0       | ------>
  ---------------------------
  ---------------------------
  |     Адрес блока 1       | ------>
  ---------------------------
  ---------------------------
  |     Адрес блока 2       | ------>
  ---------------------------
  ---------------------------
  |     Адрес блока 3       | ------>
  ---------------------------
  ---------------------------
  |     Адрес блока 4       | ------>
  ---------------------------
  ---------------------------
  |     Адрес блока 5       | ------>
  ---------------------------
  ---------------------------
  |     Адрес блока 6       | ------>
  ---------------------------
  ---------------------------
  |     Адрес блока 7       | ------>
```



Многие файловые системы UNIX, а также NTFS используют именно этот способ хранения информации на долговременном носителе.

##
###

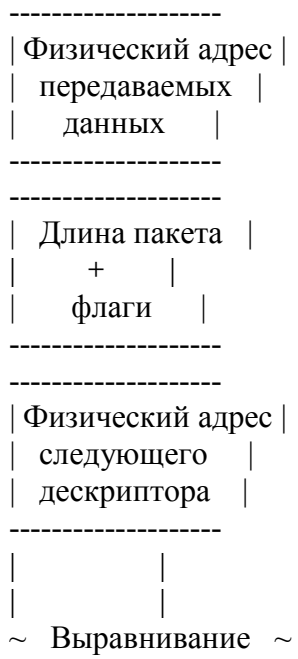
54. Сетевые возможности современных операционных систем:

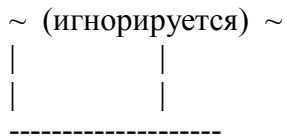
архитектура сетевых операционных систем, реализация операционных систем для различных типов компьютерных сетей, сетевые службы.

__Сетевая_операционная_система__ - операционная система, поддерживающая работу по компьютерной сети.
Примеры сетевых ОС: Linux, Windows.

Архитектура сетевой операционной системы.

0) На самом нижним уровне выступают драйвера сетевых карт.
Драйвера коммуницируют с сетевой картой посредством специальных структур данных - дескриптора.
Схема типового дескриптора изображена ниже:





Каждый дескриптор содержит указатель на следующий.
Совокупность дескрипторов образует кольцевой односвязный список.
Именно через дескрипторы происходит приём/отправка пакетов по физическому носителю (витая пара, радиоэфир и т.п.)

1) Обработка данных на канальном уровне. Реализуется совместными усилиями драйвером сетевого устройства и интерфейса ядра между сетевым стеком и драйвером сетевого устройства. На данном уровне проверяется, что mac-адрес узла-получателя совпадает с mac-адресом сетевой платы, на которую пришел пакет.
Также определяется протокол верхнего уровня (сетевого): arp, ipv4, ipv6 и т. п.
Организуется передача полученного пакета модулю ядра, реализующего протокол сетевого уровня.
При отправке происходит заполнение заголовка канального уровня и передача его драйверу сетевого устройства.

2) Обработка данных на сетевом уровне. Реализуется модулем ядра - реализацией сетевого протокола.
На данном уровне происходит анализ сетевого адреса назначения, а также определения дальнейшего пути пакета:
 * input -- пакет предназначен для текущего хоста на сетевом уровне.
 * output -- пакет сгенерирован текущим хостом и должен быть передан драйверу сетевого устройства.
 * forward -- пакет предназначен другому хосту, путь до которого содержится в таблицах маршрутизации текущего хоста.
В соответствии с определённым назначением, пакет либо передается протоколу верхнего уровня (tcp/udp, icmp),
либо передается драйверу сетевого устройства для отправки (output, forward).

3) Обработка данных на транспортном уровне. Реализуется модулем ядра - реализацией транспортного протокола.
На данном уровне происходит контроль целостности передаваемых/получаемых данных (за счёт подтверждения полученных пакетов - для tcp),
ликвидация дубликатов, а также контроль порядка (за счёт нумерации отправляемых пакетов в заголовке транспортного уровня - tcp).
При получении пакета ищется задача, его ожидающая (задача, которой принадлежит сокет).
Если задача найдена, то происходит переключение задачи в состояние готовности, а также передача ей полученных данных.

4) Интерфейс сокетов. Является интерфейсом пользовательского пространства к сетевым возможностям операционной системы. Реализуется через механизм системных вызовов.

Интерфейс традиционно включает в себя следующие примитивы:

 * socket -- создает в пространстве ядра сокет, принадлежащий определённому протоколу (address family), возвращает целое число, идентифицирующее созданный сокет (в Linux - файловый дескриптор);

* bind -- связывает сокет с сетевым адресом; адрес может включать в себя как сетевой адрес (IP-адрес), так и транспортный адрес (номер TCP/UDP порта);
* listen -- для протоколов, ориентированных на соединение (TCP), разрешает приём входящих соединений к данному сокету;
* accept -- ожидает подключения (tcp), после чего возвращает файловый дескриптор, с помощью которого можно взаимодействовать с подключившейся стороной;
* connect -- инициирует соединение к хосту с заданным адресом (для tcp);
* send -- отправляет данные по сети;
* recv -- получает данные по сети;
* close -- закрывает сокет, разрывая установленные соединения и завершая текущие операции.

Сетевые службы.

Программы пользовательского окружения, использующие сетевые возможности операционной системы (интерфейс сокетов), называются сетевыми службами. Примеры сетевых служб:
* DHCP клиент;
* WEB-сервер;
* Браузер;
* NTPD -- демон, реализующий протокол ntp (синхронизация часов).

##
###

55. БД и СУБД. Основные функции СУБД. Многоуровневая архитектура современных СУБД.

База Данных - организованная совокупность взаимосвязанных данных, предназначенных для многократного использования приложениями/пользователями.
Система Управления Базами Данных - совокупность программного обеспечения, необходимого для ведения, использования и поддержания баз данных в актуальном состоянии.

@Основные функции СУБД:
1) Администрирование баз данных

СУБД имеют развитые средства администрирования базы данных, например, для определения доступа к базе, ее архивации и защите хранимой информации. В связи с тем, что базы данных проникают сегодня во многие сферы деятельности человека, появилась новая профессия – администратор базы данных, человек, отвечающий за проектирование, создание, использование и сопровождение базы данных. В процессе эксплуатации БД администратор обычно следит за ее функционированием, обеспечивает защиту от несанкционированного доступа к хранимым данным, вносит изменения в структуру базы, контролирует достоверность информации в ней.
2) Непосредственное управление данными во внешней памяти

Эта функция предоставляет пользователю возможность выполнения основных операций с данными – хранение, извлечение и обновление информации. Она включает в себя обеспечение необходимых структур внешней памяти как для хранения данных, непосредственно входящих в БД, так и для служебных целей, например, для ускорения доступа к данным. СУБД поддерживает собственную систему именования объектов БД.
3) Управление буферами оперативной памяти

СУБД обычно работают с БД значительного размера; очень часто этот размер существенно больше доступного объема оперативной памяти. Так как при обращении к любому элементу данных будет производиться обмен с внешней памятью, то вся система будет работать со скоростью устройства внешней памяти. Практически единственным способом реального увеличения этой скорости является буферизация данных в оперативной памяти. Однако этого недостаточно для целей СУБД, поэтому в развитых СУБД поддерживается собственный набор буферов оперативной памяти.

4) Управление транзакциями

Транзакция – это последовательность операций над БД, которые рассматриваются СУБД как единое целое и позволяют добавлять, удалять или обновлять сведения о некотором объекте в базе (по существу это некоторый программный код, написанный на одном из языков управления данными). Либо транзакция успешно выполняется, и СУБД фиксирует изменения БД, произведенные этой транзакцией, либо ни одно из этих изменений никак не отражается на состоянии БД. Например, если в результате транзакции произошел сбой компьютера, база данных попадает в противоречивое положение – некоторые изменения уже внесены, остальные нет. Транзакция позволяет вернуть базу в первоначальное непротиворечивое состояние (отменить все выполненные изменения).

Свойства транзакций:

- свойство атомарности (Atomicity) выражается в том, что транзакция должна быть выполнена в целом или не выполнена вовсе;
- свойство согласованности (Consistency) гарантирует, что по мере выполнения транзакций данные переходят из одного согласованного состояния в другое — транзакция не разрушает взаимной согласованности данных;
- свойство изолированности (Isolation) означает, что конкурирующие за доступ к базе данных транзакции физически обрабатываются последовательно, изолированно друг от друга, но для пользователей это выглядит так, как будто они выполняются параллельно;
- свойство долговечности (Durability) трактуется следующим образом: если транзакция завершена успешно, то те изменения в данных, которые были ею произведены, не могут быть потеряны ни при каких обстоятельствах (даже в случае последующих ошибок).

5) Журнализация

Одним из основных требований к СУБД является надежность хранения данных во внешней памяти. Под надежностью хранения понимается то, что СУБД должна быть в состоянии восстановить последнее состояние БД после любого аппаратного или программного сбоя (аварийное выключение питания, аварийное завершение работы СУБД или аварийное завершение пользовательской программы).

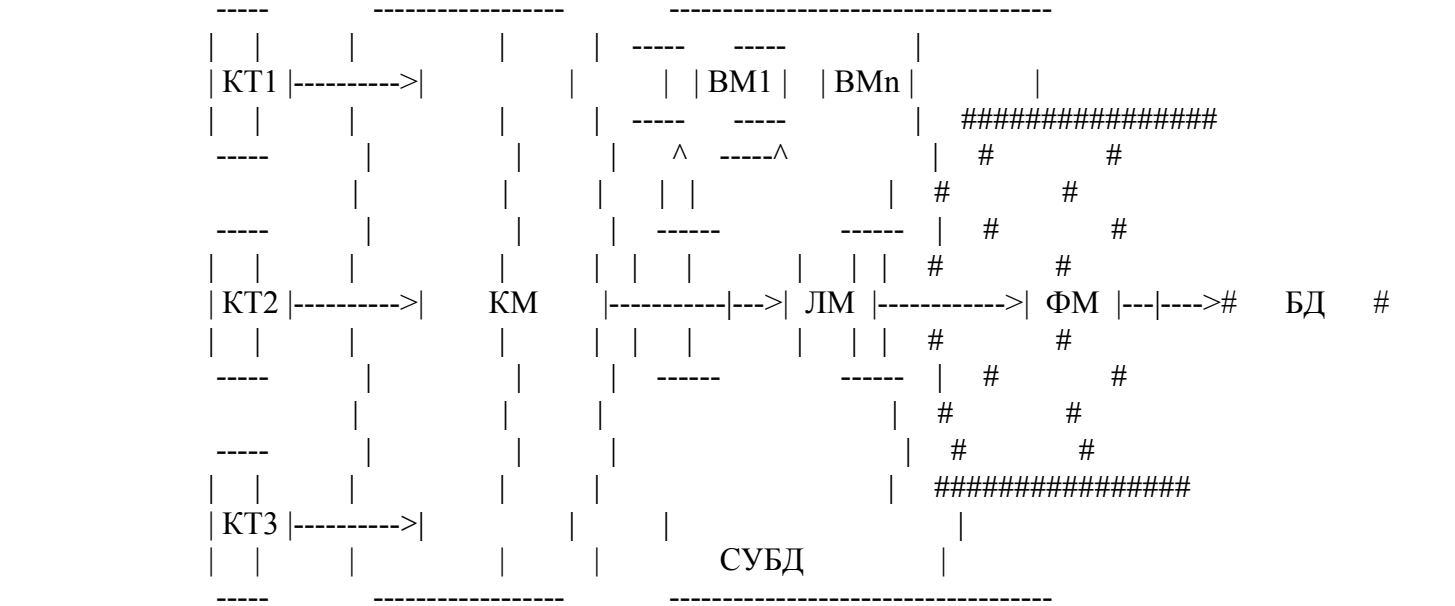
Наиболее распространенным методом поддержания надежности хранения является ведение журнала изменений БД. Журнал – это особая часть БД, недоступная пользователям и поддерживаемая с особой тщательностью (иногда поддерживаются две копии журнала, располагаемые на разных физических дисках), в которую поступают записи обо всех изменениях основной части БД. Изменения БД журнализуются следующим образом: запись в журнале соответствует некоторой операции изменения БД (например, операции удаления строки из таблицы реляционной БД). С помощью журнала можно решить все проблемы восстановления БД после любого сбоя.

6) Поддержка языков БД

СУБД включает язык определения данных, с помощью которого можно определить структуру базы, тип данных в ней, указать ограничения целостности (это язык, с помощью которого задаются различные имена, свойства объектов). Кроме того, СУБД позволяет вставлять, удалять, обновлять и извлекать информацию из базы данных посредством языка управления данными – языка запросов, который позволяет выполнять различные действия с данными, осуществлять их поиск и выборку. Он содержит набор различных

операторов (заносить данные, удалять, модифицировать, выбирать и т.д.). Процесс извлечения данных и их обработка скрыты от пользователя. Стандартным языком наиболее распространенных в настоящее время СУБД является язык SQL (Structured Query Language). Он имеет сразу два компонента: язык определения данных и язык управления данными.

@Многоуровневая архитектура современных СУБД:
СУБД имеет многоуровневую структуру, в которой реализуется принцип относительной независимости логической и физической организации данных.



КТi – это концептуальные требования на уровне понятий, их структур и взаимосвязей к данным, хранимым в БД, которые собираются и анализируются на начальном этапе проектирования БД;
КМ – концептуальная модель, которая представляет собой интегрированное представление КТi требований в терминах предметной области. В процессе построения КМ устраняются несоответствия, ошибки в КТi. КМ является средством аналогом базы знаний различных субъектов проекта об одной и той же предметной области. В значительной мере способствует успеху проектирования и сокращению сроков реализации проекта;
ЛМ – логическая модель, т.е. концептуальная модель предметной области, записанная в терминах конкретной области;
ФМ – физическая модель, т.е. описание способов хранения данных на физическом уровне, методов доступа к ним, способов индексации;
ВМi – внешние модели i-го пользователя/приложения, которые соответствуют его начальным КТi требованиям, но предоставляются в рамках конкретной СУБД;
Если на логическом уровне хранятся нормализованные данные, где информация об одном информационном объекте обычно хранится не в одной, а в нескольких взаимосвязных таблицах, то любая ВМi по сути является ненормализованным представлением, которое является не просто выпиской из ЛМ, а содержит нужные преобразования над данными (виртуальные атрибуты).

##
###

56. Понятие модели данных (МД). Основные компоненты МД. Традиционные МД. Отличительные особенности семантических МД.

Модель данных - некоторая формальная теория представления и обработки данных, включающая методы описания типов и логических структур данных (аспект структуры), методы манипулирования данными (аспект манипуляции) и методы описания и поддержки целостности (аспект целостности).
Модель базы данных — тип модели данных, которая определяет логическую структуру базы данных и принципиально определяет, каким образом данные могут быть сохранены, организованы и обработаны.

@Основные компоненты модели данных:
1) Допустимая организация данных
2) Ограничения целостности
3) Множество допустимых операций

@Традиционные модели данных:
1) Иерархическая модель данных -
2) Сетевая модель данных | https://ru.bmstu.wiki/Модели_баз_данных
3) Объекто-ориентированная модель данных -
4) Реляционная модель данных

Концепции реляционной модели впервые были сформулированы в работах американского ученого Э. Ф. Кодда, откуда происходит ее второе название - модель Кодда. В реляционной модели объекты и взаимосвязи между ними представляются с помощью таблиц. В моделях реляционных баз данных широко используются три ключевых термина: отношения, атрибуты и домены. Отношение - это таблица со столбцами и строками. Именованные столбцы отношения называются атрибутами, а домен - это набор значений, которые могут принимать атрибуты. Строки таблицы со значениями разных атрибутов называют кортежами. Атрибут, значение которого однозначно идентифицирует кортежи, называется ключевым (или просто ключом). Так ключевое поле – это такое поле, значения которого в данной таблице не повторяется. Для отражения ассоциаций между кортежами разных отношений используется дублирование их ключей. Сложный ключ выбирается в тех случаях, когда ни одно поле таблицы однозначно не определяет запись. Записи в таблице хранятся упорядоченными по ключу. Ключ может быть простым, состоящим из одного поля, и сложным, состоящим из нескольких полей. Сложный ключ выбирается в тех случаях, когда ни одно поле таблицы однозначно не определяет запись. Кроме первичного ключа в таблице могут быть вторичные ключи, называемые еще внешними ключами, или индексами. Индекс – это поле или совокупность полей, чьи значения имеются в нескольких таблицах и которое является первичным ключом в одной из них. Значения индекса могут повторяться в некоторой таблице. Индекс обеспечивает логическую последовательность записей в таблице, а также прямой доступ к записи. Важным преимуществом реляционной модели является то, что в ее рамках действия над данными могут быть сведены к операциям реляционной алгебры, которые выполняются над отношениями. Это такие операции, как: объединение, пересечение, вычитание, декартово произведение, выборка, проекция, соединение, деление. Важнейшей проблемой, решаемой при проектировании баз данных, является создание такой их структуры, которая бы обеспечивала минимальное дублирование информации и упрощала процедуры обработки и обновления данных. Коддом был предложен некоторый набор формальных требований универсального характера к организации данных, которые позволяют эффективно решать перечисленные задачи. Эти требования к состоянию

таблиц данных получили название нормальных форм. Первоначально были сформулированы три нормальные формы:

- говорят, что отношение находится в первой нормальной форме, если все его атрибуты являются простыми;
- говорят, что отношение находится во второй нормальной форме, если оно удовлетворяет требованиям первой нормальной формы и каждый не ключевой атрибут функционально полно зависит от ключа (однозначно определяется им);
- говорят, что отношение находится в третьей нормальной форме, если оно удовлетворяет требованиям второй нормальной формы и при этом любой не ключевой атрибут зависит от ключа нетранзитивно. Транзитивной называется такая зависимость, при которой какой-либо не ключевой атрибут зависит от другого не ключевого атрибута, а тот, в свою очередь, уже зависит от ключа.

В дальнейшем появилась нормальная форма Бойса-Кодда и нормальные формы более высоких порядков. Однако они не получили широкого распространения на практике. Принципиальным моментом является то, что для приведения таблиц к состоянию, удовлетворяющему требованиям нормальных форм, или, как еще говорят, для нормализации данных над ними, должны быть осуществлены перечисленные выше операции реляционной алгебры.

Основным достоинством реляционной модели является ее простота. Именно благодаря ей она положена в основу подавляющего большинства реально работающих СУБД.

В разработанной Коддом реляционной модели были определены как требования к организации таблиц, содержащих данные, так и язык, позволяющий работать с ними. Впоследствии этот язык получил название SQL (Structured Query Language - структурированный язык запросов). Очень скоро SQL стал стандартом de facto языка работы с реляционными базами данных. В составе SQL могут быть выделены следующие группы инструкций:

- язык описания данных (DDL, Data Definition Language): CREATE, DROP, ALTER;
- язык манипулирования данными (DML, Data Manipulation Language): SELECT, INSERT, UPDATE, DELETE;
- язык управления транзакциями: COMMIT, ROLLBACK, SAVEPOINT.

5) Нереляционная модель данных

Нереляционные СУБД – относительно недавнее пополнение множества систем для работы с данными. Их появление и растущая популярность вызваны, главным образом, развитием сетевых технологий и приложений. Современные сетевые приложения, наиболее яркими примерами которых являются социальные сети, должны поддерживать доступ одновременно для миллионов пользователей и хранить терабайты различных данных. К сожалению, при всех достоинствах, реляционные СУБД не могут обеспечить работу в таком режиме. Именно эта ситуация и вызвала значительный рост количества решений, позволяющих обеспечить эффективное хранение и обработку данных для высоконагруженных сетевых приложений.

@Отличительные особенности семантических моделей данных:

"Семантическое моделирование стало предметом интенсивных исследований с конца 1970-х годов. Основным побудительным мотивом подобных исследований (то есть проблемой, которую пытались разрешить исследователи) был следующий факт. Дело в том, что системы баз данных обычно обладают весьма ограниченными сведениями о смысле хранящихся в них данных. Чаще всего они позволяют лишь манипулировать данными определенных простых типов и определяют некоторые простейшие ограничения целостности, наложенные на эти данные. Любая более сложная интерпретация возлагается на пользователя. Однако было бы замечательно, если бы системы могли обладать немного более широким объемом сведений и несколько интеллектуальнее отвечать на запросы

пользователя, а также поддерживать более сложные (то есть более высокоуровневые) интерфейсы пользователя..."
Семантическое моделирование представляет собой моделирование структуры данных, опираясь на смысл этих данных. В качестве инструмента семантического моделирования используются различные варианты диаграмм сущность-связь (ER - Entity-Relationship).
В рамках семантического моделирования используются четыре вида различных элементов:
1) Сущности — некоторые различимые объекты, например, факультеты, кафедры, группы и студенты. Сущности бывают обычными и слабыми. Слабой является сущность, которая не может существовать, если не существует некоторая другая сущность.
2) Свойства — некоторая информация, описывающая сущность, например, номер группы или фамилия студента. Свойства могут простыми или составными, однозначными или многозначными, базовыми или производными. Также обособлено выделяют ключевые и отсутствующие свойства.
3) Связи — сущности, которые служат для обеспечения взаимодействия между двумя или несколькими другими сущностями. Количество сущностей, включенных в связь, определяет степень связи. При этом возможны следующие виды связей: «один к одному», «один ко многим» или «многие к одному», «многие ко многим».
4) Подтипом одной сущности является другая сущность, каждый экземпляр которой является экземпляром первой сущности.

Особенности (из старого документа):
- семантические модели отличаются большей выразительной мощностью, но, как правило, меньшим быстродействием. В частности, выразительная мощность отражается в гибкости структурных средств. Как правило семантические модели - это графовые модели, позволяющие отразить сколь угодно сложные предметные области;
- использование высокоуровневых абстракций. В семантических моделях, как правило, используются различного рода парадигматические отношения (класс - подкласс, часть - целое и т.д.);
- как правило хранятся не только данные, но и знания, данные о данных(методанные), знания о знаниях(метознания);
- богатый набор явных ограничений целостности;
- существует возможность накладывать ограничения целостности не только на данные, но и на операции. И те и другие ограничения целостности накладываются в статике, а контролируются в динамике. Ограничения целостности на данные накладываются на какой-то конкретный набор данных и контролируются после каждой операцией над этими данными. Ограничения целостности на операции накладываются на конкретные операции и контролируются при проведении этих операций над любыми данными. Подобного рода подход позволяет расширять семантику стандартных операций.

##
###

57. Администрирование современных СУБД. Обеспечения безопасности данных в современных СУБД на примере СУБД Oracle. Технологии удаленного доступа к системам баз данных, тиражирование и синхронизация в распределенных системах баз данных.

@Основные компоненты системы защиты баз данных:

1) Разграничение доступа - каждый пользователь, включая администратора, имеет доступ только к необходимой ему согласно занимаемой должности информации.
2) Защита доступа - доступ к данным может получить пользователь, прошедший процедуру идентификации и аутентификации.
3) Шифрование данных - шифровать необходимо как передаваемые в сети данные для защиты от перехвата, так и данные, записываемые на носитель, для защиты от кражи носителя и несанкционированного просмотра/изменения не-средствами системы управления БД (СУБД).
4) Аудит доступа к данным - действия с критичными данными должны протоколироваться. Доступ к протоколу не должны иметь пользователи, на которых он ведется.

@В подробностях:
1) Разграничение доступа
Для обеспечения разграничения доступа в версии СУБД 10g компания Oracle выпустила новый продукт Database Vault, предназначенный для предотвращения несанкционированного доступа к информации пользователей, в том числе наделенных особыми полномочиями, например, администраторов базы данных. Набор правил в Database Vault, разграничивающих доступ, достаточно широк. Например, руководство организации может определить правила, согласно которым для решения задач, предполагающих доступ к критичной информации, потребуется одновременное присутствие двух сотрудников. Таким образом, Database Vault решает следующие проблемы:
- ограничение доступа к данным администратора БД и других привилегированных пользователей;
- предотвращение манипулирования с базой данных и обращения к другим приложениям администратора приложений;
- обеспечение контроля над тем, кто, когда и откуда может получить доступ к приложению.

2) Защита доступа
Аутентификация в контексте Oracle означает проверку подлинности кого-либо или чего-либо - пользователя, приложения, устройства, кому или чему требуется доступ к данным, ресурсам или приложениям. После успешной процедуры аутентификации следует процесс авторизации, предполагающий назначение определенных прав, ролей и привилегий для субъекта аутентификации.
Oracle предоставляет разнообразные способы аутентификации и позволяет применять один или несколько из них одновременно. Общим для всех этих способов является то, что качестве субъекта аутентификации используется имя пользователя. Для подтверждения его подлинности может запрашиваться некоторая дополнительная информация, например, пароль. Программное обеспечение Oracle также зашифровывает пароли пользователей для безопасной передачи по сети.

Виды аутентификации:
а) Аутентификация средствами операционной системы
Ряд операционных систем позволяют СУБД Oracle использовать информацию о пользователях, которыми управляет ОС. В этом случае пользователь компьютера имеет доступ к ресурсам БД без дополнительного указания имени и пароля - используются его сетевые учетные данные. Данный вид аутентификации считается небезопасным и используется, в основном, для аутентификации администратора СУБД.
б) Аутентификация при помощи сетевых сервисов

Данный вид аутентификации обеспечивает опция сервера Oracle Advanced Security. Она предоставляет следующие службы:

- аутентификация с использованием протокола SSL
- аутентификация службами третьих сторон: Kerberos, PKI, RADIUS (Remote Authentication Dial - In User Service), LDAP

в) Аутентификация в многоуровневых приложениях

Приведенные выше методы аутентификации также могут быть применены и в многоуровневых приложениях. Как правило, для доступа к приложениям из сети Интернет используется аутентификация по имени и паролю (в том числе с использованием протокола RADIUS), либо по протоколу SSL. Прочие методы используются для работы пользователей в локальной сети.

3) Шифрование данных

Для защиты данных, передаваемых в сети, в СУБД Oracle, начиная с версии 8i, используется возможности опции Oracle Advanced Security, в которой предусмотрена функция Network encryption, позволяющая шифровать весь поток данных. Безопасность информации обеспечивается секретностью ключа, которым шифруются данные. Поддерживаются следующие алгоритмы шифрования: AES(только 10g /11g), DES, 3DES, RC 4(только 10g /11g).

Защита передаваемых в сети данных в приложениях Oracle обеспечивается протоколом SSL по алгоритмам, которые поддерживается сервером приложений, как правило, это WEB-сервер Oracle.

Защиту данных на носителе обеспечивают два компонента СУБД Oracle - пакеты, реализующие алгоритмы шифрования и опция Transparent Data Encryption (TDE). Управление ключами шифрования берет на себя ядро БД, а применение такого шифрования не требует переделки клиентского и серверного прикладного ПО.

4) Аудит доступа к данным

СУБД Oracle имеет мощные средства аудита действий пользователей, включающих как доступ к данным, так и события регистрации/выхода и изменения структуры БД. Начиная с версии 9i, СУБД оснащается опцией подробного аудита (Fine Grained Audit Control), которая позволяет проводить аудит доступа по условиям с достаточно гибкими настраиваемыми правилами. Однако, данные средства аудита не позволяет проследить за действиями, которые совершаются администратором базы данных, а также не мешают ему изменять журнал аудита, удаляя любые строки и не оставляя следов подобных действий. Возникшая необходимость аудита деятельности и защиты данных аудита от привилегированных пользователей, включая администраторов БД, побудило Oracle разработать новую концепцию аудита. В её основу положена идея, на которой базируется функционал Database Vault: администратор БД изолирован от управления аудитом, что по понятным причинам обеспечивает более высокий уровень безопасности БД.

@Модель удаленного доступа к данным:

В данной модели компонент доступа к данным реализуется в виде самостоятельной программной части СУБД, называемой SQL-сервером, и размещается на сервере. SQL-сервер выполняет низкоуровневые операции по организации, размещению, хранению и манипулированию данными. На сервере размещаются также файлы БД и системный каталог БД. На клиентских установках размещаются программы, реализующие интерфейсные и прикладные функции СУБД. Прикладной компонент клиента формирует необходимые SQL-инструкции и направляет их SQL-серверу, который принимает, интерпретирует, выполняет, проверяет эти инструкции, обеспечивает выполнение ограничений целостности и безопасности данных и направляет клиентам результаты обработки SQL-инструкций (наборы данных).

Достоинства: в результате реализации такого подхода резко уменьшается загрузка сети. Модель позволяет также унифицировать интерфейс взаимодействия прикладных компонентов СУБД с общими данными. Такое взаимодействие стандартизовано в рамках языка SQL специальным протоколом ODBC, играющим важную роль в обеспечении независимости от типа СУБД на клиентских установках. Это позволяет интегрировать уже существующие локальные БД в создаваемые распределенные информационные системы независимо от типов СУБД клиентов и сервера.
Недостатки: высокие требования к клиентским вычислительным установкам, так как на них выполняются прикладные программы обработки данных. Значительный трафик сети, поскольку с сервера направляются клиентам наборы данных, которые могут иметь существенный объем.

@Технологии тиражирования (реплицирования) данных:
Основная идея тиражирования: пользователи работают автономно с одинаковыми (общими) данными, растиражированными по локальным базам данных, что обеспечивает в силу отсутствия необходимости передачи растиражированных данных максимальную производительность используемой вычислительной системы.
Реплика – тиражируемая копия данных, предназначенных для общего пользования.

При реализации технологии тиражирования данных возникает проблема обеспечения согласованного состояния данных, т.е. согласованного состояния во всех репликах количества и значений общих данных, а также структуры данных. Решение проблемы обеспечения согласованного состояния количества и значений общих данных основывается на реализации одного из двух принципов:
- принципа непрерывного размножения обновлений (любое обновление данных в любой реплике должно быть немедленно размножено). Данный принцип реализуется при построении систем реального времени. Реализация этого принципа заключается в том, что любая транзакция считается успешно завершенной, если она успешно завершена во всех репликах. На практике реализации данного принципа препятствует возникновение тупиков. Для обнаружения и распознавания тупиков в реплицированных системах применяются те же алгоритмы, что и в мониторах транзакций централизованных систем типа «клиент-сервер».
- принципа отложенных обновлений (обновления реплик могут откладываться до специальной команды или ситуации). Накопленные в реплике изменения данных специальной командой пользователя направляются для обновления всех остальных реплик системы. Такая операция называется синхронизацией реплик. В данном случае существенно снижается возможность конфликтов и тупиков. Для реализации процесса синхронизации реплик в системном каталоге БД создаются специальные таблицы текущих изменений и организуется система глобальной идентификации (именования) всех объектов распределенной системы, включая раздельное поименование одинаковых объектов в разных репликах (вплоть до записей таблиц). Такой подход несколько увеличивает объем БД, но позволяет значительно сократить транспортные расходы на синхронизацию реплик.

Решение проблемы обеспечения согласованности структуры данных основывается на технике главной реплики, суть которой заключается в следующем. Одна из реплик БД системы объявляется главной, причем изменять структуру данных можно только в этой главной реплике. Изменения в структуре данных в главной реплике тиражируются по принципу отложенных обновлений, т.е. с помощью синхронизации реплик. Выход из строя главной реплики не влечет за собой гибель всей распределенной информационной системы, так как остальные реплики продолжают функционировать автономно, что

позволяет администратору системы преобразовать любую реплику в главную и тем самым восстановить работоспособность всей системы.
Наряду с техникой главной реплики существует возможность создания частичных реплик. Частичной репликой называется база данных, содержащая ограниченное подмножество записей главной (полной) реплики. Распространенным способом создания частичных реплик является использование фильтров, устанавливаемых для таблиц главной реплики. Такой подход позволяет решать некоторые проблемы по разграничению доступа к данным, повысить производительность обработки данных и снизить затраты на синхронизацию реплик за счет ограничения количества передаваемых по сети изменений данных.

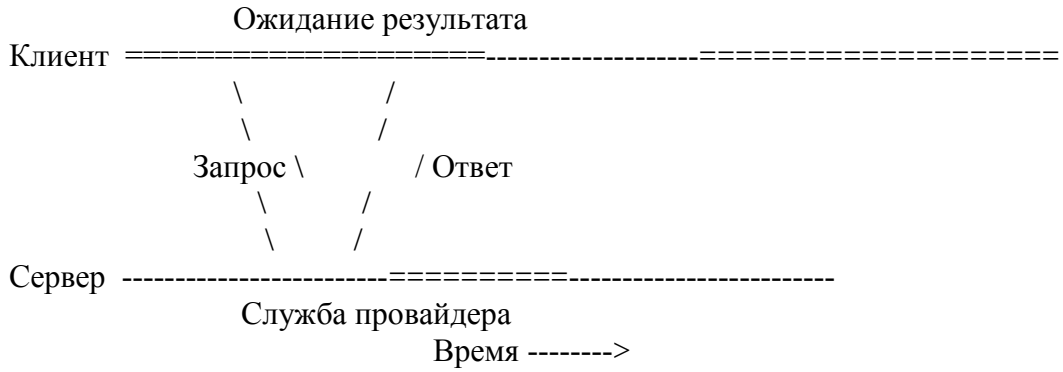
@Синхронизация данных:
В распределенных БД часто возникает проблема согласования данных, которые хранятся на различных компьютерах и в разных БД. Для решения ее разработчики БД интегрируют специальные приложения для синхронизации разрозненных данных, которые называются механизмами тиражирования. Механизм тиражирования должен обеспечить либо целостность данных в разных частях распределенной системы, либо их автономную работу.

Для ознакомления
Применение PKI для аутентификации предполагает издание цифровых сертификатов для пользователей (приложений), которые используются для непосредственной аутентификации на серверах БД в рамках одной организации. При этом не требуется использование дополнительного сервера аутентификации.
СУБД Oracle поддерживает протокол RADIUS - стандартный протокол для аутентификации удаленных пользователей. При этом становятся доступны службы и устройства аутентификации третьих производителей, с которыми может взаимодействовать сервер RADIUS (например, устройства генерации одноразовых паролей, биометрические устройства и т.п.).
Использование службы LDAP-каталога делает управление аутентификацией и управление учетными записями пользователей (приложений) очень эффективным. В инфраструктуре СУБД Oracle служба каталога представлена следующими компонентами:
- Oracle Internet Directory (OID) позволяет централизованно хранить и управлять информацией о пользователях (т.н. enterprise -пользователях). Позволяет иметь единственную учетную запись пользователя для многих баз данных. Возможна интеграция со службами каталогов третьих производителей, например, MS Active Directory или iPlanet . OID позволяет гибко управлять атрибутами безопасности и привилегиями каждого пользователя, включая тех, кто аутентифицируется по цифровым сертификатам. Для повышения безопасности во время процесса аутентификации возможно использование SSL -протокола.
- Oracle Enterprise Security Manager - утилита управления пользователями, группами, ролями и привилегиями.

##
###

58. Технология «клиент/сервер» и архитектура распределенных приложений. Понятие распределенной системы и требования, которым она должна удовлетворять. Модели распределенных вычислений и варианты распределения данных.

В модели клиент-сервер все процессы в распределённых система делятся на две возможно перекрывающиеся группы.
Процессы, реализующие некоторую службу, например службу файловой системы или базы данных, называются
серверами (servers). Процессы, запращивающие службы у серверов путем посылки запроса и последующего ожидания ответа от сервера,
называются клиентами (clients).
Взаимодействие клиента и сервера, известное также под названием режим работы запрос-ответ (request-reply behavior), иллюстрирует следующая диаграмма:



Если базовая сеть так же надёжна, как локальные сети, взаимодействие между клиентом и сервером может
быть реализовано посредством простого протокола, не требующего установления соединения.
В этом случае клиент, запрашивая службу, облекает свой запрос в форму сообщения с указанием в ней службы,
которой он желает воспользоваться, и необходимых для этого исходных данных. Затем сообщение посылается серверу.
Последний, в свою очередь, постоянно ожидает входящего сообщения, получив его, обрабатывает,
упаковывает результат обработки в ответное сообщение и отправляет его клиенту.

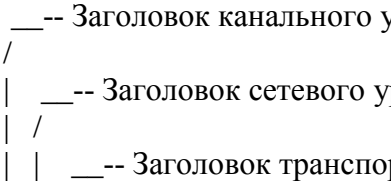
В качестве альтернативы во многих системах клиент-сервер используется надежный протокол с установкой соединения.
Хотя это решение в связи с его относительно низкой производительностью не слишком хорошо подходит для локальных сетей,
оно великолепно работает в глобальных системах, для которых ненадёжность является "врождённым" свойством соединений.
Так, практически все прикладные протоколы Интернета основаны на надёжных соединениях по протоколу TCP/IP. В этих случаях всякий раз, когда клиент запрашивает службу, до посылки запроса серверу он должен установить с ним соединение. Сервер обычно использует для посылки ответного сообщения то же самое соединение, после чего оно разрывается. Проблема состоит в том, что установка и разрыв соединения в смысле

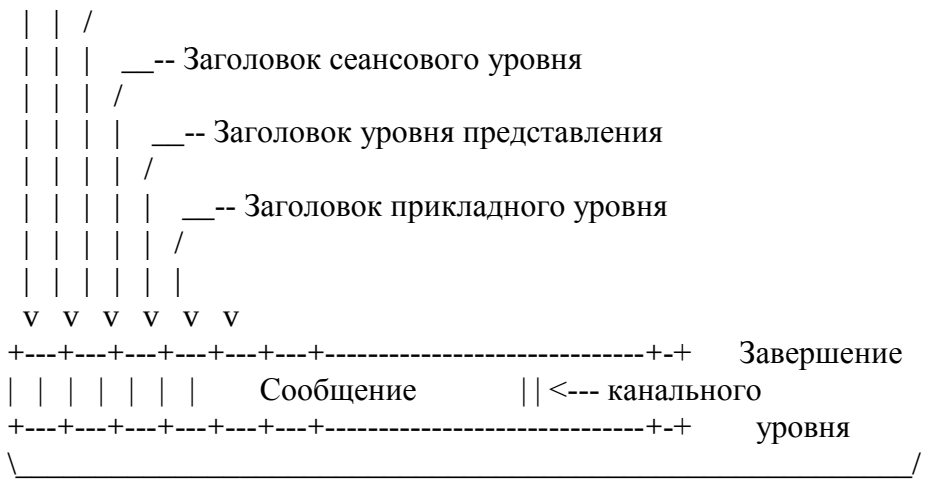
затрачиваемого времени и ресурсов относительно дороги, особенно если сообщения с запросом и ответом невелики.

Архитектура распределённых приложений.

МОДЕЛЬ OSI

ПЕРЕДАЧА ПО СЕТИ ТИПОВОГО СООБЩЕНИЯ





1) Физический уровень.
Физический уровень ответственен за передачу нулей и единиц. Сколько вольт использовать для передачи
нуля или единицы, сколько бит в секунду можно передать и можно ли осуществлять передачу одновременно
в двух направлениях - вот основные проблемы физического уровня. Кроме того, к физическому уровню
относится размер и форма сетевых коннекторов (разъемов), а также число выводов и назначение каждого из них.

Протоколы физического уровня отвечают за стандартизацию электрических, механических и сигнальных интерфейсов, чтобы, если одна машина посылает ноль, другая приняла его как ноль, а не как единицу.
Было разработано множество стандартов физического уровня для различных носителей, например стандарт
RS-232-C для последовательных линий связи (serial).

2) Канальный уровень.
Физический уровень только пересылает биты. Пока нет ошибок, все хорошо. Однако в реальных сетях происходят ошибки,
и их нужно как-то находить и исправлять. Это и является главной задачей канального уровня. Он группирует биты в модули,
обычно называемые кадрами (frames), и следит за тем, чтобы каждый кадр был передан правильно.

Канальный уровень делает это путем помещения специальной битовой маски в начало и конец каждого кадра для их маркировки,
а также путем вычисления контрольной суммы (checksum), то есть суммирования всех байтов кадра определённым образом.
Канальный уровень добавляет контрольную сумму к кадру. Когда кадр принимается, приёмник повторно вычисляет контрольную сумму данных
и сравнивает результат с контрольной суммой, пришедшей вместе с кадром. Если они совпадают, кадр считается верным и принимается.
Если они различны, получатель просит отправителя снова отправить этот кадр. Кадры последовательно нумеруются с указанием номеров в заголовке,
так что все понимают, где какой кадр.

3) Сетевой уровень.

В локальных сетях у отправителя обычно нет необходимости находить местоположение получателя.
Он просто бросает сообщение в локальную сеть, а получатель забирает его оттуда.
Глобальные сети, однако, содержат множество машин,
каждая из которых имеет собственные линии связи с другими машинами. Сообщение, посылаемое от отправителя к получателю,
должно пройти массу сетевых сегментов, на каждом из которых происходит выбор исходящей линии.
Задача выбора наилучшего пути называется маршрутизацией (routing) и является основной задачей сетевого уровня.

Проблема усложняется тем, что наиболее короткий путь не всегда является наилучшим.
На самом деле важна величина задержки на выбранном маршруте. Она, в свою очередь, зависит от объема трафика
и числа сообщений, стоящих в очереди на отправку по различным линиям. С течением времени задержка может меняться.
Некоторые алгоритмы маршрутизации могут подстраиваться под изменения загруженности линий,
некоторые же удовлетворяются тем, что принимают решение на основе усредненных значений.

В настоящее время, вероятно, наиболее широко распространённым сетевым протоколом является не требующий установки соединения протокол IP (Internet Protocol), используемый в Интернете. На сетевом уровне сообщение именуется термином пакет (packet).
IP-пакет может быть послан без какой-либо предварительной подготовки.
Маршрут каждого из IP-пакета до места назначения выбирается независимо от других пакетов.
Никакие внутренние пути не выбираются заранее и не запоминаются.

4) Транспортный уровень.
Транспортный уровень - это последняя часть того, что называют базовым стеком сетевых протоколов,
поскольку в нем реализованы все службы, которые необходимы для построения сетевых приложений и которые не вошли в интерфейс сеансового уровня.
Другими словами, транспортный уровень дает возможность разработчикам приложений использовать базовую сеть,
лежащую в его основе.

Функция транспортного уровня - надёжная доставка сообщения.
Приложение способно передать сообщение транспортному уровню в ожидании того, что оно будет доставлено без потери.
После получения сообщения с прикладного уровня транспортный уровень разбивает его для успешной передачи на достаточно мелкие части,
присваивает им последовательные номера и пересылает их. Взаимодействие на уровне заголовка транспортного уровня сводится к обсуждение того,
какой пакет был послан, какой - принят, сколько места есть у адресата для приёма дальнейших сообщений, что следует послать повторно и тому подобным вопросам.

Надёжное транспортное соединение (которое по определению представляет собой связь с установкой соединения)
можно построить поверх сетевых служб как с соединениями, так и без соединений.
В первом случае все пакеты будут доставлены в правильной последовательности (если они посылаются одновременно),
а в последнем возможно, что один из пакетов пойдет по другому маршруту и придет раньше, чем пакет,
посланный до него. Это побуждает программное обеспечение транспортного уровня складывать пакеты в правильной последовательности,
чтобы поддержать представление о транспортном соединении как о большой трубе - вы кладете в него сообщения на одном конце,
и они добираются до другого неповрежденными, в том же порядке, в котором и отправлялись.

5) Сеансовый уровень.
Сеансовый уровень представляет собой фактически расширенную версию транспортного уровня.
Он обеспечивает управление диалогом, отслеживая и запоминая, какая сторона говорит в настоящий момент,
и представляет средства синхронизации. Последние требуются для создания пользователями контрольных точек при длинных сеансах передачи данных,
а также уведомления их о сбое в ходе такого сеанса. При этом необходимо сделать откат только до последней контрольной точки
и не нужно проходить весь путь сначала.
На практике сеансовый уровень нужен немногим приложениям и поддерживается редко. Он даже не входит
в комплект протоколов Интернета (стек TCP/IP).

6) Уровень представления.
В отличие от предыдущих уровней, на которых мы заболитись о точной и эффективной пересылке битов
от отправителя к получателю, уровень представления занимается смыслом этих битов.
Большинство сообщений содержат не случайные последовательности битов, а структуированную информацию
типа фамилий, адресов, денежных сумм и т. п. На уровне представления можно определить записи, содердащие подобного рода поля,
и потребовать у отправителя уведомлять получателя, что сообщение содержит отдельные записи соответствующего формата.
Это упрощает взаимодействие между машинами с различным внутренним представлением данных.

7) Прикладной уровень.
Прикладной уровень модели OSI изначально должен был содержать набор стандартных сетевых приложений,
например для работы с электронной почтой, передачи файлов и эмуляции терминала.
В настоящее время он стал место собрания всех приложений и протоколов, которые не удалось пристроить ни на один из более низких уровней.
В свете эталонной модели OSI все распределенные системы являются просто приложениями.

__Распределенная система__ - это набор независимых компьютеров, представляющийся их пользователям единой объединенной системой.

Требования:
* Соединение пользователей с ресурсами. Основная задача распределённых систем - облегчить пользователям доступ к удалённым ресурсам
и обеспечить их совместное использование, регулируя этот процесс. Ресурсы могут быть виртуальными, однако традиционно
они включают в себя принтеры, компьютеры, устройства хранения данных, файлы и данные.
* Прозрачность. Важная задача распределённых систем состоит в том, чтобы скрыть тот факт, что процессы
и ресурсы физически распределены по множеству компьютеров. Распределённые системы,
которые представляются пользователям и приложениям в виде единой компьютерной системы, называются прозрачными (transparent).
* Открытость. Открытая распределённая система - это система, предлагающая службы, вызов которых требует стандартный синтаксис и семантику.
Например, в компьютерных сетях формат, содержимое и смысл посылаемых и принимаемых сообщений подчиняются типовым правилам.
Эти правила формализованы в протоколах.
* Модульность. В построении гибких распределённых систем решающим факторов оказывается организация
этих систем в виде набора относительно небольших и легко заменяемых или адаптируемых компонентов.
Это предполагает необходимость определения не только интерфейсов верхнего уровня, с которым работают пользователя и приложения,
но также и интерфейсов внутренних модулей системы и описания взаимодействия этих модулей.
* Масштабируемость. Масштабируемость может измеряться по трём различным показателям. Во-первых,
система может быть масштабируемой по отношению к её размеру, что означает легкость подключения к ней дополнительных пользователей и ресурсов.
Во-вторых, система может масштабироваться географически, то есть пользователя и ресурсы могут быть разнесены в пространстве.
В-третьих, система может быть масштабируемой в административном смысле, то есть быть проста в управлении
при работе во множестве административно независимых организаций.
* Отказоустойчивость. При выходе из строя одного компонента, система должна продолжать работать, пускай и с меньшей эффективностью.

Модели распределенных вычислений и варианты распределения данных.

https://books.ifmo.ru/file/pdf/1551.pdf <-- Модель распределённого вычисления. В каталоге docs/ книга `Distributed Computations.pdf`.

##
###

59. Организация взаимодействия компонентов распределенных приложений: протоколы прикладного уровня, понятие промежуточной среды и предоставляемые средой сервисы, примеры промежуточных сред. Технологии доступа к данным.

Протоколы прикладного уровня -- смотреть в вопросе 58.

__Промежуточный уровень__ - уровень между распределённым приложением и сетевым стеком операционной системы, обеспечивающий дополнительное абстрагирование.

Основная задача промежуточной среды - скрыть разнообразие базовых платформ от распределённых приложений. Для решения этой задачи многие системы промежуточного уровня предоставляют более или менее полные наборы служб и "не одобряют" желания использовать что-то ещё для доступа к этим службам, кроме своих интерфейсов. Другими словами, обход промежуточного уровня и непосредственный вызов служб одной из базовых операционных систем не приветствуется.

Примеры промежуточных сред:

1) Распределённые файловые системы.
2) Удалённые вызовы процедур (RPC).
3) Удалённые вызовы методов (RMI).
4) WWW (распределённые документы).

Технологии доступа к данным.

Существуют два способа доступа к данным, хранящимся на внешнем носителе:
* Файловая система и её операции.
* Базы данных и СУБД.

##
###

60. Понятие модели информационной системы (ИС). Статическая, динамическая и функциональная модели ИС; связь между ними; относительная важность. Концептуальная модель, модель спецификации и модель реализации; различия в интерпретации. Понятие метамодели.

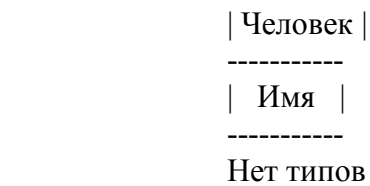
Информационная система – комплекс информационных ресурсов и технологий, предназначенных для сбора, хранения и обработки данных из некоторой предметной области (ресурсы как программные, так и аппаратные).
Модель – это абстрактное описание на некотором формальном языке некоторых аспектов системы, важных с точки зрения цели моделирования.
Моделирование – метод научного знания, заключающийся в изучении некоторого объекта посредством его моделей.

Модели по точке зрения на систему бывают:

- статические (описывают составные части системы, связи между ними)
Например: здание, охранники и камеры
- динамические (описывают поведение системы во времени, этапы, последовательность операций во времени)
Например: маршруты обхода, действия участников системы
Динамические модели имеют важность в интерактивных системах (работа в реальном времени, постоянное взаимодействие с пользователем)
- функциональные (отображает преобразование в системе)
Например: что происходит при нажатии на кнопку? нажатие на кнопку->электрические колебания->цифровой сигнал
Функциональные модели имеют важность в неинтерактивных системах (пакетный режим)
Для объектного подхода функциональные модели не характерны, все преобразования инкапсулированы, скрыты в реализации.

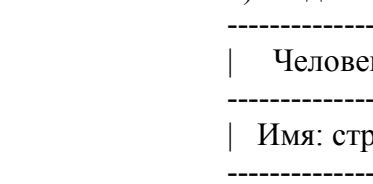
По степени абстракции модели бывают:
- концептуальные (нужна для анализа): они описывают задачу в терминах предметной области и понятия, как они между собой связаны;
- спецификации (описывается логическое решение, внешняя структура и поведение);
- реализации (максимально подробное описание, каким способом достигается это видимое поведение, структура наблюдаемой системы).

Пример:
1) Концептуальная модель человека



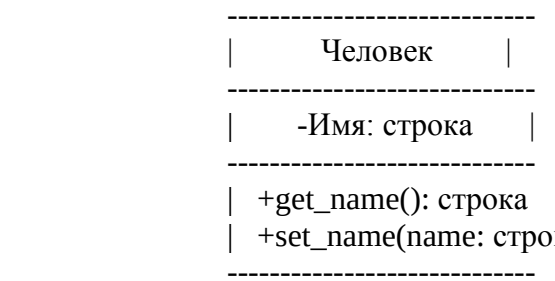
Нет типов и операций

2) Модель спецификации



Есть сущность – человек, он может сообщить свое имя, есть механизм его изменения. Нет информации о видимости интерфейса.

3) Модель реализации



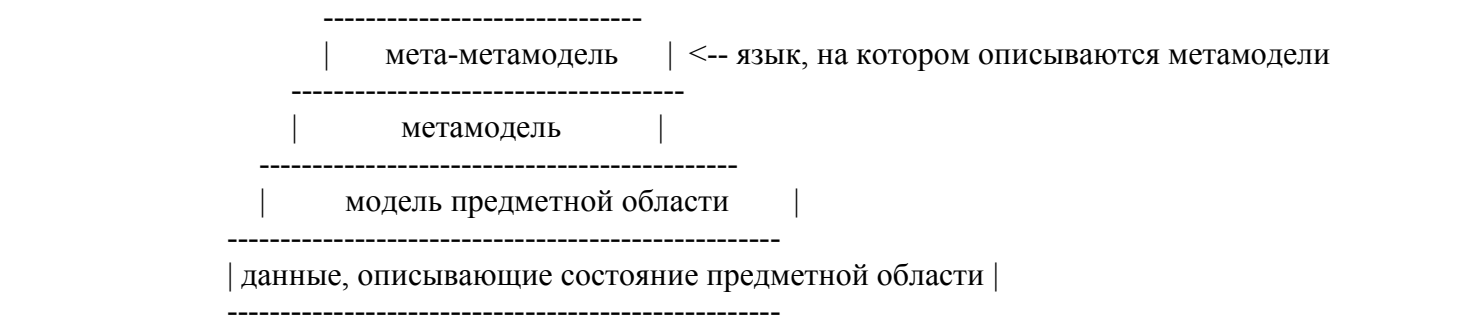
Имеется 2 открытых метода к полю имени

Концептуальная модель нужна всегда.

Модель спецификации нужна практически всегда (техническая спецификация, документация проекта).
Модель реализации строить трудоемко и она редко нужна. Рекомендуется строить в том случае, если есть хитрое решение и его надо задокументировать, описать для разработчиков-программистов.

Метамодель – это модель языка моделирования, применяемого для формализации описания системы.
Метамодели бывают лингвистические и онтологические. Лингистическая описывает предметноНЕзависимый язык моделирования, а онтологическая - предметнозависимый.

Классическая четырехуровневая иерархия моделей ИС:



##
###

61. Язык UML, определение и назначение. Обзор основных диаграмм языка. Возможности их применения на различных этапах жизненного цикла информационной системы.

UML – графический язык моделирования, представляющий собой систему обозначений, базирующуюся на диаграммах и предназначенную для визуализации, спецификации, конструирования и документирования систем, большая роль в которых принадлежит ПО.

Жизненный цикл ИС:
1) Анализ
2) Проектирование
3) Разработка
4) Тестирование

Основные диаграммы языка:
1) Диаграмма прецедентов (этап анализа)
Основная идея: описать все возможные ситуации, когда пользователю что-то нужно от системы.
Прецедент – описание множества содержательно близких сценариев взаимодействия акторов с системой, с целью достижения акторов своих целей.
Прецедент соответсвует задачам пользователя и описывается в терминах системных взаимодействий.
Три основных элемента: акторы (актор – внешний по отношению к системе объект, который с ней взаимодействует), прецеденты и отношения.
Пример описания прецедента:

Название: «Играть в игру»
Акторы: игрок
Описание: игрок бросает кости, система определяет результат и сообщает его игроку, если результат равен семи, то система предлагает игроку вписать свое имя в таблицу рекордов.

действия акторов	отклик системы
1. Игрок бросает кости	2. Система сообщает результат броска и предлагает просмотр таблицы рекордов или вернуться к игре
3. Игрок выбирает просмотр таблицы рекордов (E1)	4. Система показывает таблицу рекордов с полем ввода имени, если игрок выиграл (E2)
4. Игрок вводит имя (E3)	5. Система добавляет имя в таблицу рекордов (E4)
6. Игрок завершает работу с таблицей рекордов	7. Закрывает таблицу рекордов и возвращается к игре

E1 «вернуться к игре»: если игрок не хочет просматривать таблицу рекордов, то он возвращается к игре. Прецедент завершается.
E2 «игрок не выиграл»: если игрок не выиграл, то система показывает таблицу рекордов без поля ввода имени. Прецедент продолжается.
E3 «игрок не вводит имя»: если игрок не вводит имя, тогда он завершает работу с таблицей рекордов. Прецедент продолжается.
E4 «завершение»: если игрок не ввел имя, тогда система закрывает таблицу рекордов и возвращается к игре. Прецедент завершается.

Все прочие виды и примеры диаграмм рекомендую просмотреть бегло здесь (меняй страницы):
https://www.intuit.ru/studies/courses/1007/229/lecture/5954?page=1

##
###

62. Информационная безопасность в системе национальной безопасности Российской Федерации. Система обеспечения информационной безопасности России.

Раздел 5 документа "Доктрина ИБ РФ".

##
###

63. Основные понятия категории «безопасности», «информационная безопасность» (ФЗ «О безопасности», Доктрина информационной безопасности, Стратегия национальной безопасности, ГОСТ Р 50922-2006; системный подход). Общеметодологические принципы теории ИБ (общие понятия информационной безопасности, их взаимосвязь по ГОСТ Р ИСО/МЭК 15408-2002 (РД ОК)).

Для более общего понимания документов рекомендую БЕГЛО прочитать раздел 3 "Правовая основа обеспечения национальной безопасности РФ" из курса лекций Шободаевой "Основы теории национальной безопасности", все важные пункты там выделены жирным шрифтом :)

1) ФЗ «О безопасности»

Безопасность – состояние защищенности жизненно важных интересов личности, общества и государства от внутренних и внешних угроз. К основным объектам безопасности относятся личность, ее права и свободы, общество, его материальные и духовные ценности, государство, его конституционный строй, суверенитет и территориальная целостность.

Угроза безопасности - совокупность условий и факторов, создающих опасность жизненно важным интересам личности, общества и государства.

2) Доктрина ИБ

Под ИБ РФ понимается состояние защищенности ее национальных интересов в информационной сфере, определяющиеся совокупностью сбалансированных интересов личности, общества и государства.

Интересы личности - реализация конституционных прав человека и гражданина на доступ к информации, на использование информации в интересах осуществления не запрещенной законом деятельности, физического, духовного и интеллектуального развития, а также в защите информации, обеспечивающей личную безопасность.

Интересы общества - упрочении демократии, создании правового государства, достижении и поддержании общественного согласия, в духовном обновлении России.

Интересы государства - создание условий для гармоничного развития Российской информационной инфраструктуры, для реализации конституционных прав и свобод человека и гражданина в области получения информации и пользования ею в целях обеспечения незыблемости конституционного строя, суверенитета и территориальной целостности России, политической, экономической и социальной стабильности, в безусловном обеспечении законности и правопорядка, развитии равноправного и взаимовыгодного международного сотрудничества.

2) Стратегия национальной безопасности

Под национальной безопасностью РФ понимается безопасность ее многонационального народа как носителя суверенитета и единственного источника власти в РФ. Основными задачами в национальной безопасности РФ являются:

- своевременное прогнозирование и выявление внешних и внутренних угроз национальной безопасности РФ;
- реализация оперативных и долгосрочных мер по предупреждению и нейтрализации угроз;
- обеспечение суверенитета и территориальной целостности РФ, безопасности ее пограничного пространства;
- подъем экономики страны;
- преодоление научно-технической и технологической зависимости РФ от внешних источников;
- обеспечение на территории России личной безопасности человека и гражданина;
- совершенствование системы государственной власти РФ;
- обеспечение неукоснительного соблюдения законодательства РФ всеми;
- обеспечение сотрудничества России прежде всего с ведущими государствами мира;
- подъем и поддержание на достаточно высоком уровне военного потенциала государства;
- укрепление режима нераспространения оружия массового уничтожения;
- принятие эффективных мер по выявлению, предупреждению и пресечению разведывательной и подрывной деятельности;
- коренное улучшение экологической ситуации в стране.

3) ГОСТ Р 50922-2006

Защищаемая информация – информация, являющаяся предметом собственности и подлежащая защите в соответствии с требованиями правовых документов или требованиями, устанавливаемыми собственником информации.

Защита информации - деятельность по предотвращению утечки защищаемой информации, несанкционированных и непреднамеренных воздействий на защищаемую информацию.

Защита информации от утечки - деятельность по предотвращению неконтролируемого распространения защищаемой информации от ее разглашения, несанкционированного доступа к защищаемой информации и от получения защищаемой информации разведками.

Защита информации от НСД - деятельность по предотвращению получения защищаемой информации заинтересованным субъектом с нарушением установленных правовыми документами или собственником, владельцем информационных прав или правил доступа к защищаемой информации.

5) Общеметодологические принципы теории ИБ

Выделяют 4 компонента, которые присутствуют во всех подходах к понятию ИБ:
- обеспечение для субъекта доступа к достаточно полной и достоверной информации, необходимой для реализации его прав;
- защиту субъекта от деструктивных информационных воздействий;
- защиту от несанкционированного воздействия на информацию, принадлежащую субъекту;
- защиту информационной инфраструктуры группы субъектов от разрушительных воздействий.

Основным предметом информационного нападения всегда является информационное воздействие, т. е. то, что воспринимает субъект-нападающая сторона в случае попытки несанкционированного получения информации или объект нападения в случае попытки дезинформации, искажения информации, введения отвлекающей информации.

Системная задача обеспечения ИБ, с одной стороны, и конкретные задачи технических, юридических и других подсистем, с другой - имеют разные предметы действий; именно поэтому система информационной безопасности есть не простая сумма различных (правовых, организационных, технических) компонентов, но качественно отличное

явление. Существующее смешение понятий, объединение под одним термином «информация» различных предметов приводит либо к неоправданным попыткам оценивать содержательную сторону информационного процесса неадекватными методами, либо к самоограничению на уровне защиты исключительно документированной информации. И то и другое в конце концов приводит к нарушению защищенности объекта, создается объективная основа для произвольного определения факта нападения и для неадекватных действий защищающейся стороны.

Государственная политика обеспечения ИБ РФ основывается на принципах:
- соблюдении Конституции РФ, законодательства РФ, общепризнанных принципов и норм международного права при осуществлении деятельности по обеспечению ИБ РФ;
- открытости в реализации функций федеральных органов государственной власти, органов государственной власти субъектов РФ и общественных объединений, предусматривающей информирование общества об их деятельности с учетом ограничений, установленных законодательством РФ;
- правовом равенстве всех участников процесса информационного взаимодействия вне зависимости от их политического, социального и экономического статуса, основывающемся на конституционном праве граждан на свободный поиск, получение, передачу, производство и распространение информации любым законным способом;
- приоритетном развитии отечественных современных информационных и телекоммуникационных технологий, производстве технических и программных средств, способных обеспечить совершенствование национальных телекоммуникационных сетей, их подключение к глобальным информационным сетям в целях соблюдения жизненно важных интересов РФ.

ГОСТ Р ИСО/МЭК 15408-2002 - первая часть уже от 2012 года, а вторая и третья от 2013 года.

##
###

64. ГОСТ Р ИСО/МЭК 27002-2012 Менеджмент информационной безопасности. Политика информационной безопасности

Информация - это актив, который, подобно другим активам организации, имеет ценность и, следовательно, должен быть защищен надлежащим образом независимо от формы представления информации.
Информационная безопасность защищает информацию от широкого диапазона угроз с целью обеспечения уверенности в непрерывности бизнеса,
минимизации риска бизнеса, получения максимальной отдачи от инвестиций, а также реализации потенциальных возможностей бизнеса. Информационная безопасность достигается путем реализации соответствующего комплекса мер и средств контроля и управления, которые могут быть представлены политиками, процессами, процедурами, организационными структурами, а также функциями программных и аппаратных средств. Указанные меры и средства контроля и управления необходимо создавать, реализовывать, подвергать мониторингу, анализировать и улучшать, если необходимо, для обеспечения уверенности в том, что определенная безопасность и определенные цели бизнеса организации достигнуты. Все это необходимо выполнять наряду с другими процессами менеджмента бизнеса.

Организации, а также их информационные системы и сети сталкиваются с угрозами безопасности из широкого диапазона источников, включая компьютерное мошенничество, шпионаж, саботаж, вандализм, пожар или наводнение. Источники ущерба, например вредоносный код, компьютерное хакерство и атаки типа отказа в обслуживании, становятся более распространенными и все более и более изощренными.

При проектировании многих информационных систем проблемы безопасности не учитывались. Уровень безопасности, который может быть достигнут техническими средствами, имеет ряд ограничений и, следовательно, должен поддерживаться надлежащим менеджментом и процессами. Выбор необходимых мер и средств контроля и управления требует тщательного планирования и внимания к деталям. Менеджмент информационной безопасности нуждается, как минимум, в участии всех сотрудников организации. Кроме того, может потребоваться участие акционеров, поставщиков, представителей третьей стороны, клиентов или представителей других внешних сторон. Кроме того, могут потребоваться консультации специалистов сторонних организаций.

Организация должна определить свои требования к информационной безопасности. Существуют три основных источника требований безопасности:
- оценка рисков организации, принимая во внимание общую стратегию и цели бизнеса организации. Посредством оценки рисков идентифицируются угрозы активам организации, оцениваются уязвимости и вероятности возникновения угроз, а также оцениваются возможные последствия;
- правовые, законодательные, нормативные и договорные требования, которым должны удовлетворять организация, ее торговые партнеры, подрядчики и поставщики услуг, а также их социокультурная среда;
- набор принципов, целей и требований бизнеса для обработки информации, которые разработала организация для поддержки своей деятельности.

Оценка рисков:
Требования безопасности определяются с помощью систематической оценки рисков. Расходы на меры и средства контроля и управления должны быть соизмеримы с возможным ущербом бизнесу в результате отказа от обеспечения безопасности. Результаты оценки рисков помогут в определении конкретных мер и приоритетов в области менеджмента рисков информационной безопасности, а также внедрению мер и средств контроля и управления, выбранных для защиты от этих рисков. Оценка рисков должна периодически повторяться, чтобы учитывать любые изменения, которые могли бы повлиять на результаты оценки риска.

Выбор мер и средств контроля и управления:
После того как были определены требования к безопасности и риски безопасности и приняты решения в отношении обработки рисков, следует выбрать и внедрить такие меры и средства контроля и управления, которые обеспечат уверенность в снижении рисков до приемлемого уровня. Выбор мер и средств контроля и управления зависит от решений организации, основанных на критериях принятия рисков, вариантах обработки рисков и общем подходе к менеджменту рисков, применяемом в организации. При этом необходимо также учитывать все соответствующие национальные и международные законы и нормы.
Ключевыми мерами и средствами контроля и управления, с точки зрения законодательства, для организации являются:
- защита данных и конфиденциальность персональных данных;
- защита документов организации;
- права на интеллектуальную собственность.

Меры и средства контроля и управления, рассматриваемые как общепринятая практика в области информационной безопасности, включают:
- документирование политики информационной безопасности;
- распределение обязанностей по обеспечению информационной безопасности;
- осведомленность, обучение и тренинги;
- корректирующая обработка в прикладных программах;
- менеджмент технических уязвимостей;
- менеджмент непрерывности бизнеса;
- менеджмент инцидентов информационной безопасности и необходимое совершенствование.

Перечисленные меры и средства контроля и управления применимы для большинства организаций и сред.

Основные категории безопасности, рассмотренные в документе:
- политика безопасности;
- организационные аспекты информационной безопасности;
- менеджмент активов;
- безопасность, связанная с персоналом;
- физическая защита и защита от воздействия окружающей среды;
- менеджмент коммуникаций и работ;
- управление доступом;
- приобретение, разработка и эксплуатация информационных систем;
- менеджмент инцидентов информационной безопасности;
- менеджмент непрерывности бизнеса;
- соответствие.

Политика информационной безопасности:
Описано в вопросе №66.

##
###

65. Проблемы безопасности сети интернет

--!

##
###

66. Политика безопасности информационных систем

Политикой информационной безопасности (ИБ) называется комплекс мер, правил и принципов, которыми в своей повседневной практике руководствуются сотрудники предприятия в целях защиты информационных ресурсов. На основе ПБ строится управление, защита и распределение критической информации в системе. Она должна охватывать все особенности процесса обработки информации, определяя поведение ИС в различных ситуациях.

Эффективное обеспечение требуемого уровня информационной безопасности организации возможно только при наличии формализованного и системного подхода к выполнению мер по защите информации. Целью разработки политики информационной безопасности организации является создание единой системы взглядов и понимания целей, задач и принципов обеспечения информационной безопасности.

Основные этапы разработки политики информационной безопасности следующие:
- исследование текущего состояния информационной среды и информационной безопасности организации;
- анализ полученных сведений по результатам исследования;
- формирование плана работ по разработке политики информационной безопасности;
- разработка политика информационной безопасности организации.

Согласно стандарту ГОСТ Р ИСО/МЭК 27002-2012, политика информационной безопасности должна устанавливать ответственность руководства, а также излагать подход организации к управлению информационной безопасностью. В соответствии с указанным стандартом, необходимо, чтобы политика информационной безопасности предприятия как минимум включала:
– определение информационной безопасности, её общих целей и сферы действия, а также раскрытие значимости безопасности как инструмента, обеспечивающего возможность совместного использования информации;
– изложение целей и принципов информационной безопасности, сформулированных руководством;
– краткое изложение наиболее существенных для организации политик безопасности, принципов, правил и требований, например, таких как:
  – соответствие законодательным требованиям и договорным обязательствам;
  – требования в отношении обучения вопросам безопасности;
  – управление непрерывностью бизнеса;
  – ответственность за нарушения политики безопасности.
– определение общих и конкретных обязанностей сотрудников в рамках управления информационной безопасностью, включая информирование об инцидентах нарушения информационной безопасности;
– ссылки на документы, дополняющие политику информационной безопасности, например, более детальные политики и процедуры для конкретных информационных систем, а также правила безопасности, которым должны следовать пользователи.
Кроме того, политика информационной безопасности компании должна быть утверждена руководством, издана и доведена до сведения всех сотрудников в доступной и понятной форме.

Итоговый пакет организационно-распорядительных документов по вопросам обеспечения информационной безопасности включает следующие типы документов:
- политика информационной безопасности организации - высокоуровневый документ, описывающий основные принципы и правила, направленные на защиту информационных ресурсов организации;
- регламенты информационной безопасности, раскрывающие более подробно процедуры и методы обеспечения информационной безопасности в соответствии с основными принципами и правилами, описанными в политике;
- инструкции по обеспечению информационной безопасности для должностных лиц организации с учетом требований политики и регламентов;
- прочие документы, представляющие собой отчеты, регистрационные журналы и прочие низкоуровневые руководящие документы.

Адекватный уровень информационной безопасности в современной организации может быть обеспечен только на основе комплексного подхода, реализация которого начинается с разработки и внедрения эффективных политик безопасности. Такие политики определяют необходимый и достаточный набор требований безопасности, позволяющих

уменьшить риски информационной безопасности до приемлемой величины. Для того чтобы политика безопасности оставалась эффективной, необходимо осуществлять непрерывный контроль ее исполнения, повышать осведомленность сотрудников организации в вопросах безопасности и обучать их выполнению правил, предписываемых ею.

##
###

67. Требования к системам защиты информации

Требования по защите информации определяются владельцем ИС и согласовываются с исполнителем работ по созданию системы защиты информации (исполнитель должен иметь соответствующую лицензию на право проведения таких работ).

В процессе формирования требований к СЗИ целесообразно найти ответы на следующие вопросы:
- какие меры безопасности предполагается использовать?
- какова стоимость доступных программных и технических мер защиты?
- насколько эффективны доступные меры защиты?
- насколько уязвимы подсистемы СЗИ?
- имеется ли возможность провести анализ рисков (прогнозирование возможных последствий, которые могут вызвать выявленные угрозы и каналы утечки информации)?

В общем случае целесообразно выделить следующие группы требований к системам защиты информации:
1) Общие требования
Прежде всего, необходима полная идентификация пользователей, терминалов, программ, а также основных процессов и процедур, желательно до уровня записи или элемента. Кроме того, следует ограничить доступ к информации, используя совокупность следующих способов:
- иерархическая классификация доступа;
- классификация информации по важности и месту ее возникновения;
- указание ограничений к информационным объектам, например пользователь может осуществлять только чтение файла без права записи в него;
- определение программ и процедур, предоставленных только конкретным пользователям.
Система защиты должна гарантировать, что любое движение данных идентифицируется, авторизуется, обнаруживается и документируется.

2) Организационные требования
Организационные требования к системе защиты предусматривают реализацию совокупности административных и процедурных мероприятий. Например:
- ограничение привилегий персонала, обслуживающего ИС;
- осуществление записи протокола о доступе к системе;
- разработка последовательного подхода к обеспечению сохранности информации для всей организации;
- организация системы обучения и повышения квалификации обслуживающего персонала и т.п.

3) Конкретные требования к подсистемам защиты, техническому и программному обеспечению, документированию, способам, методам и средствам защиты
СЗИ целесообразно условно разделить на подсистемы:

а) Управления доступом к ресурсам ИС (включает также функции управления системой защиты в целом):

Идентификация, аутентификация, контроль доступа к системе; управление потоками информации; очистку освобождаемых областей памяти.

б) Регистрации и учета действий пользователей (процессов):

Регистрация и учет доступа в ИС, выдача доступа к защищаемым файлам, передача данных; учет носителей информации; оповещения о попытках нарушения защиты.

в) Криптографическую:

Шифрование информации, использование аттестованных криптографических средств.

г) Обеспечения целостности информационных ресурсов и конфигурации ИС:

Обеспечение целостности программных средств и обрабатываемой информации; наличие средств восстановления; контроль целостности; резервное копирование; обнаружение и блокирование вирусов.

Для каждой из подсистем определяются требования в виде:
1) Перечня обеспечиваемых подсистемой функций защиты
2) Основных характеристик этих функций
3) Перечня средств, реализующих эти функции

Программные средства ЗИ должны обеспечивать контроль доступа, безопасность и целостность данных и защиту самой СЗ. Для этого необходимо выполнить следующие условия:
1) Объекты защиты должны идентифицироваться в явном виде при использовании паролей, пропусков и идентификации по голосу
2) Система контроля доступа должна быть достаточно гибкой для обеспечения многообразных ограничений и различных наборов объектов
3) Каждый доступ к файлу данных или устройству должен прослеживаться через систему контроля доступа для того, чтобы фиксировать и документировать любое обращение.

##
###

68. Четырехуровневая система как метод анализа информационной безопасности

Проблема обеспечения ИБ - проблема комплексная, защищать приходится сложные системы, и сами защитные средства тоже сложны.
Существует три грани: доступность, целостность и конфиденциальность. Их можно рассматривать относительно независимо, и считается, что если все они обеспечены, то обеспечена и ИБ в целом. Введем следующие уровни системы:

- законодательные меры обеспечения информационной безопасности;
- административные меры (приказы и другие действия руководства организаций, связанных с защищаемыми информационными системами);
- процедурные меры (меры безопасности, ориентированные на людей);
- программно-технические меры.

Законы и нормативные акты ориентированы на все субъекты информационных отношений, административные меры - на все субъекты в пределах организации, процедурные – на отдельных людей, программно-технические – на оборудование и программное обеспечение. При такой трактовке в переходе с уровня на уровень можно усмотреть применение наследования (каждый следующий уровень не отменяет, а дополняет предыдущий), а также полиморфизма (субъекты выступают сразу в нескольких ипостасях - например, как инициаторы административных мер и как обычные пользователи, обязанные этим мерам подчиняться). Также действует принцип инкапсуляции (это и значит, что грани "относительно независимы").

Законодательный уровень является важнейшим для обеспечения ИБ. Большинство людей не совершают противоправных действий не потому, что это технически невозможно, а потому, что это осуждается и/или наказывается обществом, потому, что так поступать не принято.
На законодательном уровне различают две группы мер:
- меры, направленные на создание и поддержание в обществе негативного (в том числе с применением наказаний) отношения к нарушениям и нарушителям ИБ (мерами ограничительной направленности);
- направляющие и координирующие меры, способствующие повышению образованности общества в области ИБ, помогающие в разработке и распространении средств обеспечения ИБ (меры созидательной направленности).

Самое важное на законодательном уровне - создать механизм, позволяющий согласовать процесс разработки законов с реалиями и прогрессом информационных технологий. Законы не могут опережать жизнь, но важно, чтобы отставание не было слишком большим, так как на практике, помимо прочих отрицательных моментов, это ведет к снижению информационной безопасности.
Основополагающим среди российских законов, посвященных вопросам информационной безопасности, следует считать закон "Об информации, информационных технологиях и о защите информации". В нем даются основные определения, намечаются направления.

К административному уровню ИБ относятся действия общего характера, предпринимаемые руководством организации. Главная цель мер административного уровня - сформировать программу работ в области ИБ и обеспечить ее выполнение, выделяя необходимые ресурсы и контролируя состояние дел.
Основой программы является политика безопасности, отражающая подход организации к защите своих информационных активов. Руководство каждой организации должно осознать необходимость поддержания режима безопасности и выделения на эти цели значительных ресурсов.
Политика безопасности строится на основе анализа рисков, которые признаются реальными для информационной системы организации. Когда риски проанализированы и стратегия защиты определена, составляется программа обеспечения ИБ. Под эту программу выделяются ресурсы, назначаются ответственные, определяется порядок контроля выполнения программы и т.п.
Под политикой безопасности будем понимать совокупность документированных решений, принимаемых руководством организации и направленных на защиту информации и ассоциированных с ней ресурсов. С практической точки зрения политику безопасности целесообразно рассматривать на трех уровнях детализации. К верхнему уровню можно отнести решения, затрагивающие организацию в целом. Они носят весьма общий характер и, как правило, исходят от руководства организации. Примерный список подобных решений может включать в себя следующие элементы:
- решение сформировать или пересмотреть комплексную программу обеспечения информационной безопасности, назначение ответственных за продвижение программы;

- формулировка целей, которые преследует организация в области информационной безопасности, определение общих направлений в достижении этих целей;
- обеспечение базы для соблюдения законов и правил;
- формулировка административных решений по тем вопросам реализации программы безопасности, которые должны рассматриваться на уровне организации в целом.

Для политики верхнего уровня цели организации в области информационной безопасности формулируются в терминах целостности, доступности и конфиденциальности. Если организация отвечает за поддержание критически важных баз данных, на первом плане может стоять уменьшение числа потерь, повреждений или искажений данных. Для организации, занимающейся продажей компьютерной техники, вероятно, важна актуальность информации о предоставляемых услугах и ценах и ее доступность максимальному числу потенциальных покупателей. Руководство режимного предприятия в первую очередь заботится о защите от несанкционированного доступа, то есть о конфиденциальности.

В политике должны быть определены обязанности должностных лиц по выработке программы безопасности и проведению ее в жизнь. В этом смысле политика безопасности является основой подотчетности персонала.

Процедурные меры ориентированы на людей, а не на технические средства. Именно люди формируют режим ИБ, и они же оказываются главной угрозой, поэтому "человеческий фактор" заслуживает особого внимания.

В российских компаниях накоплен богатый опыт регламентирования и реализации процедурных (организационных) мер, однако дело в том, что они пришли из "докомпьютерного" прошлого, поэтому требуют переоценки.

Следует осознать ту степень зависимости от компьютерной обработки данных, в которую попало современное общество. Без всякого преувеличения можно сказать, что необходима информационная гражданская оборона. Спокойно, без нагнетания страстей, нужно разъяснять обществу не только преимущества, но и опасности, связанные с использованием информационных технологий. Акцент следует делать не на военной или криминальной стороне дела, а на гражданских аспектах, связанных с поддержанием нормального функционирования аппаратного и программного обеспечения, то есть концентрироваться на вопросах доступности и целостности данных.

На процедурном уровне можно выделить следующие классы мер:
- управление персоналом;
- физическая защита;
- поддержание работоспособности;
- реагирование на нарушения режима безопасности;
- планирование восстановительных работ.

Программно-технические меры, то есть меры, направленные на контроль компьютерных сущностей - оборудования, программ и/или данных, образуют последний и самый важный рубеж информационной безопасности. Напомним, что ущерб наносят в основном действия легальных пользователей, по отношению к которым процедурные регуляторы малоэффективны. Главные враги - некомпетентность и неаккуратность при выполнении служебных обязанностей, и только программно-технические меры способны им противостоять.

Компьютеры помогли автоматизировать многие области человеческой деятельности. Вполне естественным представляется желание возложить на них и обеспечение собственной безопасности. Даже физическую защиту все чаще поручают не охранникам, а интегрированным компьютерным системам, что позволяет одновременно отслеживать перемещения сотрудников и по организации, и по информационному пространству.

Центральным для программно-технического уровня является понятие сервиса безопасности:

- идентификация и аутентификация;
- управление доступом;
- протоколирование и аудит;
- шифрование;
- контроль целостности;
- экранирование;
- анализ защищенности;
- обеспечение отказоустойчивости;
- обеспечение безопасного восстановления;
- туннелирование;
- управление.

##
###

69. Уголовно-правовая характеристика состава преступлений, предусмотренных ст. 272-274 Уголовного кодекса РФ

Терминология:
Уголовное право — это отрасль права, регулирующая общественные отношения, связанные с совершением преступных деяний, назначением наказания и применением иных мер уголовно-правового характера, устанавливающая основания привлечения к уголовной ответственности либо освобождения от уголовной ответственности и наказания.
Состав преступления – совокупность предусмотренных законом объективных и субъективных признаков, характеризующих совершенное общественно-опасное деяние как конкретный вид преступления. Состав преступления – необходимое основание уголовной ответственности. Состав преступления образуют четыре группы признаков, характеризующие объект преступления, его объективную сторону, субъект преступления и субъективную сторону.
Объект преступления – элемент состава преступления, конкретные охраняемые уголовным законом общественные отношения, на которые посягается виновный.
Объективная сторона преступления – совершенные виновным конкретные действия (бездействие), представляющие общественную опасность и запрещенные УК.
Субъект преступления – элемент состава преступления, вменяемое физическое лицо, достигшее предустановленного уголовным законом возраста.
Субъективная сторона преступления – психическое отношение лица к совершаемому или общественно опасному деянию.

Преступления в сфере компьютерной информации — общественно опасные деяния, совершаемые в сфере компьютерной информации, признаваемые преступлениями уголовным законодательством. В соответствии с действующим уголовным законодательством Российской Федерации под преступлениями в сфере компьютерной информации понимаются совершаемые в сфере информационных процессов и посягающие на информационную безопасность деяния, предметом которых являются информация и компьютерные средства.
По УК РФ преступлениями в сфере компьютерной информации являются:

- неправомерный доступ к компьютерной информации (ст. 272 УК РФ);

- создание, использование и распространение вредоносных программ для ЭВМ (ст. 273 УК РФ);
- нарушение правил эксплуатации ЭВМ, системы ЭВМ или их сети (ст. 274 УК РФ).

Общественная опасность противоправных действий в области электронной техники и информационных технологий выражается в том, что они могут повлечь за собой нарушение деятельности автоматизированных систем управления и контроля различных объектов, серьёзное нарушение работы ЭВМ и их систем, несанкционированные действия по уничтожению, модификации, искажению, копированию информации и информационных ресурсов, иные формы незаконного вмешательства в информационные системы, которые способны вызвать тяжкие и необратимые последствия, связанные не только с имущественным ущербом, но и с физическим вредом людям.
Неправомерный доступ к компьютерной информации (ст. 272 УК РФ), а также создание, использование и распространение вредоносных программ для ЭВМ (ст. 273 УК РФ) совершаются только путём действий, в то время как нарушение правил эксплуатации ЭВМ, системы ЭВМ или их сети (ст. 274 УК РФ) — путём как действий, так и бездействием.
Неправомерный доступ к компьютерной информации и нарушение установленных правил эксплуатации ЭВМ, системы ЭВМ или их сети сформулированы как преступления с материальным составом, а создание либо использование вредоносных программ для ЭВМ — с формальным. В качестве последствий в ст. 272 и 274 УК указываются: уничтожение, модификация, блокирование либо копирование информации, нарушение работы ЭВМ или системы ЭВМ, причинение существенного вреда и т. п.
Родовым объектом преступлений в сфере компьютерной информации является общественная безопасность и порядок в отношениях, связанных с информационными процессами - процессами сбора, обработки, накопления, хранения, поиска и распространения информации, с использованием ЭВМ, их систем и сетей. Существенно то, что предметом данных преступлений является компьютерная информация, а не информационное оборудование, обеспечивающее информационные процессы. Правонарушения, совершенные в ходе данных процессов, не связанные с использованием указанного оборудования, квалифицируются с помощью иных статей УК РФ, предусматривающих ответственность за соответствующие конкретные действия.
Непосредственным объектом данных преступных деяний является безопасность информационных систем, базирующихся на использовании ЭВМ, системе ЭВМ или их сети.
Объективная сторона компьютерных преступлений характеризуется как действием, так и бездействием. Действие (бездействие) сопряжено с нарушением прав и интересов по поводу пользования компьютерной информацией.
Компьютерные преступления имеют материальные составы. Действие (бездействие) должно причинить значительный вред правам и интересам личности, общества или государства (исключением является преступление с формальным составом, предусмотренное ч. 1 ст.273 УК: создание, использование и распространение вредоносных программ для ЭВМ). Преступные последствия конкретизируются в законе применительно к конкретным видам компьютерных преступлений. Между деянием и последствиями обязательно должна быть установлена причинная связь.
Субъективная сторона компьютерных преступлений характеризуется умышленной виной. В ч. 2 ст. 24 сказано, что деяние совершенное по неосторожности признается преступлением только тогда, когда это специально предусмотрено соответствующей статьей Особенной части УК. Неосторожная форма вины названа в Особенной части лишь применительно к квалифицированным видам компьютерных преступлений, предусмотренных в ч. 2 ст. 273 и ч.2 ст. 274 УК.

Субъект компьютерного преступления общий - лицо, достигшее 16 лет. В ст. 274 и в ч. 2 ст. 272 УК формулируются признаки специального субъекта: лицо, имеющее доступ к ЭВМ, системе ЭВМ или их сети.
Преступление в сфере компьютерной информации - это предусмотренное уголовным законом виновное нарушение чужих прав и интересов в отношении автоматизированных систем обработки данных, совершенное во вред подлежащим правовой охране правам и интересам физических и юридических лиц, общества и государства.

Статья 272. Неправомерный доступ к компьютерной информации
1. Неправомерный доступ к охраняемой законом компьютерной информации, то есть информации на машинном носителе, в электронно-вычислительной машине (ЭВМ), системе ЭВМ или их сети, если это деяние повлекло уничтожение, блокирование, модификацию либо копирование информации, нарушение работы ЭВМ, системы ЭВМ или их сети, - наказывается штрафом в размере от двухсот до пятисот минимальных размеров оплаты труда или в размере заработной платы или иного дохода осужденного за период от двух до пяти месяцев, либо исправительными работами на срок от шести месяцев до одного года, либо лишением свободы на срок до двух лет.
2. То же деяние, совершенное группой лиц по предварительному сговору или организованной группой либо лицом с использованием своего служебного положения, а равно имеющим доступ к ЭВМ, системе ЭВМ или их сети, - наказывается штрафом в размере от пятисот до восьмисот минимальных размеров оплаты труда или в размере заработной платы или иного дохода осужденного за период от пяти до восьми месяцев, либо исправительными работами на срок от одного года до двух лет, либо арестом на срок от трех до шести месяцев, либо лишением свободы на срок до пяти лет.

Статья 273. Создание, использование и распространение вредоносных программ для ЭВМ
1. Создание программ для ЭВМ или внесение изменений в существующие программы, заведомо приводящих к несанкционированному уничтожению, блокированию, модификации либо копированию информации, нарушению работы ЭВМ, системы ЭВМ или их сети, а равно использование либо распространение таких программ или машинных носителей с такими программами - наказываются лишением свободы на срок до трех лет со штрафом в размере от двухсот до пятисот минимальных размеров оплаты труда или в размере заработной платы или иного дохода осужденного за период от двух до пяти месяцев.
2. Те же деяния, повлекшие по неосторожности тяжкие последствия, - наказываются лишением свободы на срок от трех до семи лет.

Статья 274. Нарушение правил эксплуатации ЭВМ, системы ЭВМ или их сети
1. Нарушение правил эксплуатации ЭВМ, системы ЭВМ или их сети лицом, имеющим доступ к ЭВМ, системе ЭВМ или их сети, повлекшее уничтожение, блокирование или модификацию охраняемой законом информации ЭВМ, если это деяние причинило существенный вред, - наказывается лишением права занимать определенные должности или заниматься определенной деятельностью На срок до пяти лет, либо обязательными работами на срок от ста восьмидесяти до двухсот сорока часов, либо ограничением свободы на срок до двух лет.
2. То же деяние, повлекшее по неосторожности тяжкие последствия, - наказывается лишением свободы на срок до четырех лет.

##
###

70. Организация государственного контроля и надзора за соблюдением защиты информации в РФ

Для решения основных задач в сфере ИБ действуют все основные органы государственной власти и управления: судебные, органы исполнительной власти, правоохранительные органы, организации и предприятия, которые контролируются государством и имеют доступ к информации, составляющей государственную тайну, и другие.
Основой современной политики РФ в сфере ИБ можно считать "Доктрину ИБ РФ". Важными организующими документами также являются: ФЗ "О государственной тайне", "Об информации, информационных технологиях и о защите информации", "Об участии в международном информационном обмене".

Основным государственным органом, определяющим политику РФ в сфере безопасности страны в целом и ИБ в частности, является Совет безопасности РФ.
Ведущим государственным учреждением, непосредственно ответственным за реализацию государственной политики в сфере ИБ и защиту государственных интересов на общенациональном уровне, является Федеральная служба по техническому и экспортному контролю – ФСТЭК. Важную роль в системе органов государственной власти, отвечающих за решение задач ИБ, играет также Служба специальной связи и информации ("Спецсвязь России"), с 2004 года входящая в состав Федеральной службы охраны.
Вопросы повышения качества информационной работы и ИБ решают также другие федеральные органы (в пределах своей компетенции):
- Министерство связи и массовых коммуникаций РФ;
- Министерство внутренних дел РФ.

, а также отдельные государственные ведомства, предъявляющие особые требования к уровню защищенности информации, реализуют собственные мероприятия по обеспечению защиты информации:
- ФСБ (Управление компьютерной и ИБ, а также Центр по лицензированию, сертификации и защите государственной тайны, Управление специальной связи и НИИ информационных технологий);
- Минатом РФ и система подведомственных ему предприятий (в составе которого функционирует Центр "Атомзащитаинформ");
- Центральный банк РФ (в составе которого функционирует Главное управление безопасности и защиты информации)
- и некоторые другие.

Совет Безопасности РФ, возглавляемый Президентом РФ, состоит из ключевых министров и рассматривает вопросы внутренней и внешней политики РФ в области обеспечения безопасности, стратегические проблемы государственной, экономической, общественной, оборонной, информационной, экологической и иных видов безопасности.
Функции СБ:
- подготовка решений Президента РФ по соответствующим вопросам, в т.ч. по вопросам ИБ;
- рассмотрение законопроектов, в рамках своей компетенции;
- организация и координация разработки стратегии в области внутренней, внешней и военной политики, военно-технического сотрудничества и ИБ РФ.

Для решения задач, связанных с обеспечением ИБ, в составе СБ функционирует Управление ИБ и Межведомственная комиссия по ИБ. Функциями Управления ИБ являются:
- подготовка предложений Совету Безопасности по выработке и реализации основных направлений политики государства в области обеспечения ИБ РФ;

- анализ и прогнозирование ситуации в области ИБ РФ;
- выявление источников опасности, оценка внешних и внутренних угроз ИБ и подготовка предложений Совету Безопасности по их предотвращению;
- рассмотрение проектов федеральных целевых программ, направленных на обеспечение ИБ РФ, подготовка соответствующих предложений;
- участие в подготовке материалов по вопросам обеспечения ИБ РФ для ежегодного послания Президента РФ Федеральному Собранию;
- подготовка предложений по проектам решений Совета Безопасности и информационно-аналитических материалов к его заседаниям по вопросам обеспечения ИБ РФ;
- подготовка предложений Совету Безопасности по разработке проектов нормативных правовых актов, направленных на обеспечение ИБ РФ.

Федеральная служба по техническому и экспортному контролю (ФСТЭК), до августа 2004 года известная как Гостехкомиссия РФ.
Основными функциями ФСТЭК являются:
- проведение единой технической политики и координация работ по ЗИ;
- организация и контроль за проведением работ по ЗИ в органах государственного управления, объединениях, концернах, на предприятиях, в организациях и учреждениях (независимо от форм собственности) от утечки по техническим каналам, от несанкционированного доступа к информации, обрабатываемой техническими средствами, и от специальных воздействий на информацию с целью ее уничтожения и искажения;
- поддержание системы лицензирования деятельности предприятий, организаций и учреждений по осуществлению мероприятий и (или) оказанию услуг в области защиты информации и сертификации средств защиты информации.

Служба специальной связи и информации (Спецсвязь России), созданная в марте 2003 года в рамках Федеральной службы охраны на базе упраздненного Федерального агентства правительственной связи и информации (ФАПСИ), в целом призвана обеспечивать функционирование президентской связи, организацию, эксплуатацию и развитие специальной связи для государственных органов и решать другие аналогичные задачи.
При этом задачами Спецсвязи также являются:
- проведение работ по защите технических средств специальной связи, устанавливаемых в категорированных помещениях государственных органов, включая особо важные;
- организация в системе специальной связи шифровальной деятельности, отнесенной к компетенции Спецсвязи России;
- участие в разработке нормативной технической документации по вопросам защиты информации в системах специальной связи;
- участие в разработке и реализации мер по обеспечению информационной безопасности Российской Федерации, защите сведений, составляющих государственную тайну;
- участие в создании, обеспечении и развитии системы электронного документооборота государственных органов с использованием удостоверяющих центров;
- организация и проведение мероприятий по предотвращению утечки по техническим каналам информации в системах специальной связи, информационно-технологических, информационно-аналитических и информационно-телекоммуникационных системах, находящихся в ведении Спецсвязи России;
- выполнение требований обеспечения информационной безопасности объектов государственной охраны.

Министерство связи и массовых коммуникаций РФ в лице подчиняющегося ему Федерального агентства по информационным технологиям (Росинформтехнологии) осуществляет и организует следующие виды работ в сфере ИБ:
- подтверждение подлинности электронных цифровых подписей уполномоченных лиц удостоверяющих центров в выданных ими сертификатах ключей подписей;
- ведение единого государственного реестра сертификатов ключей подписей удостоверяющих центров и реестра сертификатов ключей подписей уполномоченных лиц федеральных органов государственной власти, а также обеспечение доступа к ним граждан, организаций, органов государственной власти и органов местного самоуправления;
- выполнение функции государственного заказчика научно-технических и инвестиционных программ и проектов в сфере информационных технологий.

Федеральная служба по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзор) является уполномоченным федеральным органом исполнительной власти по защите прав субъектов персональных данных. В полномочия данного органа входит пресечение нарушений, которые могут возникать при обработке персональных данных граждан РФ.

В системе законодательной власти основным структурным подразделением, призванным решать вопросы формирования и реализации государственной политики в сфере ИБ, является Комитет по безопасности Государственной думы Федерального собрания Российской Федерации. В составе этого Комитета функционирует Подкомитет по ИБ. В законодательной работе в рамках этого Комитета принимают участие:
- специалисты и руководители профильных подразделений ФСБ, СВР, ФСТЭК, МВД и других ведомств;
- руководители Совета безопасности РФ и других правительственных органов;
- представители общественных организаций, фондов и профессиональных объединений;
- представители крупных коммерческих компаний – лидеров в развитии организации и технологий ИБ;
- представители ведущих научно-исследовательских учреждений и учебных заведений.

##
###

71. Классификация информации с точки зрения ФЗ «Об информации, информационных технологиях и информационной безопасности».

Указанный ФЗ регулирует отношения, возникающие при осуществлении права на поиск, получение, передачу, производство и распространение информации; применении информационных технологий; обеспечении защиты информации.

Информация – это сведения (сообщения, данные) независимо от формы их представления. Информация может являться объектом публичных, гражданских и иных правовых отношений.
По степени доступа информация делится на открытую и ограниченного доступа, распространение которой возможно в условиях конфиденциальности или секретности. Информация в зависимости от порядка ее распространения подразделяется на:

- информацию, свободно распространяемую;
- информацию, предоставляемую по соглашению лиц, участвующих в соответствующих отношениях;
- информацию, которая в соответствии с ФЗ подлежит предоставлению или распространению;
- информацию, распространение которой в РФ ограничивается или запрещается.

Открытая информация:
- информация как объект гражданских прав (произведения, патенты) – произведения науки и литературы, другие формы, отражающие информацию (карты, фотографии), а также информация, содержащаяся в документах, закрепляющих авторские права на изобретения, полезные модели, промышленные образцы;
- массовая информация – информация, содержащая сообщения информационного характера и распространяемая СМИ и (или) через Интернет целью информормирования населения, в том числе реклама деятельности физических и юридических лиц, производимых продуктов, услугах, предлагаемых потребителям;
- информация о выборах, референдуме;
- официальные документы – законы, судебные решения, информационные тексты законодательного, администтивного и судебного характера, а также их официальные переводы;
- обязательно представляемая информация – обязательные контрольные экземпляры документов, данные документов, представляемых в органы статистики, налоговая, регистрационная и др. такого типа информация; такая информация создается юридическими и физическими лицами в порядке учета и отчетности и направляется по разным органам в соответствии с законодательством;
- другая открытая информация.

Ограниченного доступа:
- государственная тайна, служебная тайна – защищаемые государством сведения, создаваемые в условиях секретности в соответствии с законодательством РФ;
- ноу-хау (секреты производства) и коммерческая тайна; коммерческая тайна – научно-техническая, технологическая, организационная или иная используемая в экономической деятельности информация, включая ноу-хау. Режим такой информации устанавливается законом;
- персональные данные – создается самими гражданами в их повседневной деятельности, в том числе связанной с реализацией прав и свобод (права на труд, на жилище, на отдых, мед. обследование, пенсионное обеспечение, на свободу слова и др.) и выполнением обязанностей (напр, воинской) и предоставляется как сведения о себе разным субъектам. Документированной информацией здесь являються анкеты, истории болезни, декларация о доходах, банковские записи;
- другие виды тайн.

##
###

72. Информация как предмет частных правоотношений.

73. Информация как предмет публичных правоотношений.

Лучше рассматривать в сравнении эти два вопроса!

«Все элементарно!», - пишет Каримов. «Частные отношения возникают между гражанами, например сохранение коммерческой тайны или при покупке имеешь полную информацию о продукте на русском языке... А вот если вмешивается государство (гостайна, или там регистрация средств связи или нового журнала), то это уже публичные правоотношения...»

Предмет правового регулирования информационного права составляют общественные отношения, возникающие, изменяющиеся и прекращающиеся при обращении информации в информационной сфере в результате осуществления информационных процессов.
Принципы информационного права:
- принцип приоритетности прав личности;
- принцип запрещения производства и распространения информации, вредной и опасной для развития личности, общества, государства;
- принцип свободного доступа (открытости) информации, не ограниченной ФЗ;
- принцип законности - субъекты информационного права обязаны строго соблюдать Конституцию РФ и законодательство РФ;
- принцип ответственности применительно к информационно-правовому регулированию означает неотвратимое наступление ответственности за нарушение требований и предписаний информационно-правовых норм;
- принцип «отчуждения» информации от ее создателя основан на юридическом свойстве физической неотчуждаемости информации (ее содержания) от ее создателя (обладателя).

Информационные отношения частноправового плана - это, главным образом, имущественные отношения и личные неимущественные отношения, проявляющиеся в информационной сфере. Особенность такого вида информационных отношений во многом зависит и даже определяется теми объектами, по поводу которых они возникают именно в информационной сфере.
Общественные отношения, возникающие между потребителем информации — с одной стороны, и производителем информации и услуг, чаше всего регулируются традиционными нормами гражданского права или публичного права.
Основным предметом правового регулирования информационного права выступают общественные отношения в информационной сфере, возникающие при осуществлении информационных процессов - процессов производства, сбора, обработки, накопления, хранения, поиска, передачи, распространения и потребления информации.
Частное право – собирательное понятие, означающее отрасли права, регулирующие частные интересы, независимость и инициативу индивидуальных собственников и объединений (корпораций) в их имущественной деятельности и в личных отношениях, в отличие от публичного права, которое регулирует и охраняет общие интересы. Ядро частного права составляет гражданское право, регулирующее имущественные и связанные с ними неимущественные отношения.
Особенности информационных отношений сводятся к тому, что эти отношения:
- возникают, развиваются и прекращаются в информационной сфере при обращении к информации;
- опосредуют государственную политику признания, соблюдения и защиты информационных прав и свобод человека и гражданина в информационной сфере;
- отражают особенности применения публично-правовых и гражданско-правовых методов правового регулирования при осуществлении информационных прав и свобод с учетом специфических особенностей и юридических свойств информации и информационных объектов.

Глобальной целью осуществления информационных прав и свобод, достигаемых посредством норм информационного права, можно считать создание условий для формирования гармонической и высокоинтеллектуальной личности, построения свободного и демократического общества и государства, обладающего информационным суверенитетом.

Для гражданского права используется метод диспозитивного регулирования:
- равенство субъектов правоотношений, выражающееся, прежде всего, в их свободной волевой ориентации и независимости своей воли;
- самостоятельность участников правоотношений и свободное осуществление ими своих прав;
- самостоятельность субъектов правоотношений в смысле ответственности по обязательствам.

Для публичного права используется императивный метод, для которого характерны централизованное осуществление властных полномочий и строгая субординация участников правоотношений.

Публично-правовой аспект информационных отношений объясняется необходимостью обеспечения гарантий осуществления информационных конституционных прав и свобод граждан, государственного управления информационными процессами формирования и использования государственных информационных ресурсов, создания и применения государственных информационных систем и средств их обеспечения, а также средств и механизмов информационной безопасности для достижения главной цели — обеспечение гарантий осуществления информационных прав и свобод.
Основными субъектами в информационных отношениях публично-правового порядка в информационной сфере выступают органы государственной власти и местного самоуправления, исполняющие обязанности по информационному обеспечению физических и юридических лиц. Для органов государственной власти и местного самоуправления участие в информационных правоотношениях является их прямой юридической обязанностью, т.к. она является главным средством практической реализации установленной для них компетенции, а отсюда и правоспособности.

##
###

74. Стандарт ISO 27000

ISO 27001 — это международный стандарт, разработанный Международной организацией по стандартизации, который описывает, как управлять информационной безопасностью в компании. Последняя версия этого стандарта была опубликована в 2013 году, и его полное название на сегодня — «ISO/IEC 27001:2013» (не переведена). Первая версия стандарта, опубликованная в 2005 году, базировалась на Британском стандарте BS 7799-2.
Стандарт ISO 27001 сосредоточен на защите конфиденциальности, сохранности и доступности информации в компании. Это реализуется путём выяснения потенциальных проблем с информацией (т.е. оценки рисков), а затем определения необходимых шагов для предотвращения появления таких проблем (т.е. снижения или обработки рисков). Поэтому основная философия ISO 27001 базируется на управлении рисками: выяснить, где находятся риски, а затем систематически обрабатывать их.
Защитные меры (или контроли), которые должны внедряться, обычно выступают в форме политик, процедур и технического внедрения (например, программного обеспечения и оборудования). Однако, в большинстве случаев, компании уже располагают у себя всем

оборудованием и программным обеспечением. Однако используют они их небезопасным способом, поэтому большая часть внедрений ISO 27001 будет связана с постановкой организационных правил (т.е. с написанием документов), которые необходимы для предотвращения нарушений в системе безопасности. Поскольку такое внедрение потребует управления множеством политик, процедур, людей, активов и т.д., в ISO 27001 описано, как состыковать вместе все эти элементы в системе менеджмента информационной безопасности. Поэтому управление информационной безопасностью — это не только ИТ-безопасность (т.е. фаерволы, антивирус и т.д.). Это также управление процессами, правовая защита, управление персоналом, физическая защита и т.д.

ISO/IEC 27001 разбит на 11 разделов плюс приложение A:
1) Введение – объясняет цель стандарта ISO 27001 и его совместимость с другими стандартами управления.
2) Область применения – объясняет, что этот стандарт применим в организации любого типа.
3) Нормативные ссылки – относятся к ISO/IEC 27000, как к стандарту, в котором даны термины и определения.
4) Термины и определения – опять же относятся к ISO/IEC 27000.
5) Особенности организации – Этот раздел является частью фазы планирования в цикле «Планирование, реализация, контроль, корректировка» и определяет требования для понимания внешних и внутренних проблем, заинтересованных сторон и их требований, а также для определения области применения системы менеджмента информационной безопасности.
6) Ответственность руководства – Этот раздел является частью фазы планирования в цикле «Планирование, реализация, контроль, корректировка» и определяет обязанности топ-менеджмента, определяет роли и обязанности, а также содержание политики информационной безопасности верхнего уровня.
7) Планирование – Этот раздел является частью фазы планирования в цикле «Планирование, реализация, контроль, корректировка» и определяет требования для оценки рисков, обработки рисков, заявления о применимости, плана по обработке рисков и постановки задач для информационной безопасности.
8) Поддержка – Этот раздел является частью фазы планирования в цикле «Планирование, реализация, контроль, корректировка» и определяет требования к доступности ресурсов, компетенций, информированности, коммуникации и контролю документации и записей.
9) Функционирование – Этот раздел является частью фазы реализации в цикле «Планирование, реализация, контроль, корректировка» и определяет внедрение оценки и обработки рисков, а также контролей и других процессов, необходимых для достижения целей информационной безопасности.
10) Оценка эффективности – Этот раздел является частью фазы проверки в цикле «Планирование, реализация, контроль, корректировка» и определяет требования к мониторингу, измерению, анализу, оценке, внутреннему аудиту и анализу управления.
11) Усовершенствование – Этот раздел является частью фазы корректировки в цикле «Планирование, реализация, контроль, корректировка» и определяет требования к несоответствиям, исправлениям, корректирующим действиям и непрерывному совершенствованию.
Приложение A – Это приложение содержит каталог из 114 контролей (защитных мер).

##
###

75. Стандарт BSI (Германия). Федеральные критерии безопасности информационных технологий (США). Международный стандарт COBIT

Германское «Руководство по защите информационных технологий для базового уровня защищенности» 1998 года посвящено детальному рассмотрению частных вопросов создания политик безопасности компании и управления безопасностью в целом.
В германском стандарте BSI представлены:
- общая методика разработки политик безопасности и управления информационной безопасностью в целом (организация менеджмента в области информационной безопасности, методология использования руководства);
- описания компонентов современных информационных технологий (организационный уровень ИБ, процедурный уровень, организация защиты данных, планирование действий в ЧС);
- описания основных компонентов организации режима информационной безопасности (организационный и технический уровни защиты данных, планирование действий в чрезвычайных ситуациях, поддержка непрерывности бизнеса);
- характеристики объектов информатизации (здания, помещения, кабельные сети, контролируемые зоны);
- характеристики основных информационных активов компании (в том числе аппаратное и программное обеспечение, например рабочие станции и серверы под управлением операционных систем семейства DOS, Windows и UNIX);
- характеристики компьютерных сетей на основе различных сетевых технологий, например сети Novell NetWare, сети UNIX и Windows;
- характеристика активного и пассивного телекоммуникационного оборудования ведущих вендоров, например Cisco Systems;
- подробные каталоги угроз безопасности и мер контроля (более 600 наименований в каждом каталоге).

Существенно, что политики безопасности компании рассматриваются по определенному сценарию: общее описание информационного актива компании - возможные угрозы и уязвимости безопасности - возможные меры и средства контроля и защиты.

Документ «Федеральные критерии безопасности информационных технологий» (далее «Федеральные критерии») представляет собой базу для разработки и сертификации компонентов информационных технологий с точки зрения обеспечения безопасности. «Федеральные критерии» охватывают практически полный спектр проблем, связанных с защитой и обеспечением безопасности, так как включают все аспекты обеспечения конфиденциальности, целостности и доступности.
Основными объектами применения требований безопасности «Федеральных критериев» являются продукты информационных технологий и системы обработки информации. Под продуктом информационных технологий (далее – ПИТ) понимается совокупность аппаратных и программных средств, которая представляет собой поставляемое конечному потребителю готовое к использованию средство обработки информации.
Положения «Федеральных критериев» касаются только собственных средств обеспечения безопасности ПИТ, т.е. механизмов защиты, встроенных непосредственно в эти продукты в виде соответствующих программных, аппаратных или специальных средств. Для повышения их эффективности могут дополнительно применяться внешние системы защиты и средства обеспечения безопасности, к которым относятся как технические средства, так и организационные меры, правовые и юридические нормы.
Ключевым понятием концепции информационной безопасности «Федеральных критериев» является понятие «профиля защиты». Профиль защиты – это нормативный документ, который регламентирует все аспекты безопасности ПИТ в виде требований к его проектированию, технологии разработки и сертификации. Как правило, один профиль

защиты описывает несколько близких по структуре и назначению ПИТ. Основное внимание в профиле защиты уделяется требованиям к составу средств защиты и качеству их реализации, а также их адекватности предполагаемым угрозам безопасности.
«Федеральные критерии» представляют процесс разработки систем обработки информации, начинающийся с формулирования требований потребителями и заканчивающийся введением в эксплуатацию, в виде следующих основных этапов:

- разработка и анализ профиля защиты. Требования, изложенные в профиле защиты, определяют функциональные возможности ПИТ по обеспечению безопасности и условия эксплуатации, при соблюдении которых гарантируется соответствие предъявляемым требованиям. Кроме требований безопасности, профиль содержит требования по соблюдению технологической дисциплины в процессе разработки, тестирования, анализа и сертификации ПИТ.
- разработка и сертификация ПИТ. Разработанные ПИТ подвергаются независимому анализу, целью которого является определение степени соответствия характеристик продукта сформулированным в профиле защиты требованиям и спецификациям.
- компоновка и сертификация системы обработки информации в целом. Успешно прошедшие второй этап ПИТ интегрируются в систему обработки информации. Полученная в результате система должна удовлетворять заявленным в профиле защиты требованиям при соблюдении указанных в нем условий эксплуатации.

«Федеральные критерии» регламентируют только первый этап этой схемы – разработку и анализ профиля защиты; процесс создания ПИТ и компоновка систем обработки информации остаются вне рамок этого стандарта.

COBIT («Задачи управления для информационных и смежных технологий») — методология управления информационными технологиями, принадлежащая и разрабатываемая некоммерческой организацией ISACA. Представляет собой пакет открытых документов, около 40 международных и национальных стандартов и руководств в области управления IT, аудита и IT-безопасности, основанных на анализе и гармонизации существующих стандартов и ведущих практик в области управления IT.
Задача COBIT заключается в ликвидации разрыва между руководством компании с их видением бизнес-целей и IT-департаментом, осуществляющим поддержку информационной инфраструктуры, которая должна способствовать достижению бизнес-целей. Нередко руководство компании в силу объективных причин не понимает IT-специалистов. По представлению руководства, сотрудники IT-подразделения разговаривают на каком-то птичьем языке. Те, в свою очередь, не понимают бизнес-терминов, на основании которых строятся распоряжения руководства. Всё это приводит к росту издержек, выполнению лишней работы, что, конечно же, сказывается на эффективности деятельности компании. COBIT, благодаря единой терминологии, служит своеобразной платформой-буфером для конструктивного диалога между всеми участниками бизнеса.
В COBIT детально описаны цели и принципы управления, объекты управления, чётко определены все IT-процессы (задачи), протекающие в компании, и требования к ним, описан возможный инструментарий (практики) для их реализации. В описании IT-процессов также приведены практические рекомендации по управлению IT-безопасностью.
Кроме того, COBIT вводит целый ряд показателей (метрик) для оценки эффективности реализации системы управления IT, которые часто используются аудиторами IT-систем. В их число входят показатели качества и стоимости обработки информации, характеристики её доставки получателю, показатели, относящиеся к субъективным аспектам обработки информации (например стиль, удобство интерфейсов).
COBIT позволяет связать бизнес-цели с непосредственными IT — процессами, оценивать текущее состояние процессов управления IT, определять направления для

совершенствования бизнеса. Оцениваются показатели, описывающие соответствие компьютерной IT-системы принятым стандартам и требованиям, достоверность обрабатываемой в системе информации, её действенность, общепринятые показатели информационной безопасности — конфиденциальность, целостность и доступность обрабатываемой в системе информации.
Управление IT по COBIT можно представить в следующем ступенчатом виде (по порядку реализации):
- стратегии;
- политики;
- стандарты;
- процедуры.

При разработке стандарта была заложена возможность использования его как для проведения аудита IT-системы компании, так и для проектирования IT-системы.

##
###

76. Общие требования по защите информации, предусмотренные РД и СТР-К ФСТЭК России.

РД “СВТ. Защита от НСД к информации. Показатели защищенности от НСД к информации”
Устанавливает классификацию СВТ (средств вычислительной техники) по уровню защищенности от НСД к информации на базе перечня показателей защищенности и совокупности описывающих их требований. Устанавливается 7 классов защищенности СВТ от НСД к информации. Самый низкий 7, самый высокий 1. Классы делятся на 4 группы, отличаются уровнем защиты:
- 1 группа содержит только 7 класс;
- 2 группа характеризуется дискреционной защитой и содержит 6 и 5 классы;
- 3 группа - мандатной защитой и содержит 4, 3 и 2 классы;
- 4 группа - верифицированной защитой, только 5 класс.

РД “АС. Защита от НСД к информации. Классификация АС и требования по защите информации”
Устанавливает классификацию АС, подлежащих защите от НСД к информации, и требования по ЗИ в АС различных классов. АС группируются в классы по признакам:
- наличие в АС информации различного уровня конфиденциальности;
- уровень полномочий субъектов доступа АС на доступ к конфиденциальной информации;
- режим обработки данных в АС - коллективный или индивидуальный.

Всего 9 классов защищенности АС от НСД к информации. Каждый класс характеризуется определенной минимальной совокупностью требований по защите. Классы делятся на 3 группы, отличающиеся особенностями обработки информации в АС. В пределах каждой группы соблюдается иерархия требований по защите в зависимости от ценности и конфиденциальности информации и, следовательно, иерархия классов защищенности АС.

РД "СВТ. МЭ. Защита от НСД к информации. Показатели защищенности от НСД к информации"
Используется при анализе системы защиты внешнего периметра корпоративной сети в качестве основных критериев. Определяет показатели защищенности МЭ. Каждый

показатель защищенности представляет собой набор требований безопасности, характеризующих определенную область функционирования МЭ.
Всего 5 показателей:
- управление доступом;
- идентификация и аутентификация;
- регистрация событий и оповещение;
- контроль целостности;
- восстановление работоспособности.

На основании показателей защищенности определяются классы защищенности МЭ:
- простейшие фильтрующие маршрутизаторы - 5;
- пакетные фильтры сетевого уровня - 4;
- простейшие МЭ прикладного уровня - 3;
- МЭ базового уровня - 2;
- продвинутые МЭ - 1.

МЭ 1 класса защищенности могут использоваться в АС класса 1А, обрабатывающих информацию “Особой важности”.
2 класс защищенности МЭ соотвует классу защищенности АС 1Б, предназначенной для обработки “Совершенно секретной” информации и т. п.

В настоящее время описанные РД уже устарели и содержащаяся в них классификация АС, СВТ и МЭ не может признаваться состоятельной. Достаточно заметить, что классификация АС и СВТ, разрабатывалась без учета распределенной (сетевой) природы современных АС, а все современные коммерческие МЭ по своим возможностям существенно превосходят требования 1-го класса защищенности (за исключением требования по использованию сертифицированных криптографических алгоритмов).
Развитием нормативной базы в этом направлении является разработка “Профилей защиты” для различных классов СВТ, АС и МЭ на базе “Общих критериев”.
Значительные усилия в этом направлении предпринимаются под эгидой Гостехкомиссии России.
Проект РД Гостехкомиссии России “Специальные требования и рекомендации по защите конфиденциальной информации” (СТР-К) содержит достаточно полный набор требований и рекомендаций организационного уровня по защите речевой информации, информации, обрабатываемой средствами вычислительной техники, а также по защите информации при подключении к сетям общего пользования.
В документе рассматриваются:
- ЗИ на рабочих местах на базе автономных ПЭВМ;
- ЗИ при использовании съемных накопителей большой емкости для автоматизированных рабочих мест на базе автономных ПЭВМ;
- ЗИ в ЛВС;
- ЗИ при межсетевом взаимодействии;
- ЗИ при работе с СУБД.

СТР-К может использоваться при проведении аудита безопасности АС для оценки полноты и правильности реализации организационных мер защиты информации в АС.

##
###

77. Общие нормативные требования по защите персональных данных

Федеральный закон РФ № 152-ФЗ «О персональных данных» от 27.07.2006 регулирует отношения, связанные с обработкой персональных данных, осуществляемой

государственными и муниципальными органами, органами местного самоуправления, физическими и юридическими лицами с использованием средств автоматизации
В тексте:
- определены принципы и условия обработки персональных данных;
- предусматриваются случаи, когда согласие на обработку персональных данных не требуется;
- регулируются отношения по обработке специальных категорий персональных данных (сведения о расовой, национальной принадлежности, политических взглядах, религиозных или философских убеждениях, состоянии здоровья, интимной жизни);
- определяется важнейшая гарантия прав субъекта персональных данных, которой является обязанность операторов и третьих лиц, получивших доступ к персональным данным, обеспечивать их конфиденциальность;
- устанавливаются принципы трансграничной передачи данных, при которой должна обеспечиваться адекватная защита прав субъектов персональных данных.

Цель закона - обеспечение защиты прав и свобод человека и гражданина при обработке его ПДн, в том числе защиты прав на неприкосновенность частной жизни, личную и семейную тайну (Конституция РФ, статья 23). Под ПДн понимается любая информация, относящаяся к определенному или определяемому на основании такой информации физическому лицу (субъекту ПДн), в том числе его фамилия, имя, отчество, год, месяц, дата и место рождения, адрес, семейное, социальное, имущественное положение, образование, профессия, доходы, другая информация. Закон определяет принципы и условия обработки ПДн. Субъект ПДн принимает решение о предоставлении своих ПДн и дает согласие на их обработку своей волей и в своих интересах, кроме случаев, определенных законом.
Законом определяются специальные категории ПДн: данные, касающиеся расовой, национальной принадлежности, политических взглядов, религиозных или философских убеждений, состояния здоровья, интимной жизни. Обработка таких данных не допускается, кроме случаев, описанных в законе. Субъект имеет право на получение сведений об операторе, о месте его нахождения, о наличии у оператора ПДн, относящихся к соответствующему субъекту ПДн, а также на ознакомление с такими ПДн, за исключением случаев, описанных в законе.
Определена ответственность за нарушение требований закона о ПДн: лица, виновные в нарушении требований настоящего закона, несут гражданскую, уголовную, административную, дисциплинарную и иную ответственность, предусмотренную законодательством РФ.

Документ предписывает ФСБ и ФСТЭК утвердить в пределах своей компетенции нормативные правовые акты и методические документы, необходимые для выполнения требований.
ФСТЭК:
1) Базовая модель угроз безопасности ПДн при их обработке в информационных системах ПДн (=ИСПДн).
2) Методика определения актуальных угроз безопасности ПДн при их обработке в ИСПДн.
3) Основные мероприятия по организации и техническому обеспечению безопасности ПДн, обрабатываемых в ИСПДн.
4) Рекомендации по обеспечению безопасности ПДн при их обработке в ИСПДн.

Для построения системы защиты ПДн необходимо:
1) Определить класс ИСПДн
- для типовых ИСПДн: класс определяется по связке объем-категория;

- для специальных ИСПДн (=содержат ПДн о здоровье или на основе обработки ПДн принимаются юридические решения относительно субъекта ПДн) – класс определяется после построения модели угроз.

2) Построить модель угроз
3) Определить актуальные угрозы
4) Защитить систему по соответствующим требования

Для определения класса ИСПДн:

- Категории ПДн

1) Персональные данные, касающиеся расовой, национальной принадлежности, политических взглядов, религиозных и философских убеждений, состояния здоровья, интимной жизни.

2) Персональные данные, позволяющие идентифицировать субъекта персональных данных и получить о нем дополнительную информацию, за исключением персональных данных, относящихся к категории 1.

3) Персональные данные, позволяющие идентифицировать субъекта персональных данных;

4) Обезличенные и (или) общедоступные персональные данные.

- Объем обрабатываемых ПДн

1) В информационной системе одновременно обрабатываются персональные данные более чем 100 000 субъектов персональных данных или персональные данные субъектов персональных данных в пределах субъекта Российской Федерации или Российской Федерации в целом.

2) В информационной системе одновременно обрабатываются персональные данные от 1000 до 100 000 субъектов персональных данных или персональные данные субъектов персональных данных, работающих в отрасли экономики Российской Федерации, в органе государственной власти, проживающих в пределах муниципального образования.

3) В информационной системе одновременно обрабатываются данные менее чем 1000 субъектовперсональных данных или персональные данные субъектов персональных данных в пределах конкретной организации.

- Классы ИСПДн

1) Класс 1 (К1) – информационные системы, для которых нарушение заданной характеристики безопасности ПДн, обрабатываемых в них, может привести к значительным негативным последствиям для субъектов ПДн.

2) Класс 2 (К2) – информационные системы, для которых нарушение заданной характеристики безопасности ПДн, обрабатываемых в них, может привести к негативным последствиям для субъектов ПДн.

3) Класс 3 (К3) – информационные системы, для которых нарушение заданной характеристики безопасности ПДн, обрабатываемых в них, может привести к незначительным негативным последствиям для субъектов ПДн.

4) Класс 4 (К4) – информационные системы, для которых нарушение заданной характеристики безопасности ПДн, обрабатываемых в них, не приводит к негативным последствиям для субъектов ПДн.

Категория ПДн	<1000	1000 - 100 000	>100 000
4	К4	К4	К4
3	К3	К3	К2

2	К3	К2	К1
1	К1	К1	К1

Определение актуальности угроз:

Возможность реализации	Показатель опасности		
	Низкий	Средний	Высокий
Низкая	не акт	не акт	акт
Средняя	не акт	акт	акт
Высокая	акт	акт	акт
Очень высокая	акт	акт	акт

ФСБ:

1) Методические рекомендации по обеспечению с помощью криптосредств безопасности ПДн при их обработке в ИСПДн с использованием средств автоматизации.

2) Описана методология формирования модели угроз. Различают модель угроз верхнего уровня и детализированную модель угроз.

3) Определены требования к контролю встраивания криптосредств. Типовые требования по организации и обеспечению функционирования криптосредств, предназначенных для защиты информации (=ЗИ), не содержащей сведений, составляющих государственную тайну в случае их использования для обеспечения безопасности ПДн при их обработке в ИСПДн.

3) Определяют порядок организации и обеспечения функционирования шифровальных средств, предназначенных для ЗИ, не содержащей сведений, составляющих государственную тайну в случае их использования для обеспечения безопасности ПДн при их обработке в ИСПДн.

4) Описаны мероприятия по организации и обеспечению безопасности обработки ПДн с использованием криптосредств.

5) Описан порядок обращения с криптосредствами и криптоключами к ним. Указаны правила хранения, учета, использования и пересылки криптосредств, а также способы утилизации. Итог защиты – аттестация (для 1 и 2 класса – обязательна, для 3 – по усмотрению оператора, для 4 - нет).

6) ПДн отнесены к конфиденциальной информации (Указ Президента РФ «Об утверждении перечня сведений конфиденциального характера»).

СТР-К:
1) Основными доками для защиты ПДн являются методички ФСБ и ФСТЭК, написанные на основе СТР-К
2) Устанавливает порядок организации работ, требования и рекомендации по обеспечению технической защиты конфиденциальной информации на территории РФ.
3) Описаны основные требования и рекомендации по защите ПДн. АС, обрабатывающие ПДн, должны быть отнесены по уровню защищенности к классам 3Б, 2Б и не ниже 1Д.
4) Для передачи информации по каналам связи, выходящим за пределы КЗ, необходимо использовать защищенные каналы связи, в том числе защищенные волоконно-оптические линии связи или предназначенные для этого криптосредства ЗИ. Применяемые средства должны быть сертифицированы.

##
###

78. Алгоритмы блочного шифрования. ГОСТ 34.12-2015

Блочный шифр — разновидность симметричного шифра. Его особенностью является обработка блока нескольких байт за одну итерацию. Блочные криптосистемы разбивают текст сообщения на отдельные блоки фиксированного размера и затем осуществляют преобразование этих блоков с использованием ключа.

Симметричная схема шифрования:
M – сообщение, открытый текст, C – закрытый текст, K – ключ шифрования,
E – функция шифрования, D – функция расшифрования.
Шифрование: C=E(K,M)
Расшифрование: M=D(K,C)

Отечественный стандарт: ГОСТ 34.12-2015
Западный стандарт: AES

Свойства современных блочных шифров:
- лавинный эффект – нарастающая потеря соответствия битов между блоками открытых и зашифрованных данных;
- высокая скорость;
- простота аппаратной реализации;
- меньшая длина ключа при большой стойкости;
- изученность.

Недостатки блочных шифров:
- сложность управления ключами;
- проблема обмена ключами.

Режим шифрования — метод применения блочного шифра, позволяющий преобразовать последовательность блоков открытых данных в последовательность блоков зашифрованных данных.
Стандарт ГОСТ Р 34.13-2015 "Режимы работы блочных шифров" определяет следующие режимы шифрования (<+> == исключающее "или"):
1) Режим простой замены (Electronic Code Book, ECB)

В режиме ECB шифрование/дешифрование i-го блока открытого текста/шифротекста выполняется независимо от остальных блоков:

Ci = Ek(Mi), Mi=Dk(Ci)

Недостатком данного режима шифрования является то, что одинаковые блоки входного текста будут кодироваться в одинаковые блоки шифротекста, что дает возможность злоумышленнику, во-первых, делать предположения о характере информации в открытом тексте, а во-вторых, подменить один или несколько блоков шифротекста. Достоинством режима можно назвать простоту реализации, а также возможность распараллеливания процедуры шифрации.

2) Режим гаммирования (Counter, CTR)

Для этого режима формируется специальная гамма, которая представляет собой случайную последовательность бит. Гамма побитово суммируется по модулю 2 с блоками открытого текста. Гамма получается следующим образом: с помощью алгоритмического рекуррентного генератора последовательности чисел (РГПЧ) вырабатываются 64-битовые блоки данных, которые далее подвергаются зашифрованию в режиме простой замены, в результате чего получаются блоки гаммы. Благодаря тому, что наложение и снятие гаммы осуществляется при помощи одной и той же операции побитового исключающего или, алгоритмы зашифрования и расшифрования в режиме гаммирования идентичны.

3) Режим гаммирования с обратной связью по выходу (Output Feedback, OFB)

В режиме OFB исходное сообщение вообще не подвергается криптопреобразованию, оно складывается с шифруемыми на секретном ключе блоками Si (S0 является задаваемым несекретным параметром режима):

Ci = Mi <+> Si, Mi = Ci <+> Si, Si = Ek(Si-1)

В этом режиме, как и в режиме ECB, ошибки, которые могут возникнуть при передаче шифротекста по каналам связи, локализуются в блоке, не распространяясь на соседние, причем в режиме OFB ошибочными будут только биты, подвергшиеся изменению (в ECB изменится весь блок). Это дает возможность злоумышленнику незаметно для принимающей стороны подменить блок шифротекста. Возможности распараллеливания процедур шифрации/дешифрации затруднены.

4) Режим простой замены с зацеплением (Cipher Block Changing, CBC)

Режим CBC предполагает следующие алгоритмы шифрации/дешифрации:

Ci = Ek(Mi <+> Ci-1), Mi = Dk(Ci) <+> Ci-1

В режиме CBC каждый блок открытого текста складывается с блоком шифротекста, полученным на предыдущем этапе. Таким образом, происходит сцепление блоков друг с другом и независимая манипуляция с каждым из них невозможна, а одинаковые входные блоки будут давать на выходе разные блоки. Однако, задача распараллеливания процедуры кодирования в этом режиме затруднена. Дополнительным параметром процедур шифрования/дешифрования является параметр C0.

5) Режим гаммирования с обратной связью по шифротексту (Cipher Feedback, CFB)

В режиме CFB также происходит «маскировка» блока открытого текста уже зашифованными блоками:

Ci = Mi <+> Ek(Ci-1), Mi = Dk(Ci-1) <+> Ci

По своим возможностям данный режим похож на режим CBC, но если длина сообщения не кратна размеру блока шифра, то в режиме CBC необходимо дополнять последний блок дополнительными битами и сообщать на принимающую сторону истинный размер сообщения, а режим CFB позволяет сформировать шифротекст того же размера, что и исходное сообщение.

6) Режим выработки имитовставки (Message Authentication Code algorithm)

Режим выработки имитовставки предназначен для обнаружения случайных и преднамеренных ошибок при передаче шифрованных данных потребителям и одинаков для любого из режимов шифрования открытых данных. Имитовставка представляет собой

дополнительный блок данных из L бит, который формируется либо перед шифрованием всего сообщения, либо совместно с шифрованием по блокам.

ГОСТ Р 34.12-2015 (кратко и с картинками):
https://habr.com/post/266359/

##
###

79. Алгоритмы шифрования с открытым ключом. Алгоритм RSA.

Суть шифрования с открытым ключом заключается в том, что для шифрования данных используется один ключ, а для расшифрования другой (поэтому такие системы часто называют асимметричными).
Чтобы гарантировать надежную защиту информации, к криптосистемам с открытым ключом предъявляются два важных и очевидных требования:
- преобразование исходного текста должно быть условно необратимым и исключать его восстановление на основе открытого ключа;
- определение закрытого ключа на основе открытого также должно быть невозможным на современном технологическом уровне.

Алгоритмы с открытым ключом разрабатывались для решения двух наиболее трудных задач, возникших при использовании симметричного шифрования:
1) Распределение ключа. Так как в симметричном шифровании используется один общий ключ, он должен быть передан обеим сторонам. При этом ключ должен оставаться засекреченным.
2) Невозможность подмены одного из участников или цифровая подпись. В электронной коммерции необходим аналог подписи, содержащейся в бумажных документах. Цифровая подпись позволяет подтвердить, что сообщение было послано конкретным участником.

Алгоритм RSA описан в вопросе №85.

##
###

80. Криптографические хеш-функции. ГОСТ Р 34.11-2012

Хеширование — преобразование массива входных данных произвольной длины в (выходную) битовую строку установленной длины, выполняемое определённым алгоритмом. Функция, воплощающая алгоритм и выполняющая преобразование, называется «хеш-функцией» или «функцией свёртки».
Криптографические хеш-функции — это выделенный класс хеш-функций, который имеет определенные свойства, делающие его пригодным для использования в криптографии.

Идеальной криптографической хеш-функцией является такая криптографическая хеш-функция, к которой можно отнести пять основных свойств:
- детерминированность. При одинаковых входных данных результат выполнения хеш-функции будет одинаковым (одно и то же сообщение всегда приводит к одному и тому же хешу);

- высокая скорость вычисления значения хеш-функции для любого заданного сообщения;
- невозможность сгенерировать сообщение из его хеш-значения, за исключением попыток создания всех возможных сообщений;
- наличие лавинного эффекта. Небольшое изменение в сообщениях должно изменить хэш-значения, так широко, что новые хэш-значения не совпадают со старыми хэш-значениями;
- невозможность найти два разных сообщения с одинаковыми хеш-значениями.

Атака «дней рождения» — используемое в криптоанализе название для метода поиска коллизий хеш-функций на основе парадокса дней рождения. Суть парадокса в том, что в группе, состоящей из 23 или более человек, вероятность совпадения дней рождения (число и месяц) хотя бы у двух людей превышает 50 %. Например, если в классе 23 ученика или более, то более вероятно то, что у кого-то из одноклассников дни рождения придутся на один день, чем то, что у каждого будет свой неповторимый день рождения.

ГОСТ Р 34.11-2012 определяет алгоритм и процедуру вычисления хэш-функции для любой последовательности двоичных символов, которые применяются в криптографических методах обработки и защиты информации, в том числе для реализации процедур обеспечения целостности, аутентичности, электронной цифровой подписи (ЭЦП) при передаче, обработке и хранении информации в автоматизированных системах.
Функция хэширования ГОСТ Р 34.11-2012 используется при реализации систем электронной цифровой подписи на базе ассиметричного криптографического алгоритма по ГОСТ Р 34.10–2012.

Внутреннее устройство "Стрибог":
Семейство хэш-функций Стрибог состоит из двух хэш-функций с длинами результирующего значения в 256 и 512 бит, которые отличаются начальным внутренним состоянием и его частью, принимаемой за результат вычислений.
Устройство новой хэш-функции во многом следует старому стандарту. Входное сообщение разбивается на блоки фиксированного размера, для сообщений размером не кратным длине блока используется дополнение. Начальное внутреннее состояние хэш-функции обновляется последовательной обработкой блоков сообщения функцией сжатия. Параллельно с этим вычисляются число обработанных бит и контрольная сумма блоков. После всех блоков сообщения функция сжатия обрабатывает блок с общей длиной сообщения и блок с контрольной суммой для завершения вычисления значения хэш-функции. Размер блоков сообщения и внутреннего состояния хэш-функции составляет 512 бит.
Основное отличие хэш-функции Стрибог от своего предшественника — функция сжатия.

Функция сжатия:
Основная операция функции сжатия обозначается как LPS и состоит из трёх преобразований: подстановки на байтах, транспонирования матрицы байт и умножения 64-битных векторов на матрицу 64 × 64 в GF(2):
1) S — нелинейная биекция. 512 бит аргумента рассматриваются как массив из шестидесяти четырёх байт, каждый из которых заменяется по заданной стандартом таблице подстановки;
2) P — переупорядочивание байт. Байты аргумента меняются местами по определённому в стандарте порядку;

3) L — линейное преобразование. Аргумент рассматривается как 8-мь 64-битных векторов, каждый из которых заменяется результатом умножения на определённую стандартом матрицу 64 × 64 над GF(2).

Переупорядочивание байт P, определённое стандартом, является операций транспонирования матрицы байт размером 8 × 8.
В функции сжатия используются только преобразование LPS и побитовое исключающее ИЛИ над 512-битными блоками. Вместе со сложением по модулю 2^512 они составляют полный набор операций, использующихся в функции хеширования ГОСТ Р 34.11-2012.
Значение функции сжатия на каждом шаге зависит от предыдущего шага, в связи с чем невозможно обрабатывать блоки одного потока данных параллельно. Это свойство не является особенностью Стрибога, а присуще многим хэш-функциям.

Статья про "Стрибог" на Хабре:
https://habr.com/post/188152/

##
###

81. Электронная цифровая подпись. ГОСТ Р 34.10-2012

Электронная цифровая подпись — реквизит электронного документа, полученный в результате криптографического преобразования информации с использованием закрытого ключа подписи и позволяющий проверить отсутствие искажения информации в электронном документе с момента формирования подписи (целостность), принадлежность подписи владельцу сертификата ключа подписи (авторство), а в случае успешной проверки подтвердить факт подписания электронного документа (неотказуемость).
Применяется при совершении гражданско-правовых сделок, оказании государственных и муниципальных услуг, исполнении государственных и муниципальных функций, при совершении иных юридически значимых действий. В России юридически значимый сертификат электронной подписи выдаёт удостоверяющий центр. Правовые условия использования электронной цифровой подписи в электронных документах регламентирует Федеральный закон Российской Федерации от 6 апреля 2011 года № 63-ФЗ «Об электронной подписи».
Поскольку подписываемые документы — переменного (и как правило достаточно большого) объёма, в схемах ЭП зачастую подпись ставится не на сам документ, а на его хэш.

Параметры схемы цифровой подписи, алгоритмы её формирования и проверки:
https://ru.wikipedia.org/wiki/ГОСТ_Р_34.10-2012

Криптографическая стойкость данной схемы цифровой подписи основывается на сложности решения задачи дискретного логарифмирования в группе точек эллиптической кривой, а также на стойкости используемой хэш–функции.

##
###

82. Криптографический генератор псевдослучайных чисел.

Большинство современных криптографических приложений используют случайные числа. Они нужны для генерации ключей, получения одноразовых случайных чисел, создания соли и т. д. Если случайные числа будут небезопасными, то это влечёт за собой появление уязвимостей в приложениях, которые невозможно закрыть с помощью различных алгоритмов и протоколов.

Генератор псевдослучайных чисел — алгоритм, порождающий последовательность чисел, элементы которой почти независимы друг от друга и подчиняются заданному распределению (обычно равномерному).
Криптографически стойкий генератор псевдослучайных чисел — это генератор псевдослучайных чисел с определёнными свойствами, позволяющими использовать его в криптографии.

Требования к обычному генератору псевдослучайных чисел выполняются и криптографически стойким ГПСЧ, обратное неверно. Требования к КСГПСЧ можно разделить на две группы: во-первых, они должны проходить статистические тесты на случайность; а во-вторых, они должны сохранять непредсказуемость, даже если часть их исходного или текущего состояния становится известна криптоаналитику. А именно:
- КСГПСЧ должен удовлетворять «тесту на следующий бит». Смысл теста в следующем: не должно существовать полиномиального алгоритма, который, зная первые k битов случайной последовательности, сможет предсказать (k+1)-ый бит с вероятностью более 50 %;
- КСГПСЧ должен оставаться надёжным даже в случае, когда часть или все его состояния стали известны (или были корректно вычислены). Это значит, что не должно быть возможности получить случайную последовательность, созданную генератором, предшествующую получению этого знания криптоаналитиком.

Большинство генераторов псевдослучайных чисел не подходят для использования в качестве КСГПСЧ по обоим критериям. Во-первых, несмотря на то, что многие ГПСЧ выдают последовательность случайную с точки зрения разнообразных статистических тестов, они не надёжны по отношению к обратной разработке: могут быть обнаружены специализированные, особым образом настроенные тесты, которые покажут, что случайные числа, получаемые из ГПСЧ не являются по настоящему случайными. Во-вторых, для большинства ГПСЧ возможно вычислить всю псевдослучайную последовательность, если их состояние скомпрометировано, что позволит криптоаналитику получить доступ не только к будущим сообщениям, но и ко всем предыдущим. КСГПСЧ разрабатываются с учётом сопротивляемости к различным видам криптоанализа.

Классы реализации КСГПСЧ:
1) На основе криптографических алгоритмов
- безопасный блочный шифр можно преобразовать в КСГПСЧ, запустив его в режиме счетчика. Таким образом, выбрав случайный ключ, можно получать следующий случайный блок, применяя алгоритм к последовательным натуральным числам. Счет можно начинать с произвольного натурального числа. Очевидно, что безопасность такой схемы полностью зависит от секретности ключа;
- криптографически стойкая хеш-функция также может быть преобразована в КСГПСЧ. В таком случае исходное значение счетчика должно оставаться в секрете;
- большинство потоковых шифров работают на основе генерации псевдослучайного потока бит, которые некоторым образом комбинируется (почти всегда с помощью

операции XOR) с битами открытого текста. Запуск такого шифра на последовательности натуральных чисел даст новую псевдослучайную последовательность, возможно, даже с более длинным периодом. Такой метод безопасен только если в самом потоковом шифре используется надёжный КСГПСЧ (что не всегда так). Опять же, начальное состояние счётчика должно оставаться секретным.
2) На основе математических задач
- алгоритм Блюма — Блюма — Шуба имеет высокую криптостойкость, основанную на предполагаемой сложности факторизации целых чисел. Однако, этот алгоритм отличается очень медленной работой;
- алгоритм Блюма — Микали (англ. Blum-Micali algorithm) основан на задаче дискретного логарифма.
3) Специальные реализации
- алгоритм Ярроу
- /dev/random
- CryptGetRandom (CryptoAPI от Microsoft)
- ISAAC, базирующийся на RC4

Пример Российской разработки – CryptoPRO-CSP, у которой при генерации ключа необходимо двигать мышью и нажимать ее клавиши.

##
###

83. Протокол SSL

Отличная статья от ИТМО:
https://neerc.ifmo.ru/wiki/index.php?title=SSL/TLS

##
###

84. Протокол Kerberos

Kerberos — сетевой протокол аутентификации, позволяющий передавать данные через незащищённые сети для обеспечения безопасной идентификации. Ориентирован, в первую очередь, на клиент-серверную модель и обеспечивает взаимную аутентификацию — оба пользователя подтверждают личности друг друга через доверенный сервер. Данная модель является одним из вариантов протокола аутентификации Нидхема — Шрёдера на основе доверенной третьей стороны.

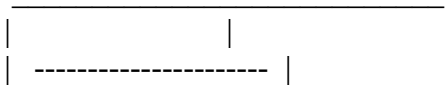
@Общие сведения:
- протокол Kerberos был специально разработан для того, чтобы обеспечить надежную аутентификацию пользователей;
- предусматривается, что начальный обмен информацией между клиентом и сервером происходит в незащищённой среде, а передаваемые пакеты могут быть перехвачены и модифицированы;
- протокол Kerberos может использовать централизованное хранение аутентификационных данных и является основой для построения механизмов Single Sign-On (возможность использования единой учетной записи пользователя для доступа к любым ресурсам области);

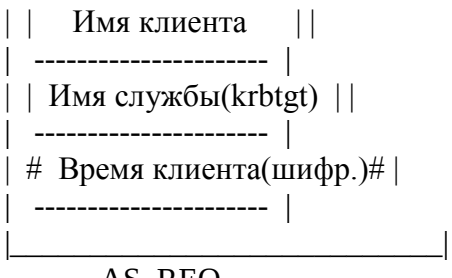
- протокол основан на понятии Ticket (билет). Ticket (билет) является зашифрованным пакетом данных, который выдается доверенным центром аутентификации, в терминах протокола Kerberos — Key Distribution Center (KDC, центр распределения ключей);
- когда пользователь выполняет первичную аутентификацию, после успешного подтверждения его подлинности KDC выдает первичное удостоверение пользователя для доступа к сетевым ресурсам — Ticket Granting Ticket (TGT). В дальнейшем, при обращении к отдельным ресурсам сети, пользователь, предъявляя TGT, получает от KDC удостоверение для доступа к конкретному сетевому ресурсу — Service Ticket (TGS);
- одним из преимуществ протокола Kerberos, обеспечивающим высокий уровень безопасности, является то, что при любых взаимодействиях не передаются ни пароли, ни значения хеша паролей в открытом виде;
- работая с протоколом Kerberos, необходимо, чтобы системные часы всех участвующих во взаимодействии узлов были синхронизированы;
- в качестве примера реализации протокола Kerberos имеет смысл отметить доменную аутентификацию пользователей в операционных системах Microsoft, начиная с Windows 2000.

@Процесс работы с протоколом Kerberos (кратко):
Пользователь регистрируется на своей рабочей станции, которая обрабатывает последовательность сообщений AS_REQ и AS_REP с центром KDC, откуда пользователь получает билет TGT, если учетные данные верны. Затем TGT пользователя кэшируется в памяти, и каждый раз, когда пользователю нужно получить доступ к службе (например, к серверу файлов, серверу печати, веб-приложению), пользователь предъявляет TGT центру KDC и запрашивает билет службы для конкретной службы. Пользователь получает билет службы и предъявляет его приложению, чтобы запросить доступ.
Схематично:

AS == Authentication Server
TGS == Ticket Granting Server
В процессее повествования отмечаю оба сервера как один - KDC, но на практике они могут рассматриваться как различные: в таком случае за процесс аутентификации отвечает сервер AS, а процесс выдачи билетов обеспечивает TGS. Просто заметка)

@Проверка подлинности:
- после того, как пользователь вводит имя и пароль на клиентской машине, эта машина хеширует введенный пароль. Полученный хеш становится секретным ключом клиента;
- для проверки подлинности введенных данных в KDC отправляется сообщение AS_REQ (Authentication Service Request). Для защиты от атаки с повторной передачей пакетов текущее время шифруется с использованием хеша пароля пользователя. Допустимое расхождение времени при этом (по умолчанию) - 5 минут. Элементы запроса AS_REQ представлены ниже:





AS_REQ

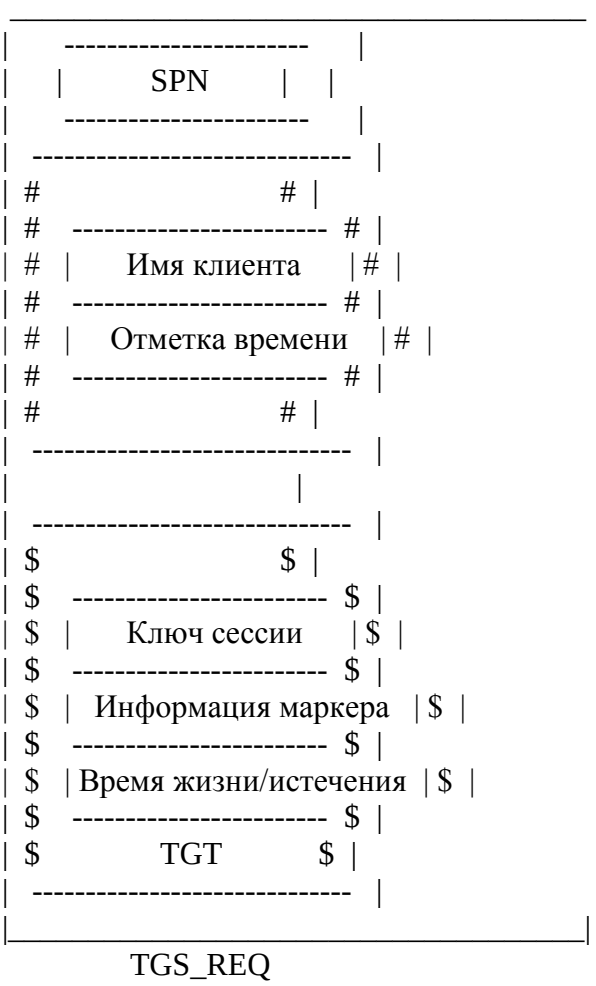
- когда KDC получает запрос AS_REQ, он в первую очередь пытается расшифровать отметку времени с использованием локальной копии хеша пароля пользователя. Если попытка заканчивается неудачей, клиент получает сообщение об ошибке, и обработка запроса прекращается. Если расшифрование происходит удачно И значение отметки времени находится в допустимых пределах, KDC отправляет пользователю сообщение AS_REP (Authentication Service Reply) со встроенным билетом TGT (Ticket Granting Ticket). Ответ AS_REP содержит в себе имя пользователя и два блока зашифрованных данных: первый блок шифруется с применением хеша пароля пользователя и содержит сеансовый ключ и отметку времени окончания существования билета (10 часов по умолчанию); а второй блок шифруется при помощи секрета KDC, который хранится в Active Directory как пароль для учетной записи krbtgt. Элементы ответа AS_REP представлены ниже:

AS_REP

Сеансовый ключ используется для шифрования будущих соединений с центром KDC. После получения AS_REP компьютер сохраняет в кэше билет TGT и сеансовый ключ на время существования TGT, а затем удаляет хеш пароля пользователя.

@Получение билета службы:

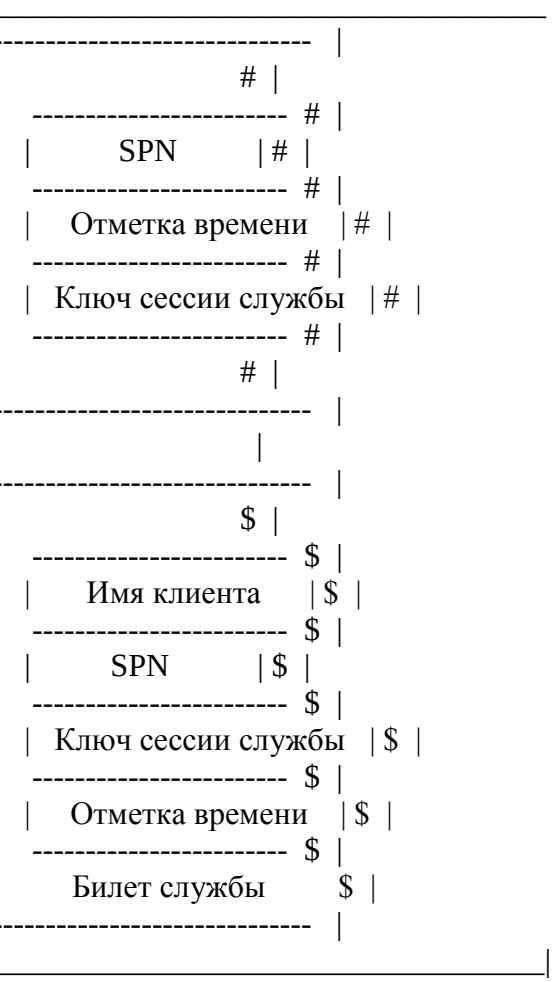
- в Kerberos любой объект, к которому требуется получить доступ, называется службой (например, серверы файлов и печати, серверы базы данных, внутренние веб-приложения). Для доступа к службе пользователь предоставляет билет службы. Перед этим компьютер или приложение пользователя определяет имя участника службы service principal name (SPN), к которой нужно получить доступ.
- для получения билета службы клиент обращается к KDC, отправляя ему запрос TGS_REQ (Ticket Granting Service Request). Первый фрагмент информации в запросе - имя SPN службы, для которой клиент запрашивает билет. Второй фрагмент - имя клиента и отметка текущего времени - шифруются с помощью сеансового ключа, полученного из AS_REP. Третий фрагмент - экземпляр билета TGT, полученного ранее также из AS_REP, зашифрованный при помощи секрета KDC. Элементы запроса TGS_REQ представлены ниже:



TGS_REQ

- после получения запроса TGS_REQ KDC проверяет, что указан один элемент для SPN, отметка времени находится в допустимом диапазоне и билет TGT не просрочен. Если все условия выполнены, то клиенту отправляется ответ TGS_REP, содержащий в себе зашифрованный билет службы. Первый блок TGS_REP шифруется при помощи сеансового ключа. Билет службы шифруется с помощью секрета службы (например, пароля учетной записи компьютера или учетной записи службы). Клиент кэширует билет службы и использует всегда, когда необходим доступ к службе. Так же, как у билетов TGT, время, в течение которого разрешено повторно использовать билеты службы, ограничено (десять часов по умолчанию в реализации Kerberos в AD). Имея билет

службы, клиент может запросить доступ к ней. Элементы ответа TGS_REP представлены ниже:

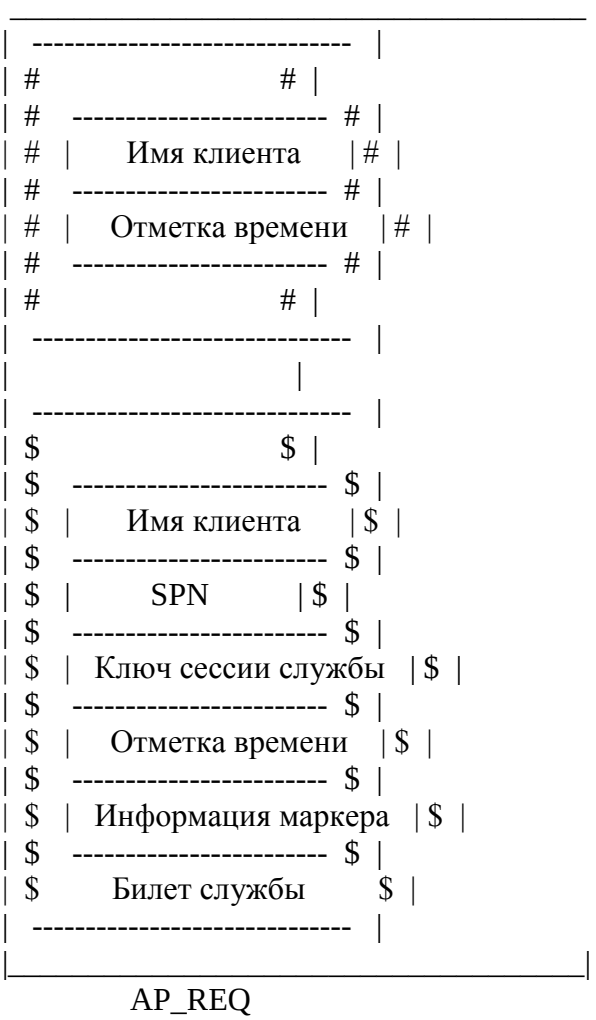


TGS_REP

@Доступ к службам:

- после того, как клиент получает билет службы, приложение, обращающееся к службе, может предъявить этот билет службе и запросить доступ. Механика предъявления билета службы не так стандартизована, как получение билета, из-за различий, свойственных приложениям. Например, в случае со службой HTTP билет службы встраивается в заголовки запроса HTTP.
- клиент отправляет запрос AP_REQ, содержащий в себе билет службы. Служба дешифрует билет службы и получает сеансовый ключ, который можно использовать для дешифрации первого блока данных: полей отметки времени и имени клиента, которые в свою очередь используются для проверки подлинности билета службы. Даже если служба принимает билет службы, на данном этапе клиент просто прошел проверку в службе. Выполнить задачу авторизации службе предстоит на основе информации о клиенте.
- в билет службы также обычно входят данные, известные как сертификат атрибута привилегий Privilege Attribute Certificate (PAC). Это та же информация маркера, которую KDC включает в билет TGT пользователя. Сертификат PAC составлен из такой информации, как идентификатор безопасности (SID) пользователя, сведения о членстве в группе и правах безопасности/привилегиях пользователя. Когда пользователь предъявляет билет TGT в центр KDC, чтобы запросить билет службы, KDC копирует информацию маркера из TGT и вставляет в поле PAC билета службы. Служба использует эту информацию, чтобы подготовить маркер доступа для пользователя и проверить авторизацию пользователя, обычно на основе членства в группе.

- допускается передача дополнительного сообщения Kerberos, известного как AP_REP или Application Reply, после того как пользователь предъявляет билет службы в сообщении AP_REQ. Сообщение Application Reply — необязательное; как правило, приложение не отправляет такое сообщение, если не происходит ошибки. Пример ситуации, когда формируется сообщение AP_REP: клиент запрашивает (в сообщении AP_REQ) у службы подтверждение подлинности для обоюдной проверки подлинности. Элементы запроса AP_REQ представлены ниже:



##
###

85. Алгоритм RSA. Принцип работы, взаимная обратность отображений шифрования и дешифрования, вопросы выбора параметров, приложения, основные виды атак.

RSA - криптографический алгоритм с открытым ключом, который основывается на вычислительной сложности задачи факторизации больших целых чисел. Используется в большом числе криптографических приложений (PGP, SSL/TLS, ...).

В криптографической системе с открытым ключом каждый участник располагает как открытым ключом, так и закрытым ключом. В криптографической системе RSA каждый ключ состоит из пары целых чисел. Каждый участник создаёт свой открытый и закрытый ключ самостоятельно. Закрытый ключ каждый из них держит в секрете, а открытые ключи можно сообщать кому угодно или даже публиковать их. Открытый и закрытый ключи

каждого участника обмена сообщениями в криптосистеме RSA образуют «согласованную пару» в том смысле, что они являются взаимно обратными, то есть:
∀ допустимых пар открытого и закрытого ключей (p, s)
  ∃ соответствующие функции шифрования E(x) и расшифрования D(x) такие, что
    ∀ сообщения m ∈ M, где M — множество допустимых сообщений,
      m = D(E(m)) = E(D(m)).

Алгоритм создания ключей:
1) Выбираются два различных случайных простых числа p и q заданного размера (например, 2048 бит каждое)
2) Вычисляется их произведение N = p * q, которое называется модулем
3) Вычисляется значение функции Эйлера от числа N: φ(N) = (p - 1) * (q - 1)
4) Выбирается целое число e (1 < e < φ(N)), взаимно простое с φ(N). Число e называется открытой экспонентой, и в качестве значения выбирают простые числа, содержащие небольшое количество единичных бит в двоичной записи (прим. 65537), благодаря чему время шифрования с использованием быстрого возведения в степень будет значительно меньше
5) Вычисляется число d, мультипликативно обратное к числу e по модулю φ(N), то есть число, удовлетворяющее сравнению: d * e = 1 (mod φ(N))
6) Пара (e, N) публикуется в качестве открытого ключа RSA
7) Пара (d, N) играет роль закрытого ключа RSA и хранится в секрете

Шифрование:
c = m^e (mod N)
Расшифрование:
m = c^d (mod N)

Система RSA может использоваться не только для шифрования, но и для цифровой подписи. Предположим, что Алисе (стороне A) нужно отправить Бобу (стороне B) сообщение m, подтверждённое электронной цифровой подписью. Тогда для Алисы алгоритм будет следующим:
1) Взять открытый текст m
2) Создать цифровую подпись с помощью своего секретного ключа: s = m^d (mod N)
3) Передать пару {m, s} Бобу

Алгоритм для Боба:
1) Принять пару {m, s}
2) Взять открытый ключ {e, N} у Алисы
3) Вычислить прообраз сообщения из полученной подписи: m' = s^e (mod N)
4) Проверить неизменность сообщения, сравнив m и m'
Цифровая подпись обеспечивает как аутентификацию автора сообщения, так и подтверждение целостности содержимого подписанного сообщения.

О выборе параметров p и q:
- для исключения возможностей применения методов факторизации накладываются следующие ограничения: числа p1 = (p - 1)/2, p2 = (p + 1)/2, q1 = (q - 1)/2, q2 = (q + 1)/2 должны быть простыми, причем p1 - 1 и q1 - 1 не должны разлагаться на произведение маленьких простых чисел.

О выборе параметров e и d:
- при использовании малого значения параметра e искомое сообщение можно найти путем извлечения корня степени e

- при использовании малого значения параметра d искомое сообщение можно будет найти путем перебора малых значений до получения корректного расшифрованного сообщения

Атаки на алгоритм RSA:
https://ru.wikipedia.org/wiki/Криптоанализ_RSA

##
###

86. Методы факторизации натуральных чисел

https://ru.wikipedia.org/wiki/Факторизация_целых_чисел
http://old.kpfu.ru/f9/bibl/Monograph_ishm.pdf

##
###

87. Сравнительная характеристика моделей OSI и TCP/IP.

У моделей OSI и TCP/IP имеется много общих черт. Обе модели основаны на концепции стека независимых протоколов.
Функциональность уровней тоже во многом схожа. Например, в обеих моделях уровни, начиная с транспортного и выше, предоставляют сквозную,
не зависящую от сети транспортную службу для процессов, желающих обмениваться информацией. Эти уровни образуют поставщика транспорта.
Также в каждой модели уровни выше транспортного являются прикладными потребителями транспортных сервисов.

Несмотри на это фундаментальное сходство, у этих моделей имеется и ряд отличий.
Для модели OSI центральными являются три концепции: службы, интерфейсы, протоколы. Вероятно, наибольшим вкладом модели OSI
стало явное разделение этих трёх концепций. Каждый уровень предоставляет некоторые сервисы для расположенного выше уровня.
Сервис определяет, что именно делает уровень, но не то, как он это делает и каким образом объекты, расположенные выше,
получают доступ к данному уровню.
Интерфейс уровня определяет способ доступа к уровню для расположенных выше процессов. Он описывает параметры и ожидаемый результат.
Он также ничего не сообщает о внутреннем устройстве уровня. Наконец, равноранговые протоколы, применяемые в уровне,
являются внутренним делом самого уровня. Для выполнения поставленной ему задачи (то есть предоставления сервиса) он может использовать любые протоколы.
Кроме того, уровень может менять протоколы, не затрагивая работу приложений более высоких уровней.

Эти идеи очень хорошо соответствуют современным идеями объекто-ориентированного программирования.
Уровень может быть представлен в виде объекта, обладающего набором методов (операций), к которым может обращаться внешний процесс.

Семантика этих методов определяет набор служб, предоставляемых объектом. Параметры и результаты методов образуют интерфейс объекта.
Внутреннее устройство объекта можно сравнить с протоколом уровня. За пределами объекта оно никого не интересует и никому не видно.

Изначально в модели TCP/IP не было чёткого разделения между службами, интерфейсом и протоколами,
хотя и производились попытки изменить это, чтобы сделать её более похоженй на модель OSI.
Так, например, единственными настоящими сервисами, предоставляемыми межсетевым уровнем, являются SEND IP PACKET и RECEIVE IP PACKET.

В результате в модели OSI протоколы скрыты лучше, чем в модели TCP/IP, и при изменении технологии они могут быть относительно легко заменены.
Возможность проводить подобные изменения, не затрагивая другие уровни, является одной из главных целей многоуровневых протоколов.

Модель OSI была придумана раньше, чем были реализованы протоколы для неё. Поэтому она, с одной стороны, была универсальной,
способной покрывать все конфигурации сетей и стеки протоколов, используемый в ней.
С другой стороны, она мало сооветствовала практике, поэтому приходилось вводить дополнительные уровни, чтобы "сгладить" существующие "шероховатости".

Модель TCP/IP же была описана уже после того, как был реализован соответствующий ей стек протоколов.
Она прекрасно его описывает, тем не менее, ввиду её специфичности, она не использовалась для описания других стеков протоколов, не основанных на TCP/IP.

При близком рассмотрении данных моделей бросается различие в количестве уровней: в модели OSI 7 уровней, в модели TCP/IP - 4.
В обеих моделях имеются межсетевой, транспортный и прикладной уровни, а остальные уровни различные.

Ещё одно различие между моделями лежит в сфере возможности использования связи на основе соединений и связи без установления соединения.
Модель OSI на сетевом уровне поддерживает оба типа связи, а на транспортном - только связь на основе соединений
(поскольку транспортные службы являются видимыми для пользователя). В модели TCP/IP на сетевом уровне есть только один режим связи (без установления соединения), но на транспортном уровне она поддерживает оба режима, предоставляя пользователям выбор. Этот выбор особенно важен для простых протоколов запрос-ответ.

##
###

88. Протоколы модемной связи.

Скорость

Аналоговые каналы тональной частоты характеризуются тем, что спектр передаваемого по ним сигнала ограничен диапазоном от 300 Гц до 3400 Гц.
Именно это ограничение спектра и является основной преградой в использовании телефонных каналов для высокоскоростной передачи цифровой информации.

Электрический сигнал, распространяющийся по каналу, характеризуется тремя параметрами - амплитудой, частотой и фазой.
Именно изменение одного из этих параметров, или даже совместно некоторой их совокупности в зависимости от значений информационных бит и составляет физическую сущность процесса модуляции.
Каждому информационному элементу соответствует фиксированный отрезок времени, на котором электрический сигнал имеет определенные значения своих параметров, характеризующих значение этого информационного элемента. Этот отрезок времени называют бодовым интервалом. Если кодируемый элемент соответствует одному биту информации,
который может принимать значение 0 или 1, то на бодовом интервале параметры сигнала соответственно могут принимать одну из двух предопределенных совокупностей значений амплитуды, частоты и фазы.
В этом случае модуляционная скорость (еще ее называют линейной или бодовой) равна информационной, т.е. 1 бод = 1 бит/с.
Но кодируемый элемент может соответствовать не одному, а, например, двум битам информации. В этом случае информационная скорость будет вдвое превосходить бодовую,
а параметры сигнала на бодовом интервале могут принимать одну из четырех совокупностей значений, соответствующих 00, 01, 10 или 11.

В общем случае, если на бодовом интервале кодируется n бит, то информационная скорость будет превосходить бодовую в n раз.
Но количество возможных состояний сигнала в трехмерном (в общем случае) пространстве - амплитуда, частота, фаза - будет равно 2^n.
Это значит, что демодулятор модема, получив на бодовом интервале некий сигнал, должен будет сравнить его с 2^n эталонными сигналами и безошибочно выбрать
один из них для декодирования искомых n бит. Таким образом, с увеличением емкости кодирования и ростом информационной скорости относительно бодовой, расстояние в сигнальном пространстве
между двумя соседними точками сокращается в степенной прогрессии. А это, в свою очередь, накладывает все более жесткие требования к "чистоте" канала передачи.
Теоретически возможная скорость в реальном канале определяется известной формулой Шеннона:

$$V = F * \log(1 + S/N),$$

где F - ширина полосы пропускания канала, S/N - отношение сигнал/шум.

Второй сомножитель и определяет возможности канала с точки зрения его зашумленности по достоверной передаче сигнала, кодирующего не один бит информации в бодовом интервале.
Так, например, если отношение сигнал/шум соответствует 20 dB, т.е. мощность сигнала, доходящего до удаленного модема, в 100 раз превосходит мощность шума,
и используется полная полоса канала тональной частоты (3100 Гц), максимальная граница по Шеннону равна 20640 бит/с.

Модуляция

В модемной связи используются следующие виды модуляции:
* частотная,
* фазоразностная,
* многопозиционная амплитудно-фазовая модуляция.

При частотной модуляции (FSK, Frequency Shift Keying) значениям 0 и 1 информационного бита соответствуют свои частоты физического сигнала при неизменной его амплитуде.
Частотная модуляция весьма помехоустойчива, поскольку искажению при помехах подвергается в основном амплитуда сигнала, а не частота.
При этом достоверность демодуляции, а значит и помехоустойчивость тем выше, чем больше периодов сигнала попадает в бодовый интервал.
Но увеличение бодового интервала по понятным причинам снижает скорость передачи информации.
С другой стороны, необходимая для этого вида модуляции ширина спектра сигнала может быть значительно уже всей полосы канала.
Отсюда вытекает область применения FSK - низкоскоростные, но высоконадежные стандарты, позволяющие осуществлять связь на каналах с большими искажениями амплитудно-частотной характеристики,
или даже с усеченной полосой пропускания.

При фазоразностной модуляции (DPSK, Differential Phase Shift Keying) изменяемым в зависимости от значения информационного элемента параметром является фаза сигнала при неизменных амплитуде и частоте. При этом каждому информационному элементу ставится в соответствие не абсолютное значение фазы, а ее изменение относительно предыдущего значения.
Если информационный элемент есть дибит, то в зависимости от его значения (00, 01, 10 или 11) фаза сигнала может измениться на 90, 180, 270 градусов или не измениться вовсе.
Из теории информации известно, что фазовая модуляция наиболее информативна, однако увеличение числа кодируемых бит выше трех (8 позиций поворота фазы)
приводит к резкому снижению помехоустойчивости. Поэтому на высоких скоростях применяются комбинированные амплитудно-фазовые методы модуляции.

Многопозиционную амплитудно-фазовую модуляцию называют еще квадратурной амплитудной модуляцией (QAM, Quadrature Amplitude Modulation).
Здесь помимо изменения фазы сигнала используется манипуляция его амплитудой, что позволяет увеличивать число кодируемых бит. В настоящее время используются модуляции,
в которых количество кодируемых на одном бодовом интервале информационных бит может доходить до 8, а, соответственно, число позиций сигнала в сигнальном пространстве - до 256.
Однако, применение многоточечной QAM в чистом виде сталкивается с серьезными проблемами, связанными с недостаточной помехоустойчивостью кодирования.
Поэтому во всех современных высокоскоростных протоколах используется разновидность этого вида модуляции, т.н. модуляция с решетчатым кодированием или треллис-кодированием
(TCM, Trellis Coded Modulation), которая позволяет повысить помехозащищенность передачи информации - снизить требования к отношению сигнал/шум в канале на величину от 3 до 6 дБ.

Суть этого кодирования заключается в введении избыточности. Пространство сигналов расширяется вдвое путем добавления к информационным битам еще одного, который образуется посредством сверточного кодирования над частью информационных бит и введения элементов запаздывания.
Расширенная таким образом группа подвергается все той же многопозиционной амплитудно-фазовой модуляции.
В процессе демодуляции принятого сигнала производится его декодирование по весьма изощренному алгоритму Виттерби, позволяющему за счет введенной избыточности и знания предистории
выбрать по критерию максимального правдоподобия из сигнального пространства наиболее достоверную точку и, тем самым, определить значения информационных бит.

Дуплекс

Под дуплексным режимом работы понимается возможность передавать информацию в обе стороны одновременно. Обычный телефонный канал - типичный пример дуплексного канала.
В модемной связи поддержка дуплексного режима работы определяется возможностями протокола физического уровня.
Проблема для модема заключается не в способности канала передавать дуплексную информацию, а в возможности демодулятора модема распознать входной сигнал
на фоне отраженного от аппаратуры АТС собственного выходного сигнала, который фактически становится для модема шумом. При этом его мощность может быть не только сравнима,
но в большинстве случаев значительно превосходить мощность принимаемого полезного сигнала.

При необходимости обеспечивать дуплекс при работе по двухпроводной линии, используется частотное разделение каналов.
Вся полоса пропускания канала разделяется на два частотных подканала, по каждому из которых производится передача в одном направлении.
Выбор подканала передачи осуществляется на этапе установки соединения и, как правило, однозначно связан с ролью модема в сеансе связи: вызывающий или отвечающий.
Очевидно, что этот метод не позволяет использовать возможности канала в полном объеме ввиду значительного сужения полосы пропускания.
Тем более, что для исключения проникновения боковых гармоник в соседний подканал, разносить их приходится со значительным "зазором",
в результате чего частотные подканалы занимают отнюдь не половину полного спектра.
Соответственно (см. формулу Шеннона),
данный метод обеспечения дуплексной связи ограничивает скорость передачи информации. Существующие протоколы физического уровня, использующие частотное разделение каналов,
обеспечивают симметричную дуплексную связь со скоростями, не превышающими 2400 бит/с.

Ряд протоколов обеспечивают более скоростную связь, но в одном направлении, в то время как обратное направление - значительно медленнее.
Разделение частот в этом случае осуществляется на неравные по ширине полосы пропускания подканалы. Эта разновидность дуплексной связи называется асимметричной.

Другим методом обеспечения симметричного дуплекса, который используется во всех высокоскоростных протоколах, является технология эхо-подавления (эхо-компенсации).

Суть ее заключается в том, что модемы, обладая информацией о собственном выходном сигнале, могут использовать это знание для фильтрации собственного "рукотворного" шума из принимаемого сигнала.
На этапе вхождения в связь каждый модем, посылая некий зондирующий сигнал, определяет параметры эхо-отражения: время запаздывания и мощность отраженного сигнала.
А в процессе сеанса связи эхо-компенсатор модема "вычитает" из принимаемого входного сигнала свой собственный выходной сигнал, скорректированный в соответствии с полученными параметрами эхо-отражения.
Эта технология позволяет использовать для дуплексной передачи информации всю ширину полосы пропускания канала, однако требует при реализации весьма серьезных вычислительных ресурсов на сигнальную обработку.

Многие протоколы не обеспечивают дуплексную связь. Это так называемые полудуплексные протоколы. В частности, все протоколы, предназначенные для факсимильной связи - полудуплексные.
В этом случае в каждый момент времени информация передается только в одну сторону. По окончании приема/передачи некоторой порции информации оба модема (факса) синхронно переключают направление передачи данных (ping-pong). Ввиду отсутствия проблем с взаимным проникновением подканалов передачи,
а также с эхо-отражением, полудуплексные протоколы в общем случае характеризуются большей помехоустойчивостью и возможностью использования всей ширины полосы пропускания канала.
Однако эффективность использования канала для передачи данных по сравнению с дуплексными протоколами ниже. Связано это прежде всего с тем, что практически все протоколы передачи данных,
как канального уровня (MNP, V.42), так и уровня передачи файлов (X, Y, Zmodem, не говоря уже о протоколах типа BiDirectional), требуют двустороннего обмена,
по крайней мере для подтверждения принятой информации. А любое переключение направления передачи, помимо невозможности в данный момент передавать очередную порцию пользовательской информации,
требует дополнительных накладных расходов по времени на взаимную пересинхронизацию приемной и передающей сторон.

Общеупотребительные модемные протоколы ITU-T

1) V.21

Это дуплексный протокол с частотным разделением каналов и частотной же модуляцией FSK. На нижнем канале (его обычно использует для передачи вызывающий модем) "1" передается частотой 980 Гц,
а "0" - 1180 Гц. На верхнем канале (передает отвечающий) "1" передается частотой 1650 Гц, а "0" - 1850 Гц. Модуляционная и информационная скорости равны - 300 бод, 300 бит/с.
Несмотря на невысокую скорость, данный протокол находит применение прежде всего в качестве "аварийного",
при невозможности вследствие высокого уровня помех использовать другие протоколы физического уровня. Кроме того, ввиду своей неприхотливости и помехоустойчивости, он используется в специальных высокоуровневых приложениях, требующих высокой надежности передачи.

2) V.22

Это дуплексный протокол с частотным разделением каналов и модуляцией DPSK.
Несущая частота нижнего канала (передает вызывающий) - 1200 Гц, верхнего (передает отвечающий) - 2400 Гц.
Модуляционная скорость - 600 бод. Имеет режимы двухпозиционной (кодируется бит) и четырехпозиционной (дибит) фазоразностной модуляции с фазовым расстоянием между точками,
соответственно, в 180 и 90 град. Соответственно, информационная скорость может быть 600 или 1200 бит/с. Этот протокол фактически поглощен протоколом V.22bis.

3) V.22bis

Это дуплексный протокол с частотным разделением каналов и модуляцией QAM.
Несущая частота нижнего канала (передает вызывающий) - 1200 Гц, верхнего - 2400 Гц.
Модуляционная скорость - 600 бод. Имеет режимы четырехпозиционной (кодируется дибит) и шестнадцатипозиционной (кодируется квадробит) квадратурной амплитудной модуляции.
Соответственно, информационная скорость может быть 1200 или 2400 бит/с. Режим 1200 бит/с полностью совместим с V.22, несмотря на другой тип модуляции.
Дело в том, что первые два бита в режиме 16-QAM (квадробит) определяют изменение фазового квадранта относительно предыдущего сигнального элемента и потому за амплитуду не отвечают,
а последние два бита определяют положение сигнального элемента внутри квадранта с вариацией амплитуды.
Таким образом, DPSK можно рассматривать как частный случай QAM, где два последних бита не меняют своих значений.
В результате из шестнадцати позиций выбираются четыре в разных квадрантах, но с одинаковым положением внутри квадранта, в том числе и с одинаковой амплитудой.
Протокол V.22bis является стандартом де-факто для всех среднескоростных модемов.

4) V.32

Это дуплексный протокол с эхо-подавлением и квадратурной амплитудной модуляцией или модуляцией с решетчатым кодированием.
Частота несущего сигнала - 1800 Гц, модуляционная скорость - 2400 бод. Таким образом, используется спектр шириной от 600 до 3000 Гц.
Имеет режимы двухпозиционной (бит), четырехпозиционной (дибит) и шестнадцатипозиционной (квадробит) QAM.
Соответственно, информационная скорость может быть 2400, 4800 и 9600 бит/с. Кроме того, для скорости 9600 бит/с имеет место альтернативная модуляция - 32-позиционная TCM.

5) V.32bis

Это дуплексный протокол с эхо-подавлением и модуляцией TCM. Используются те же, что в V.32, частота несущего сигнала - 1800 Гц, и модуляционная скорость - 2400 бод.
Имеет режимы 16-TCM, 32-TCM, 64-TCM и 128-TCM. Соответственно, информационная скорость может быть 7200, 9600, 12000 и 14400 бит/с.
Режим 32-TCM полностью совместим с соответствующим режимом V.32. Протокол V.32bis является стандартом де-факто для всех скоростных модемов.

Нестандартные модемные протоколы

1) V.32terbo

Этот протокол, разработанный фирмой AT&T, является открытым для реализации разработчиками модемов. В частности, помимо БИС фирмы AT&T,
данный протокол реализован в некоторых модемах фирмы U.S.Robotics. Протокол фактически является механическим развитием технологии V.32bis: дуплекс с эхо-подавлением,
модуляция с решетчатым кодированием, модуляционная скорость - 2400 бод, несущая - 1800 Гц, расширение информационных скоростей значениями 16800 и 19200 бит/с за счет 256-TCM и 512-TCM.
Следствием такого подхода является весьма жесткие требования, предъявляемые данным протоколом к линии.
Так, например, для устойчивой работы на скорости 19200 бит/с отношение сигнал/шум должно быть не менее 30 dB.

2) ZyX

Протокол разработан фирмой ZyXEL Communications Corporation и реализован в собственных модемах.
Этот протокол также, как и V.32terbo, расширяет V.32bis значениями информационных скоростей 16800 и 19200 бит/с с сохранением технологии эхо-подавления,
модуляции с треллис-кодированием и несущей 1800 Гц. Модуляционная же скорость 2400 бод сохраняется лишь для 16800 бит/с.
Скорость 19200 бит/с обеспечивается повышением модуляционной скорости до 2743 бод при сохранении режима модуляции 256-TCM для обоих скоростей.
Такое решение позволяет снизить требование к отношению сигнал/шум на линии на 2.4 dB,
однако расширение полосы пропускания может негативно сказываться при больших искажениях амплитудно-частотной характеристики канала.

3) HST

Протокол HST (High Speed Technology) разработан фирмой U.S.Robotics и реализован в модемах фирмы серии Courier.
Это асимметричный дуплексный протокол с частотным разделением каналов. Обратный канал имеет режимы 300 и 450 бит/с. Основной канал - 4800, 7200, 9600, 12000, 14400 и 16800 бит/с.
Применяется модуляция с решетчатым кодированием и модуляционной скоростью 2400 бод.
Характеризуется сравнительной простотой и высокой помехоустойчивостью вследствие отсутствия необходимости в эхо-компенсации и отсутствия же взаимовлияния каналов.

От Яны:

Модем (аббревиатура из слов модулятор-демодулятор) – это устройство, способное передавать цифровые данные через аналоговые каналы. Модемы широко применяются для связи компьютеров через телефонную сеть (телефонный модем). Для того чтобы модемы различных производителей могли работать друг с другом, существуют протоколы модемной связи. Эти протоколы однозначно определяют метод преобразования цифрового сигнала в аналоговый (способ модуляции) (аналоговый сигнал

распространяется непрерывно, а цифровой сигнал – дискретно, прерывисто), скорость передачи данных, последовательность передачи служебных данных, коррекцию ошибок. Протоколы модемной связи делятся на протоколы физического уровня и канального уровня.

Протоколы физического уровня можно разделить на 2 группы:
1) международные
Протоколы международного уровня разрабатываются Международным союзом электросвязи.
Существует серия протоколов V.21-V.92.
– увеличение скорости передачи данных началось с 300 бит/с
– менялся режим работы: от симплексного(передача только в одну сторону) к полудуплексному(передача в две стороны, но в конкретный момент времени передача только в одну сторону) и к дуплексному(одновременная передача в обе стороны)
– появился протокол сжатия данных
– появился протокол видеосвязи, обеспечивает скорость передачи видео до 10—15 кадров в сек
Самый современный протокол V.92 – Скорость 56000 бит/с
2) фирменные
Они были разработаны производителями оборудования. Например, свои протоколы были у компании Bell, но ими перестали пользоваться, когда появились международные протоколы.

Протоколы канального уровня по-другому называют протоколы передачи файлов. Виды протоколов:
1) Старт-стоп протокол – характеризуется тем, что, прежде чем посылать новый кадр (блок данных определенной длины) информации, передатчик ждет подтверждения о правильном получении приемником предыдущего кадра
2) Конвейерный протокол – такое подтверждение может быть получено после передачи нескольких кадров.
В последнем случае меньше задержки на ожидание подтверждений , но больше затраты на повторную пересылку в случае ошибок.

Протоколы:
1) XModem – используется старт-стопное управление, размер одного блока (пакета) равен 128 байт и 1 байт отводится под контрольную сумму. Недостаток: большая вероятность не обнаружить ошибку.
2) XModem-CRC – длина проверочной последовательности составляет 16 бит.
Недостатки этих протоколов: передает только 1 пакет за раз
3) YModem – реализована групповая передача кадров
4) ZModem – используется конвейерное управление, длина пакета от 64 до 1024 байт.
Возможно восстановление связи при обрыве: прерванная передача продолжается с места прерывания.

##
###

89. Протоколы маршрутизации.

Классификация протоколов маршрутизации:

* Внутридоменная маршрутизация - протокол маршрутизации, применяемый внутри автономной системе.
Пример: OSPF (Open Shortest Path First).
* Междоменная маршрутизация - протокол маршрутизации, применяемый на границе независимых сетей.
Пример: BGP (Border Gateway Protocol).

OSPF.

Протокол OSPF разработан как протокол маршрутизации, используемый внутри автономной системы,
такой как локальная вычислительная сеть (ЛВС). Протокол реализует алгоритм Дейкстры для вычисления кратчайшего маршрута.
Как протокол, учитывающий состояние канала (link-state), OSPF поддерживает базу данных,
описывающую состояния каналов (сред передачи данных) в сети.
Каждый роутер, на котором работает OSPF, передаёт через каждый свой интерфейс, в широковещательном пакете,
информацию о стоимости канала (link cost), к которому подключён данный интерфейс.
Стоимость канала обычно обратно пропорциональна битрейту (скорости),
на котором данный канал способен передавать данные. Данная процедура называется стадией приветствия (hello procedure).
Все роутеры, реализующие протокол OSPF, периодически отправляют данные пакеты, называемые "пакетами-приветствиями" (hello-packets).
Таким образом, изменения стоимости каналов, к которым подключён роутер, становятся известны его соседям.
Информация о стоимости канала, связанная со скоростью двухточёчного соединения между двумя роутерами, затем распространяется по сети,
поскольку роутеры, реализующие OSPF, передают информацию, полученную от их соседей, другим соседям.
Данный процесс распространения информации о состоянии каналов по сети называется синхронизацией.
Основываясь на данной информации, все роутеры, реализующие OSPF, непрерывно обновляют свои
базы данных состояний каналов, а также правят таблицы маршрутизации.

Сеть OSPF может быть цельной или же разделённой на отдельные области маршрутизации, для облегчения
администрирования и управления трафиком.
Области идентифицируются при помощи 32-х битных беззнаковых целых величин, обозначаемых либо в
десятичной форме, либо в форме IPv4 адреса. По соглашению, нулевая область (0.0.0.0) является
основной (core/backbone area) областью OSPF-сети. Хотя идентификаторы других областей в сети могут быть выбраны произвольно,
администраторы обычно выбирают IP-адрес главного роутера данной области в качестве идентификатора.
Каждая дополнительная облсть должна быть соединена с основной областью OSPF-сети.
Такие соединения поддерживаются межобластными роутерами, называемыми также пограничными роутерами области (area border router, ABR).

Пограничный роутер области поддерживает несколько отдельных баз данных состояний каналов,
по одной для каждой области, к которой он подключён.
Кроме этого поддерживаются суммарные маршруты для каждой из областей в OSPF-сети.

OSPF обнаруживает изменения в топологии, такие как обрывы связи (link failures), и сходится к новой маршрутной структуре, свободной от петель маршрутизации, в течение нескольких секунд.

BGP.

Соседние роутеры, называемые, в терминологии BGP, пирами, назначаются из числа роутеров вручную, для создания TCP-соединения на порту 179.
BGP-пир каждые 60 секунд отправляет 19-байтное сообщения "я живой" (keep-alive), чтобы поддерживать соединение.
Среди протоколов маршрутизации BGP выделяется тем, что использует TCP в качестве протокола транспортного уровня.

Роутеры, расположенные в различных автономных системах, обменивающиеся друг с другом данными по протоколу BGP, называются пограничными роутерами.
Пограничные роутеры, как правило, непосредственно соединены друг с другом, в пределах одного ethernet-сегмента.
Новые маршруты, полученные от погранченого роутера, распространяются среди всех роутеров, образующих автономную систему.

Способ распространения маршрутов управляется через механизм, называемый "маршрутными соответствиями" (route-maps).
Данный механизм состоит из набора правил. Каждое правило описывает, для маршрутов, удовлетворяющих некоторому критерию,
какое именно действие должно быть предпринято для данного маршрута. В число действий входит отбрасывание маршрута или модификация некоторых его параметров, перед добавлением его в таблицу маршрутизации.

##
###

90. Протоколы IPX/SPX, Netbios.

IPX Headers & Operation --> https://ru.bmstu.wiki/IPX_(Internetwork_Packet_Exchange)

SPX Headers & Operation --> https://ru.bmstu.wiki/SPX_(Sequenced_Packet_Exchange)

Netbios Headers & Operation --> https://ru.bmstu.wiki/NetBIOS

##
###

91. Методы обеспечения безопасности и распределения доступа в UNIX-подобных ОС.

:--:

##
###

92. Журналируемые файловые системы (на примере ОС семейства UNIX/Linux).

---> https://www.ibm.com/developerworks/ru/library/l-anatomy-ext4/index.html
---> https://www.ibm.com/developerworks/ru/library/l-journaling-filesystems/index.html

В режиме обратной записи журналированию подвергаются только метаданные, а блоки с данными записываются непосредственно на диск. Это способствует нерушимости структуры файловой системы и защищает от повреждений, однако повреждение самих данных все же возможно (например, если крах системы наступает после записи метаданных в журнал, но до записи блока с данными). Решить указанную проблему позволяет режим упорядочивания. В этом режиме в журнал заносятся также только метаданные, но сами данные записываются до журналирования метаданных. Этим гарантируется согласованность данных файловой системы после восстановления. И наконец, возможно журналирование в режиме данных, при котором в журнал заносятся как метаданные, так и сами данные. Этот режим обладает наивысшим уровнем устойчивости к повреждению и потере данных, но имеет недостаток в виде низкой производительности, поскольку все данные записываются дважды (сначала в журнал, потом на диск).

##
###

93. Командные оболочки ОС семейства UNIX/Linux.

:--:

##
###

94. Реализация системы защиты операционных систем Microsoft Windows.

---> https://www.anti-malware.ru/analytics/Technology_Analysis/Built_in_protection_system_Windows_8

##
###

95. Реализация системы защиты UNIX-подобных операционных систем

:--:

##
###

96. Вредоносные программы: классификация, основные характеристики, современные тенденции в развитии вредоносных программ

Вредоносная программа – программа, используемая для осуществления несанкционированного доступа к информации и (или) воздействия на информацию или ресурсы автоматизированной информационной системы (из ГОСТ 51275-2006 Защита информации. Объект информатизации…).
Классификация вредоносных программ:
1) Вирус – самовоспроизводящийся программный код, который внедряется в установленные программы без согласия пользователя. В дополнение к этому вирус может быть запрограммирован на выполнение вредоносных действий (например, удаление или порча файлов) и самомодификацию. Примеры: Virus 1, 2, 3, Elk Cloner, «Чернобыль».
2) Червь – саморазмножающаяся программа, которая поселяется на компьютер жертвы, а затем ищет уязвимости в Сети или системе для дальнейшего распространения себя. Некоторые черви существуют в виде сохраненных на жестком диске файлов, а некоторые поселяются в оперативной памяти компьютера. Примеры: червь Морриса, Stuxnet, Wanna Cry.
3) Троян – вредоносная программа, которая отличается от самопроизвольно распространяющихся вирусов и червей тем, что она распространяется злоумышленниками. Большинство троянских программ маскируется под безвредные или полезные программы, чтобы пользователь загрузил/установил их на свой компьютер. Злоумышленники помещают троянские программы на открытые и индексируемые ресурсы, носители информации, присылают их предполагаемым жертвам по электронной почте, также трояны устанавливаются на компьютер через бреши безопасности. Примеры: Trojan.Spy, Trojan.Downloader, Trojan.SMS.
4) Руткит – вредоносная программа, специально разработанная для сокрытия присутствия вредоносного кода и его действий от пользователя и установленного защитного программного обеспечения. Некоторые руткиты могут начинать свою работу прежде, чем загрузится операционная система (буткит). Примеры: Blue Pill, Haxdoor, Mebroot.
5) Бэкдор (средство удаленного администрирования) – приложение, которое позволяет злоумышленнику управлять компьютером на расстоянии. В зависимости от функциональных особенностей конкретного бэкдора, злоумышленник может установить и запустить на компьютере любое программное обеспечение, сохранять все нажатия клавиш, загружать и сохранять файлы, делать снимки с веб-камеры и т.п. Примеры: Linux.Backdoor.*, Python.Backdoor.*, Backdoor:MSIL/Sorcas.A.
6) Загрузчик – небольшая программа, которая используется лишь для дальнейшей загрузки и установки полной версии вредоносной программы. После того, как загрузчик попадает на компьютер жертвы (например, после открытия вложения из полученного письма), он соединяется с удаленным сервером и загружает всю вредоносную программу. Пример: Nemucode.
7) Вредоносные утилиты – программы, разработанные для автоматизации создания других вредоносных программ, организации DoS-атак на удаленные сервера, взлома компьютеров и т.д. В отличие от предыдущих категорий, такие программы не

представляют угрозы компьютеру, на котором исполняются. Примеры: Email-Flooder, DoS, Spoofer.
8) Нежелательное ПО – программы, которые по своей сути не являются вредоносными, но в большинстве случаев могут надоедать пользователю. Примеры: AdWare, SpyWare, zip-bomb.

Основные характеристики вредоносных программ:
− целевая среда. Устройства, операционные системы, приложения и т.п.;
− механизм передачи. Съемные носители, общие сетевые диски, сеть, электронная почта и т.п.;
− вредоносные действия. Порча, уничтожение, хищение информации, отказ в обслуживании и т.п.;
− механизмы активации. Ручной (социальная инженерия), полуавтоматический, автоматический (в том числе по событию);
− механизмы защиты. Обфускация, упаковка, генерация мусора, олигоморфизм, полиморфизм, метаморфизм, обратные атаки на антивирусное программное обеспечение.

Современные тенденции в развитии вредоносных программ:
− использование техники «living off the land». Злоумышленники, чтобы избежать обнаружения, всё чаще используют программы, которые уже установлены у жертв. Так, например, NotPetya распространялся по сети благодаря утилите PSExec и инструментария управления Windows (WMI). Эти инструменты не классифицируются как угрозы, потому что являются легитимным программным обеспечением, и поэтому не детектируются сканерами;
− использование «plug-and-play» червей. Злоумышленники после инцидента с WannaCry стали намного чаще полагаться на возможности распространения червей по сети и их последующее закрепление в системе путем планирования задач и формирования бэкдоров;
− использование аппаратных уязвимостей для проведения атак. После обнаружения уязвимостей Meltdown и Spectre, позволяющих вредоносным приложениям получать доступ к конфиденциальным данным из памяти, злоумышленники выделили их в качестве приоритетных. По материалам ЛК, отдельные APT группировки уже тестируют образцы вредоносных программ, использующих эти уязвимости.
Дополнительно можно рассказать про внимание к IoT устройствам и увеличению количества вредоносных программ для мобильных устройств.

##
###

97. Компьютерные вирусы: классификация, основные характеристики, способы внедрения в программный код, способы сокрытия факта заражения и основные демаскирующие признаки

Компьютерный вирус – самовоспроизводящийся программный код, который внедряется в установленные программы без согласия пользователя. В дополнение к этому вирус может быть запрограммирован на выполнение вредоносных действий (например, удаление или порча файлов) и самомодификацию.

Классификация вирусов:
1) По целевой среде

- компьютерные вирусы для различных аппаратных платформ (вирусы для определенной аппаратной платформы, межплатформные вирусы, платформно-независимые вирусы (вирусы виртуальных машин – Java-вирусы, .NET-вирусы));
- компьютерные вирусы для различных операционных систем (вирусы для определенных ОС, переносимые вирусы (за счет бинарной совместимости ОС, за счет переносимости исходного кода, например, скрипт-вирусы)).

2) По объекту-носителю
- вирусы для исполняемых файлов (COM-вирусы, EXE-вирусы (различают MZ-, PE-вирусы), COFF/ELF-вирусы);
- вирусы для исполняемых объектов (COM/ActiveX-вирусы, Java-вирусы, .NET-вирусы);
- загрузочные вирусы;
- скрипт-вирусы;
- макровирусы;
- комбинированные.

Выбор объекта-носителя может происходить следующим образом:
- произвольная жертва (заражется всё подряд);
- по определенным критериям (заражение происходит, например, только определенных ОС);
- только конкретные объекты (например, только PE-файлы).

3) По способу заражения
- классические вирусы (внедряются в объект-носитель, стараясь максимально скрыть своё присутствие);
- "вандалы" (внедряются в объект-носитель, не стараясь скрыть своё присутствие, могут повредить объект-носитель);
- "спутники" (существуют в виде отдельного объекта "рядом" с носителем, заражения как такового не происходит).

4) По принципу выбора жертвы
- вирусы-сканеры (определяют жертву в момент своей активации);
- вирусы-мониторы (отслеживают активность потенциальных объектов-носителей с целью определения возможности заражения).

5) По размещению в системе
- резидентный вирус при заражении компьютера оставляет в оперативной памяти свою резидентную часть, которая потом перехватывает обращение операционной системы к объектам заражения (файлам, загрузочным секторам дисков и т.п.) и внедряется в них. Резидентные вирусы находятся в памяти и являются активными вплоть до выключения или перезагрузки компьютера;
- нерезидентный вирус не заражает память компьютера и является активным ограниченное время. Активизируется в определенные моменты, например, при обработке документов текстовым редактором.

6) По способу активации
- ручная (социальная инженерия);
- автоматическая (характерная для загрузочных вирусов, а также для файловых вирусов с возможностью автозапуска программы);
- полуавтоматическая (характерная для "спутников");
- логические бомбы (по наступлению определенного события);
- временные бомбы (по наступлению определенного момента времени).

7) По способу защиты от удаления
- незащищенные вирусы;
- зашифрованные вирусы (в том числе упаковка);

- размазывание вируса (записывается в свободные участки объекта-носителя с использованием переходов);
- обфускация;
- олигоморфизм (низкий уровень защиты от сигнатурного поиска);
- полиморфизм (средний уровень защиты от сигнатурного поиска);
- метаморфизм (высокий уровень защиты от сигнатурного поиска);
- активная защита (нападение на антивирусные средства, дополнительные средства от анализа).

Способы внедрения в программный код:
- размещение X-кода поверх оригинальной программы (затирание);
- размещение X-кода в свободном месте программы (интеграция);
- дописывание X-кода в начало, середину или конец файла с сохранением оригинального содержимого;
- размещение X-кода вне основного тела файла-носителя (например, в динамической библиотеке или NTFS-потоке), загружаемого "головой" X-кода, внедренной в файл способами 1-3.

Способы сокрытия факта заражения:
- стелс-вирусы. При попытке чтения зараженного сектора диска эти вирусы "подставляют" вместо себя незараженный оригинал;
- обфускация. Код вируса обрабатывается таким образом, что его становится сложно анализировать исследователю, но функциональность при этом сохраняется;
- упаковка. Тело вируса сжимается и исполняется только после распаковки декриптором, который прикрепляется к упакованному телу вируса (примеры: UPX, ASpack);
- генерация мусора. Код вируса "разбавляется" бесполезными инструкциями, которые затрудняют его анализ и мешают сигнатурному анализу;
- пермутация. Перестановка логических блоков в теле вируса;
- полиморфизм. Для упаковки вируса каждый раз используется новый ключ, зависимый, например, от объема файла, который он инфицирует;
- метаморфизм. То же, что и полиморфизм, но в каждом поколении вирусов генерируется новый код декриптора, что мешает сигнатурному анализу;
- "антипесочница". Вирус не проявляет (либо проявляет позднее) свои деструктивные свойства, если определяет окружающую среду как виртуальную.

Основные демаскирующие признаки:
- наличие нетипичного стартового поведения в момент загрузки программы;
- наличие известных сигнатур (в том числе строковые, который явно выдают присутствие вируса);
- нетипичный набор секций, измененные имена секций;
- нетипичные таблицы импорта.

##
###

98. Антивирусные программы: классификация антивирусных программ, способы обнаружения и уничтожения вредоносного кода, характеристика современных антивирусных программ

Антивирусная программа - это программа, предназначенная для противодействия ВПО.

Классификация антивирусных программ:

1) По средствам блокирования

- программные;
- программно-аппаратные.

2) По размещению в оперативной памяти

- резидентные (находятся в памяти компьютера и осуществляют автоматическую проверку файлов и происходящих событий);
- нерезидентные (запускаются по требованию пользователя или по определенному расписанию).

3) По способу защиты от ВПО

- программы-детекторы (сканеры). Находят ВПО в оперативной памяти, на внутренних и(или) внешних носителях, выводя сообщение при обнаружении вируса;
- программы-доктора (фаги). Находят зараженные файлы и "лечат" их, удаляя тело вируса из файла. Среди этого вида программ существуют полифаги, предназначенные для поиска и удаления разнообразных видов ВПО;
- программы-вакцины (иммунизаторы). Выполняют "иммунизацию" системы (файлов, каталогов), блокируя возможное действие ВПО;
- программы-ревизоры. Запоминают
- программы-мониторы. Начинают свою работу при запуске операционной системы, постоянно находятся в памяти компьютера и осуществляют автоматическую проверку файлов;
- программы-фильтры. Резидентные программы, которые оповещают пользователя обо всех подозрительных действиях.

В соответствии с "Требованиями к средствам антивирусной защиты" ФСТЭК:

- тип «А» – средства антивирусной защиты (компоненты средств антивирусной защиты), предназначенные для централизованного администрирования средствами антивирусной защиты, установленными на компонентах информационных систем (серверах, автоматизированных рабочих местах);
- тип «Б» – средства антивирусной защиты (компоненты средств антивирусной защиты), предназначенные для применения на серверах информационных систем;
- тип «В» – средства антивирусной защиты (компоненты средств антивирусной защиты), предназначенные для применения на автоматизированных рабочих местах информационных систем;
- тип «Г» – средства антивирусной защиты (компоненты средств антивирусной защиты), предназначенные для применения на автономных автоматизированных рабочих местах.

Средства антивирусной защиты типа «А» не применяются в информационных системах самостоятельно и предназначены для использования только совместно со средствами антивирусной защиты типов «Б» и (или) «В».

Способы обнаружения и уничтожения вредоносного кода:

На территории Российской Федерации деятельность, связанная с обнаружением вредоносных программ и последующим их устранением определяется стандартом ГОСТ Р 51188-98. Согласно этому стандарту, при испытаниях программных средств на наличие вредоносного кода используются две основные группы методов обнаружения: программные и аппаратно-программные. К программным методам относятся:

1) Сигнатурное сканирование – это один из самых простых методов обнаружения вредоносных программ, которой обычно применяется в первую очередь. Принцип его работы заключается в проверке содержимого анализируемого объекта на предмет наличия в нём сигнатур уже известных угроз. Сигнатурой в данном случае называется некоторая последовательность байт, необходимая и достаточная для однозначной идентификации

угрозы. При этом сравнение содержимого исследуемого объекта с сигнатурами может производиться не напрямую, а по их контрольным суммам, что позволяет значительно снизить размер записей в вирусных базах, сохранив при этом однозначность соответствия и, следовательно, корректность обнаружения угроз и лечения инфицированных объектов. Следует отметить, что сигнатурное сканирование может не фиксировать наличие полиморфных вирусов, то есть вредоносных программ, которые способны формировать свой программный код «на лету», уже во время исполнения. Для обнаружения таких угроз существуют другие методы, например, эвристический, который будет рассмотрен далее;

2) Эвристическое сканирование – метод обнаружения вирусов, нацеленный на обнаружение ранее неизвестных вирусным базам вредоносных программ. Этот метод сканирования не обеспечивает какой-либо гарантированной защиты от новых, отсутствующих в сигнатурном наборе компьютерных вирусов, что обусловлено использованием в качестве объекта анализа сигнатур ранее известных вирусов, а в качестве правил эвристической верификации – знаний о механизме полиморфизма сигнатур. Работа данного метода основывается на наборе эвристик, то есть предположений, статистическая значимость которых подтверждена опытным путем, о характерных признаках вредоносного и, наоборот, безопасного исполняемого кода. Каждый признак имеет определенный вес, то есть число, показывающее важность и достоверность этого признака. Если признак указывает на наличие вредоносного кода, то вес оценивается как положительный, а если на наличие безопасного кода, то вес оценивается как отрицательный. На основании суммарного веса, характеризующего содержимое объекта, эвристический анализатор вычисляет вероятность содержания в нём неизвестного вредоносного объекта. Если эта вероятность превышает некоторое пороговое значение, то исследуемый объект объявляется вредоносным. Следует отметить, что поскольку этот метод базируется на некоторых эмпирических предположениях, при его использовании существует вероятность ложного срабатывания;

3) Обнаружение аномалий – метод обнаружения вредоносных программ, основанный на выявлении необычных и подозрительных событий в наблюдаемой системе. Так, например, если программа попробует записать какие-то данные в исполняемый файл, антивирус, использующий этот метод, может отметить зафиксированное действие как небезопасное. В отличие от предыдущих методов, метод обнаружения подозрительного поведения позволяет гарантированно обнаружить совершенно новые вирусы, которых еще нет ни в одной вирусной базе. Однако этот метод также может выдавать большое количество ложных срабатываний, что делает пользователя маловосприимчивым к подобным предупреждениям;

4) Обнаружение изменений – метод обнаружения вредоносных программ, основанный на выявлении изменений, вызываемых вирусами в системе. Работа данного метода базируется на использовании систем контроля целостности. Программы, использующие метод обнаружения изменений, периодически сканируют содержимое дисков компьютера, записывая в свою базу данных контрольные суммы файлов и критически важных внутренних областей файловых систем. При этом при сканировании новые значения контрольных сумм сравниваются со старыми значениями, и если при сравнении обнаруживаются изменения, программа сообщает об этом пользователю. Этот метод часто используют совместно с сигнатурным и эвристическим анализом, направляя антивирусную программу только на файлы и каталоги, в которых произошли изменения;

5) Вакцинирование программ – метод обнаружения вредоносных программ, принцип работы которого заключается в присоединении к исполняемому файлу специального модуля контроля, который будет следить за целостностью этого файла. Проверке при использовании данного метода подлежат любые характеристики файла, например, его контрольная сумма. При заражении вредоносной программой вакцинированного файла модуль контроля обнаруживает изменения и сообщает об этом пользователю.

Аппаратно-программные методы основаны на реализации одного или нескольких из указанных выше методов защиты с использованием технических устройств и представляют собой один из самых надежных способов защиты программных средств от вредоносного кода. Имея полный контроль над всеми обращениями к дисковой подсистеме компьютера, аппаратно-программный комплекс при необходимости может не только сообщить о каких-либо нарушениях пользователю, но и заблокировать дальнейшую работу компьютера.

##
###

99. Угрозы информационной безопасности программного обеспечения. Модели безопасности информационных систем.

Угрозами информационной безопасности называются потенциальные источники нежелательных событий, которые могут нанести ущерб ресурсам информационной системы. Безопасность ПО в широком смысле является свойством данного ПО функционировать без проявления различных негативных последствий для конкретной компьютерной системы.

Обобщенная классификация угроз информационной безопасности программного обеспечения КС может выглядеть следующим образом:

- вредоносные программы - программы, используемые для осуществления несанкционированного доступа к информации и (или) воздействия на информацию или ресурсы автоматизированной информационной системы;
- программные закладки - программные компоненты, заранее внедряемые в компьютерные системы, которые по сигналу или в установленное время приводятся в действие, уничтожая или искажая информацию, или дезорганизуя работу программно-технических средств;
- способы и средства, позволяющие внедрять вредоносные программы и программные закладки в компьютерные системы и управлять ими на расстоянии.

В настоящее время одним из наиболее опасных средств информационного воздействия на компьютерные системы является использование вредоносных программ (вирусы, черви, сетевые снифферы...). В качестве основных средств вредоносного (деструктивного) воздействия на КС необходимо, наряду с вредоносными программами, рассматривать алгоритмические и программные закладки.
Алгоритмическая закладка - преднамеренное (или случайное) искажение какой-либо части алгоритма, либо построение его таким образом, что в результате конечной программной реализации этого алгоритма программа будет иметь ограничения на выполнение требуемых функций или вовсе не выполнять их при определенных условиях.
Программная закладка - совокупность операторов и (или) операндов, преднамеренно (или случайно) в завуалированной форме включаемая в состав выполняемого кода программного компонента на любом этапе его разработки. Пример: зашитые в программу учетные данные.

Действия алгоритмических и программных закладок условно можно разделить на три класса:

- изменение функционирования вычислительной системы (сети);
- несанкционированное считывание информации;
- несанкционированная модификация информации, вплоть до ее уничтожения.

Указанные классы воздействий могут пересекаться.

С точки зрения времени внесения программных закладок в программы их можно разделить на две категории:
- «врожденные», то есть закладки, внесенные при разработке ПО;
- «приобретенные», то есть закладки, внесенные при испытаниях, эксплуатации или модернизации ПО.

--- Дальше описывается лабуда из ГОСТ Р ИСО/МЭК 15408 ---
Часть 1 "Введение и общая модель" является введением в ИСО/МЭК 15408. В ней определяются общие понятия и принципы оценки безопасности ИТ и приводится общая модель оценки.

Объект оценки - совокупность программного, программно-аппаратного и/или аппаратного обеспечения, возможно сопровождаемая руководствами.
Модель политики безопасности объекта оценки - структурированное представление политики безопасности, которая должна быть осуществлена объектом оценки.
К нарушениям безопасности обычно относят:
- раскрытие актива несанкционированным получателем, наносящее ущерб (потеря конфиденциальности);
- ущерб активу вследствие несанкционированной модификации (потеря целостности);
- несанкционированное лишение доступа к активу (потеря доступности).

Компонент - наименьшая выбираемая совокупность элементов, на которой могут основываться требования.
Семейство - совокупность компонентов, которые направлены на достижение сходной цели, но отличаются акцентами или строгостью.
Класс – совокупность семейств, объединенных общим назначением.

Компоненты считаются объединенными друг с другом комплиментарно (могут дополнять друг друга) или связью замещения (ужесточение требований безопасности, когда i-ый компонент должен включать в себя все предыдущие).
Содержание компонентов является довольно гибким за счет наличия операций:
1) итерация: позволяет неоднократно использовать компонент при различном выполнении в нем операций;
2) назначение: позволяет определять параметры;
3) выбор: позволяет выбирать один или более пунктов из перечня;
4) уточнение: позволяет осуществлять детализацию.

Модель безопасности:
Общие Критерии → Профиль Защиты → Задание по Безопасности

Состав профиля защиты:
1) задача по защите (среда и цели безопасности);
2) чем реализуем защиту;
3) обоснование.

Задание по безопасности: описание функций безопасности в ИС.
Создается для того, чтобы:
1) конкретизировать функции безопасности конкретной ИС;
2) предоставить органам аттестации и сертификации средства для проведения проверки ИС на соответствие определенным уровням безопасности.

Недостаток ОК: они направлены на описание требований безопасности информационной системы (оценку этих требований), а основной источник угроз – это сам человек.

##
###

100. Функциональные требования безопасности: методика формирования требований, реализация функциональных требований безопасности

Часть 2 "Функциональные компоненты безопасности" устанавливает совокупность функциональных компонентов, предназначенных для использования в качестве стандартных шаблонов, на основе которых следует устанавливать функциональные требования к ОО. ИСО/МЭК 15408-2 содержит каталог функциональных компонентов, систематизированных по семействам и классам.

В ОК представлены две различные категории требований безопасности – функциональные требования и требования доверия.
Функциональные требования безопасности – набор требований к функциям объекта информатизации, отвечающим за безопасность. Можно описать требования к самому программному продукту.
Функциональные требования налагаются на те функции ОО, которые предназначены для поддержания безопасности ИС и определяют желательный безопасный режим функционирования ОО. Функциональные требования определены в части 2 ОК.
Примерами функциональных требований являются требования к идентификации, аутентификации, аудиту безопасности, неотказуемости источника (невозможности отказа от факта отправления сообщения).
Предположение безопасности – это набор условий, в котором возможно соблюдение политики безопасности.
ФБО – функции безопасности объекта. Описать в терминах РД ОК среду объекта (персонал, внешние системы, которые взаимодействуют с объектом) нельзя.
Представление класса – комментарий (для каких целей может быть использован).
Семейство конкретизирует значение отдельных функций. Дается уникальное и краткое имя (пример: FAU_GEN – семейство генерации данных аудита безопасности).
Затем определяется структура семейства. Действия по управлению определяют, каким образом можно использовать данное семейство.
Существует 11 функциональных классов:

1) Аудит
2) Связь
3) Криптографическая поддержка
4) Защита данных пользователя
5) Идентификация и аутентификация
6) Управление безопасностью
7) Приватность
8) Защита ФБО
9) Использование ресурсов
10) Доступ к объекту оценки
11) Доверенный канал

##
###

101. Требования доверия к безопасности информационных систем: методика формирования требований, поддержание доверия к безопасности информационных систем и программных продуктов

В ОК представлены две различные категории требований безопасности – функциональные требования и требования доверия. Доверие – основа для уверенности в том, что продукт или система ИТ отвечают целям безопасности.
Основная концепция ИСО/МЭК 15408 - обеспечение доверия, основанное на оценке (активном исследовании) продукта ИТ, который должен соответствовать определенным критериям безопасности. Активное исследование – это оценка продукта или системы ИТ для определения его свойств безопасности. Оценка является традиционным способом достижения доверия, и она положена в основу ОК. Методы оценки могут, в частности, включать в себя:

а) анализ и проверку процессов и процедур;
б) проверку, что процессы и процедуры действительно применяются;
в) анализ соответствия между представлениями проекта ОО;
г) анализ соответствия каждого представления проекта ОО требованиям;
д) верификацию доказательств;
е) анализ руководств;
ж) анализ разработанных функциональных тестов и полученных результатов;
и) независимое функциональное тестирование;
к) анализ уязвимостей, включающий предположения о недостатках;
л) тестирование проникновения.

Классы доверия:

1) Управление конфигурацией (помогает обеспечить сохранение целостности ОО)
2) Поставка и эксплуатация (определяет требования к мерам, процедурам и стандартам, применяемым для безопасной поставки, установки и эксплуатации ОО, обеспечивая, чтобы безопасность ОО не нарушалась во время его распространения, установки и эксплуатации)
3) Разработка (определяет требования для пошагового уточнения ФБО, вплоть до фактической реализации)
4) Руководства (определяет требования, направленные на обеспечение понятности, достаточности и законченности эксплуатационной документации)
5) Поддержка жизненного цикла (определяет требования доверия посредством принятия для всех этапов разработки ОО четко определенной модели жизненного цикла, включая политики и процедуры устранения недостатков, правильное использование инструментальных средств и методов, а также меры безопасности для защиты среды разработки)
6) Тестирование (устанавливает требования к тестированию)
7) Оценка уязвимостей (определяет требования, направленные на идентификацию уязвимостей, которые могут быть активизированы)

Если объект оценки имеет функции безопасности, которые реализуются вероятностными или перестановочными механизмами (такими, как пароль или хэш-функция), то требования доверия могут определять, что заявленный минимальный уровень стойкости согласуется с целями безопасности. От каждой такой функции потребуется соответствие указанному минимальному уровню стойкости или, по меньшей мере, дополнительно определенной специальной метрике.
Степень доверия для заданной совокупности функциональных требований может меняться; это, как правило, выражается через возрастание уровня строгости, задаваемого компонентами доверия. Часть 3 ОК определяет требования доверия и шкалу оценочных

уровней доверия, формируемых с использованием этих компонентов. Требования доверия налагаются на действия разработчика, представленные свидетельства и действия оценщика. Примерами требований доверия являются требования к строгости процесса разработки, по поиску потенциальных уязвимостей и анализу их влияния на безопасность.

Иерархическая структура представления требований доверия (класс - семейство - компонент - элемент):
Требования доверия
- Классы доверия
  - Имя класса
  - Представление класса
  - Семейства доверия
    - Имя семейства
    - Цели
    - Ранжирование компонентов
    - Замечения по применению
    - Компоненты доверия
      - Идентификация компонента
      - Цели
      - Замечания по применению
      - Зависимости
      - Элементы доверия

##
###

102. Классификация технических каналов утечки информации.

Если распространение информации происходит с помощью технического средства, то соответствующий канал называется техническим каналом утечки информации.
Структура технического канала утечки информации:
Источник информации -> Канал связи -> Несанкционированный получатель информации
Канал связи = Источник сигнала -> Среда распространения -> Приемник сигнала

Классификация технических каналов утечки:
1) По виду носителя (оптические, акустические, радиоэлектронные, вещественные)
2) По структуре (одноканальные, составные)
3) По способу организации (случайные, организационные)
4) По времени функционирования (постоянные, эпизодические, случайные)
5) По степени сокрытия информации (открытые, технически закрытые, шифрованные)

Виды носителей информации:
1) Оптические (электромагнитное поле в диапазоне видимого света и ИК излучения)
2) Радиоэлектронные (электрическое, магнитное, электромагнитное поля в радиодиапазоне, электрический ток)
3) Акустические (акустические волны в инфразвуковом (<16 Гц), звуковом (16 Гц - 20 кГц) и ультразвуковом (свыше 20 кГц) диапазонах)
4) Вещественные (черновики документов, демаскирующие вещества, забракованные детали, по которым можно выявить выпускаемую продукцию)

@Акустические КУИ.

В акустическом канале утечки носителем информации от источника к несанкционированному получателю является акустическая волна в атмосфере, воде и твердой среде.
Источник акустического сигнала -> Среда распространения -> Приёмник акустического сигнала
Источниками акустического сигнала могут быть:
- говорящий человек или озвучивающее его речь звуковоспроизводящее устройство;
- механические узлы механизмов и машин, которые при работе издают акустические волны.

Источники сигналов характеризуются диапазоном частот, можностью излучения (Вт), интенсивностью излучения (Вт/м2), громкостью звука (дБ).
Аккустические волны характеризуются следующими свойствами:
- мощность (энергией);
- скорость распространения носителя;
- величина (коэффициент) затухания и поглощения;
- условия распространения акустической волны (коэффициент отражения от границ различных сред, дифракция).
Аккустическая волна в отличие от электромагнитной в большей степени поглощается в среде распространения, следовательно дальность акустического канала утечки информации мала, и как правило не обеспечиваеит возможности её съема за пределами контролируемой зоны.

@Структура оптического канала утечки информации
Объект наблюдения (источник сигнала) -> Среда распространения -> Оптический приёмник

Объект наблюдения (объект, отражающий внешний свет; объект, излучающий свет)
Среда распространения (воздух, космос (безвоздушное пространство), вода, оптическое волокно)
Оптический приёмник (визуально-оптический, фото-и киноаппараты, приборы ночного видения и тепловизоры, телевизионные средства наблюдения)
Объекты под действием солнечной радиации в течение дня по-разному отдают накопленное тепло в окружающее пространство. Различия в температуре излучения могут рассматриваться как демаскирующие признаки.
В видимой области прохождению света препятствуют абсорбирующие молекулы кислорода и воды.

@Радиоэлектронные каналы утечки информации.
Носителем информации является ток и электромагнитное поле с частотами колебаний от звукового диапазона до десятков ГГц.
Это наиболее информативный канал утечки в силу его особенностей:
- независимость функционирования канала от времени суток и года, существенно меньшая зависимость его параметров от метеоусловий;
- высокая достоверность добывания информации, особеннно при перехвате её в функциональных каналах связи (за исключением дезинформирования);
- оперативность получения информации вплоть до реального масштаба времени;
- скрытность перехвата сигналов и радиотеплового наблюдения.

В радиоэлектронном канале производится прехват радио и электросигналов, радиолокационное и радиотепловое наблюдение. Радиоэлектронные каналы утечки информации используют такие виды разведок, как: - радиоразведка

- радиотезническая разведка
- радиотепловая разведка
- радиолокационныя разведка

Структура радиоэлектронного канала утечки информации:
Входная информация -> [источник радио/электро сигналов] -> [среда распространения] -> [приёмник радио/эл сигналов] -> Выходная информация
В радиоэлектронных каналах утечки информации источниками сигналов могут быть:
- передающие устройства функциональных каналов связи;
- источники ПЭМИН;
- объекты, отражающие электромагнитные волны в радиодиапазоне;
- объекты, излучающие собственные электромагнитные волны в радиодиапазоне.

В зависимости от способа перехвата информации разделяют два вида радиоэлектронных каналов утечки информации:
1) Перехват информации, передаваемой по функиональному каналу связи: приёмник сигнала настраивается на параметры сигнала функционального радиоканала или подкючается к проводам соответствующего функционального канала. Такой канал имеет общий с функциональным каналом связи источник сигналов – передатчик и часть среды радиоканала и проводного функционального канала до точки подключения средства съема.
2) Радиоэлектронный канал утечки 2-го вида имеет собственный набор элементов: передатчик сигналов, среду распространения и приёмник сигналов. Передатчик этого канала утечки информации образуется случайно (без участия источника или получателя информации) или специально устанавливается в помещении злоумышленником. В качесиве такого передатчика применяются источники опасных сигналов и закладные устройства. Особенностями передатчиков являются малые амплитуда электросигналов и мощность радиосиганлов => протяженность таких каналов невелика и составляет десятки/сотни метров.
Электросигналы могут быть аналоговыми и дискретными. Наиболее широко применяются сигналы, ширина которых соответствует ширине спектра стандартного телефонного канала.
Различают воздушные и кабельные проводные линии связи. Кабельные линии связи составляют большинства телефонных линий Росии.

##
###

103. Виды и источники носителей защищенной информации.

Защищаемая информация - информация, являющаяся предметом собственности и подлежащая защите в соответствии с требованиями правовых документов или требованиями, устанавливаемыми обладателем информации. ГОСТ Р 50922-2006 "Защита информации. Основные термины и определения"
С точки зрения защиты информации ее источниками являются субъекты и объекты, от которых информация может поступить к несанкционированному получателю (злоумышленнику). Очевидно, что ценность этой информации определяется информированностью источника. Основными источниками информации являются следующие (из Торокина):
– люди;
– документы;

– продукция;
– измерительные датчики;
– интеллектуальные средства обработки информации;
– черновики и отходы производства;
– материалы и технологическое оборудование.

Основные объекты защищаемой информации можно объединить в следующие группы:
– собственники, владельцы и пользователи;
– носители и технические средства передачи и обработки информации;
– системы информатизации связи и управления, военная техника;
– объекты органов управления, военные и промышленные объекты.

В группе носителей и технических средств передачи и обработки информации защите подлежат следующие объекты:
– носители информации в виде информационных физических полей, химических сред, сигналов, документов на различных основах;
– средства вычислительной техники;
– средства связи;
– средства преобразования речевой информации;
– средства визуального отображения;
– средства размножения документов;
– вспомогательные технические средства, расположенные в помещении, где информация обрабатывается;
– помещения, выделенные для проведения мероприятий.

В интересах ЗИ о вооружении и военной технике защите подлежат:
– характеристики и параметры конкретных образцов вооружений и военной техники на всех этапах их жизненного цикла;
– научно-исследовательские, опытно-конструкторские и экспертные работы военно-прикладной направленности.

Для объектов органов управления, военных промышленных объектов защите подлежит следующая информация:
– о местоположении объекта;
– о предназначении, структуре объекта и режимах его функционирования;
– информация, циркулирующая в технических средствах, используемых на объекте;
– информация о разрабатываемых и эксплуатационных образцах вооружения, военной техники и технологии;
– информация о научно-исследовательских и опытно-конструкторских работах.

##
###

104. Виды контроля и эффективности защиты информации

Контроль эффективности защиты - важное и необходимое направление работ по защите информации. Этот вид деятельности проводится прежде всего силами службы безопасности, а также руководителями структурных подразделений. Контроль инженерно-технической защиты является составной частью контроля зашиты информации в организации и заключается, прежде всего, в определении (измерении) показателей эффективности защиты техническими средствами и сравнении их с нормативными.

Виды контроля (по периодичности):
1) Предварительный
Предварительный контроль проводится при любых изменениях функционирования системы защиты информации, в том числе: после установки нового технического средства защиты или изменении организационных мер; после проведения профилактических и ремонтных работ средств защиты; после устранения выявленных нарушений в системе защиты.
2) Периодический
Периодический контроль осуществляется с целью обеспечения систематического наблюдения за уровнем защиты. Он проводится выборочно (применительно к отдельным темам работ, структурным подразделениям или всей организации) по планам, утвержденным руководителем организации, а также вышестоящими органами.
3) Постоянный
Постоянный контроль осуществляется выборочно силами службы безопасности и привлекаемых сотрудников организации с целью объективной оценки уровня защиты информации и выявления слабых мест в системе защиты информации. Кроме того, такой контроль оказывает психологическое влияние на сотрудников, вынуждая их работать качественнее.

Виды контроля (по составу работ):
1) Организационный
Проверка соответствия мероприятий по технической защите информации требованиям руководящих документов.
2) Технический
Контроль эффективности технической защиты информации, проводимый с использованием технических средств контроля.

Методы технического контроля эффективности защиты информации:
1) Инструментальный
Инструментальные методы контроля обеспечивают наиболее точные результаты, так как они реализуются с помощью средств измерительной техники в местах контроля, прежде всего на границе контролируемой зоны. Так как измеряемые уровни опасных сигналов сравнимы с уровнями шумов, то для инструментального контроля необходимы высокочувствительные дорогостоящие измерительные приборы.
2) Инструментально-расчетный
Инструментально-расчетный технический контроль позволяет снизить требования к параметрам измерительной техники. Эти методы предполагают проведение измерений не на границе контролируемой зоны, а вблизи возможных источников сигналов. Возле источников сигналов уровни сигналов выше и, соответственно, требования к чувствительности измерительных приборов ниже. Уровни же сигналов в местах проведения контроля рассчитываются по соответствующим методикам расчета. Так как в качестве исходных данных для расчета применяются результаты измерений, то точность контроля будет определяться точностью измерений и используемого математического аппарата.
3) Расчетный
Если отсутствуют требуемые для инструментального или инструментально-расчетного контроля измерительные приборы, то осуществляется расчетный технический контроль путем проведения расчетов по априорным или справочным исходным данным.

При проведении технического контроля проверяется эффективность принятых мер специальной защиты объектов от утечки информации за счет:

- побочных электромагнитных излучений;
- наводок побочных электромагнитных излучений на различные проводники и другие токопроводящие конструкции;
- высокочастотного навязывания;
- электроакустических преобразований;
- акустических колебаний;
- неравномерности потребления тока в сети электропитания.

[Из чужих ответов, может пригодится]
Средства радиоконтроля помещения предназначены для обнаружения закладных устройств, излучающих радиоволны во время их поиска.
К обнаружителям радиоизлучений закладных устройств относятся:
- обнаружители электромагнитных полей (индикаторы поля и частотометры). Индикаторы поля световым и звуковым сигналом информируют оператора о наличии в месте расположения антенны индикатора элм-ого поля с напряженностью выше фоновой. Часотометры обеспечивают, кроме того, измерение частоты колебаний поля. Чувствительность обнаружителей поля мала, поэтому с их помощью можно обнаружить поля радиозакладок в непосредственной близости от источника излучения;
- сканирующие приемники (в блоках памяти этих приемников можно запомнить частоты сигналов, о которых достоверно известно, что они не принадлежат закладным устройствам). Эти приемники имеют высокие электрические параметры в широком диапазоне частот настройки, перекрывающем частоты радиоизлучений иеющихся на рынке закладок. Эти приемники автоматически последовательно настраиваются на частоты радиосигналов во всем диапазоне. Оператор, прослушивая звуковые сигналы на выходе приемника на каждой из частот, принимает решение о продолжении или прекращении поиска. Для продолжения поиска он нажимает соотв. кнопку, передавая устройству управления приемника команду о перестройке на следующую частоту;
- специальные приемники используются для оперативного поиска закладок, они содержат в себе сканирующий приемник и излучатель акустического тестового сигнала и микропроцессор. Излучатель акустического сигнала имитирует источник акустической информации. Микропроцессор выявляет радиосигналы, на которые настраивается сканирующий приемник, по критерию «свой-чужой» и быстро обнаруживает радиосигнал закладки, если таковая имеется.;
- дальнейшее развитие специальных приемников привело к появлению автоматизированных программно-аппаратных комплексов для поиска средств негласного съема акустической информации.

Нелинейный локатор:
При работе НРЛ излучает высокочастотный сигнал, который легко проникает во многие материалы, мебель, может проходить (с ослаблением) через внутренние перегородки помещений, бетонные стены и полы, отражается от исследуемой поверхности и принимается приемником НРЛ. Существенное отличие заключается в том, что если приемник радиолокационной станции принимает отраженный от объекта эхо-сигнал на частоте излучаемого сигнала, то приемник НРЛ принимает кратные гармоники отраженного сигнала (2f, 3f). В результате нелинейного преобразования электрического сигнала, индуцируемого в элементах схемы ЗУ высокочастотным полем локатора, образуется сигнал, в спектре которого присутствуют, кроме основной частоты, ее кратные гармоники с частотами 2f, 3f и т. д. Так как амплитуда гармоник резко убывает с увеличением ее номера, то при работе НРЛ используют 2-ю и 3-ю гармоники. При этом амплитуды гармоник во многом зависят от характера нелинейности электрорадиоэлементов, входящих в состав ЗУ, и мощности излученного электромагнитного поля.

Обнаружители пустот:
Так как в пустотах сплошных сред (кирпичных и бетонных стенах, деревянных конструкциях и т.д.) могут останавливтаься долговременные дистанционно-управляемые ЗУ, то можно использовать технические средства обнаружения пустот, которые позволяют повысить вероятность выявления пустот для дальнейшего их исследования. В качестве таких средств могут применяться различные ультразвуковые приборы.

Металлодетекторы:
Обнаруживают ЗУ по магнитным и электрическим свойствам их элементов. Любая закладка содержит токопроводящие элементы: резисторы, индуктивности, соединительные токопроводники, анетнну, корпус элементов питания.

Рентгеновские установки:
Используются для интерскопии предметов непонятного назначения.

Средства подавления сигналов ЗУ:
Генераторы помех (линейного и пространственного зашумления). Выходы генератора линейного зашумления соединяются с проводами телефонной линии и электросети и в них подаются электрические сигналы, перекрывающие опасные сигналы по спектру и мощности. Генераторы пространственного зашумления повышают уровень электромагнитных помех помех в помещении и следовательно на входе злоумышленника. Для эффективного подавления сигнала закладки уровень помехи в полосе спектра сигнала должен в несколько раз превышать уровень сигнала.

##
###

105. Оценка угроз акустических каналов утечки информации. Непреднамеренное прослушивание. Технические средства контроля звукоизоляции ограждающих конструкций.

[Взял из старых ответов, лучше ничего не нашел]

Оценка угроз акустических каналов утечки информации:
Сначала происходит анализ того, по каким каналам возможна утечка информации. Виды акустических каналов утечки информации :
- непреднамеренное прослушивание (все, что связано с полом, потолком, стенами, дверью);
- акустоэлектрические преобразования (перехват с линий связи);
- вибро-акустический канал (батареи -> экранировать, окна -> закрыть жалюзи);
- направленные микрофоны (оценить расположение других зданий, контролируемой зоны, окон и т.д.).

Рассматривается помещение: двери, стены, пол, потолок, окна.

При непреднамеренном прослушивании утечка наиболее вероятна через двери, однако подтвердить это предположение можно только в результате контрольно измерительной экспертизы.
При акустоэлектрическом канале утечки определяется, какая техника находится в помещении. Визуально этот канал утечки не определить, только путем технических

проверок и замеров. Часто акустоэлектрическим каналом утечки являются системные блоки. Трубы и металл хорошо воспринимают акустическую волну.
При виброакустике проверяется воздействие на инженерно-технические средства. Утечка возможна только в том случае, если рядом находится помещение сторонней организации, которая заинтересована в получении информации.
Направленные микрофоны:
Оценивается окно, вся территория корпуса (находится ли он на контролируемой территории или нет). Если окно защищено рольставнями и двойным стеклопакетом, то техническая утечка информации по данному каналу невозможна.

Технические средства контроля звукоизоляции ограждающих конструкций:
Инструментальный контроль акустической защищенности помещения проводится при помощи специальной измерительной аппаратуры. В ее состав входят шумомер и экранированная акустическая колонка с усилителем. Например, если мы проверяем дверь, то необходимо замерить уровень шума за дверью, уровень сигнала в помещении и уровень сигнал/шум за дверью. Шум замеряется всегда по минимуму без голосов и топота. Затем, в зависимости от того является помещение выделенным или защищаемым, проводятся специальные вычисления на основе полученных результатов и делается вывод о том, соответствует помещение предъявленным нормативным документами нормам или нет. Если не соответствует, то можно применить активные средства защиты (например, генераторы шума и виброизлучатели, которые устанавливаются на поверхность ограждающих конструкций).

##
###

106. Порядок и методика аттестации защищаемых помещений.

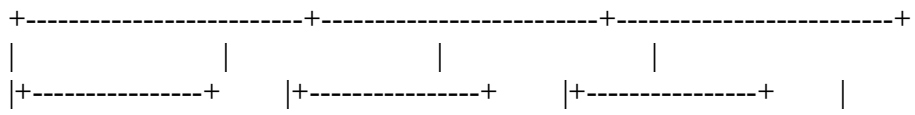
Составляется заявка в соответствующий орган. При этом, если помещение защищаемое, то речь идет о защите конфиденциальной информации, и заявка отправляется в местный орган; если же помещение выделенное, то речь идет о защите государственной тайны, и заявка пишется в контролирующий орган (Нижний Новгород?).
Порядок аттестации:
1) Предварительное ознакомление с составом, структурой и организацией эксплуатации объектов информатизации:
- анализ документов, определяющих состав и порядок эксплуатации объектов:
- проверка соответствия предоставленных данных тому, что есть в действительности, анализ информационных потоков;
2) Проверка объекта информатизации на соответствие организационно-техническим требованиям по защите информации:
- проверка достаточности представленных документов и соответствия их содержания требованиям по безопасности информации;
- проверка уровня подготовки кадров и распределения ответственности персонала;
- проверка выполнения требований по безопасности информации к помещениям, в которых производится обработка информации.
3) Проведение испытаний объекта информатизации на соответствие требованиям по защите информации от утечки по техническим каналам
4) Проведение комплексных испытаний с целью оценки соответствия использованного комплекса мер и средств защиты требуемому уровню безопасности информации
5) Подготовка отчетной документации и оценка результатов испытаний аттестуемого объекта;

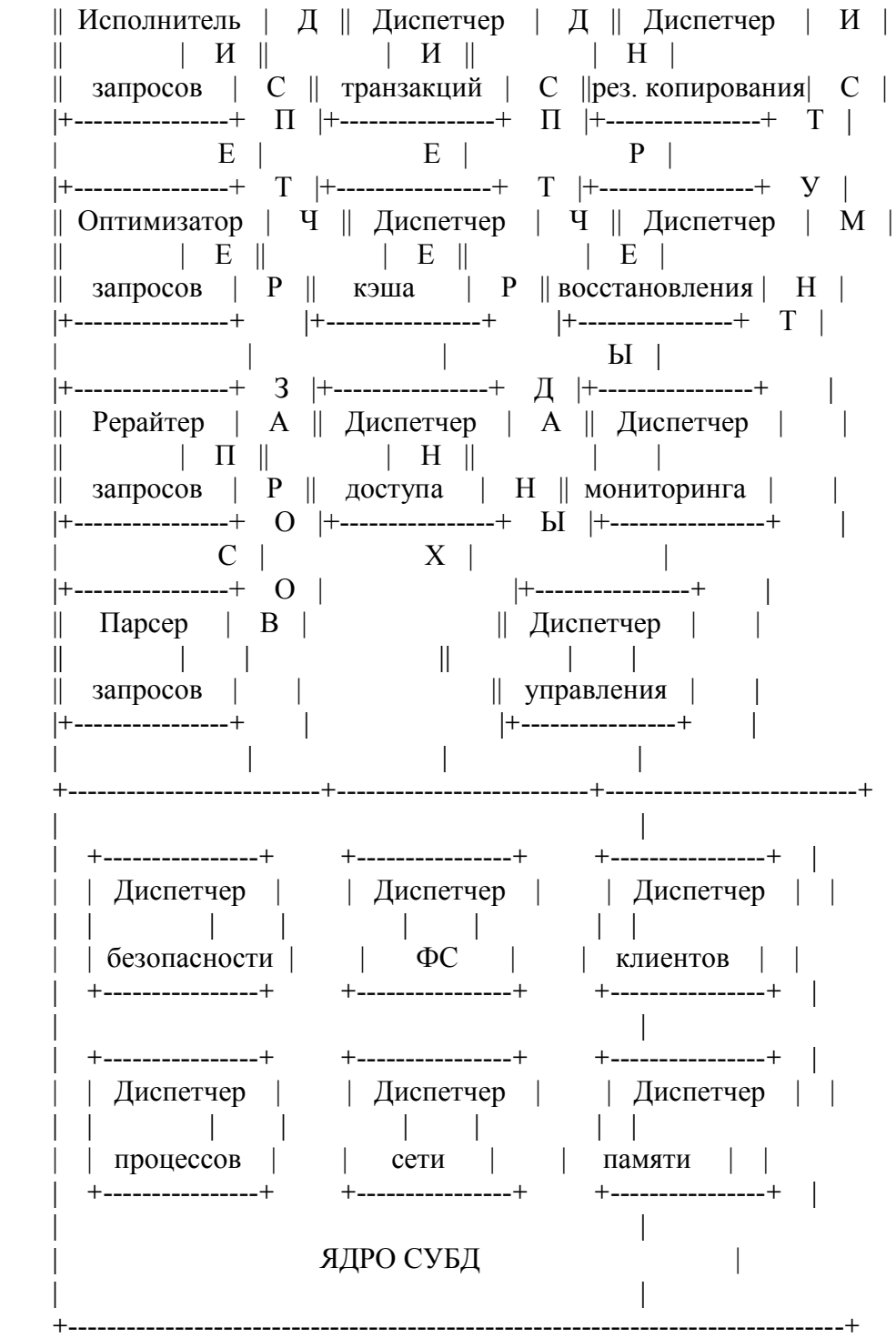
6) Продолжительность работ по аттестации выделенных помещений определена в соответствии с нормативными документами ФСТЭК России. Продолжительность работ может быть увеличена в случае несвоевременного устранения заявителем выявленных в ходе аттестации недостатков и нарушений.
На основании полученных результатов испытаний принимается заключение, включающее:
- оценку соответствия выделенного помещения требованиям по безопасности информации;
- перечень выявленных недостатков и нарушений;
- рекомендации по устранению выявленных недостатков и нарушений;
- вывод о возможности (невозможности) выдачи «Аттестата соответствия».

Методика аттестации:
1) Анализ полноты исходных данных, проверка их соответствия реальным условиям размещения, монтажа и эксплуатации технических средств защиты информации
Для проведения испытаний заявитель представляет аттестационной комиссии следующие исходные данные и документацию:
- оформленный технический паспорт на ВП (выделенное помещение);
- состав технических средств, установленных в ВП;
- акт категорирования ВП;
- состав и схемы размещения средств защиты информации;
- инструкции по эксплуатации средств защиты информации;
- планы размещения технических средств;
- предписания на эксплуатацию ВТСС (вспомогательные технические средства и системы);
- протоколы специальных исследований ВТСС;
- акты или заключения о специальной проверке ВТСС;
- план контролируемой зоны;
2) Проверка состояния организации работ и выполнения организационно-технических требований по защите информации, оценка правильности категорирования ВП, оценка полноты и уровня разработки организационно-распорядительной, проектной и эксплуатационной документации, оценка уровня подготовки кадров и распределения ответственности за выполнение требований по обеспечению безопасности информации
3) Проверка выполнения требований по защите ВП от утечки по акустическому и виброакустическому каналам
4) Проверка ВП на соответствие требованиям по защите информации от утечки за счет электроакустических преобразований и паразитной генерации
5) Проверка выполнения требований по защите ВП от утечки информации за счет внедренных в технические средства, установленные в ВП, специальных электронных устройств перехвата информации
6) Подготовка отчетной документации.

##
###

107. Архитектурные особенности и транзакционные модели современных СУБД.





Ядро СУБД:

1) Диспетчер процессов. Управляет пулом рабочих процессов/потоков (worker).
2) Диспетчер сети. Планирует и исполняет сетевые операции, чтобы добиться максимальной пропускной способности сети.
3) Диспетчер файловой системы. Планирует и исполняет операции файловой системы, чтобы добиться максимальной производительности.
Некоторые диспетчеры полностью заменяют файловую систему ОС.
4) Диспетчер памяти. Управляет оперативной памятью, используемой для кэширования данных.
5) Диспетчер безопасности. Управляет аутентификацией и авторизацией пользователей.

6) Диспетчер клиентов. Используется для управления соединениями с клиентами — веб-серверами или конечными пользователями/приложениями.
Диспетчер клиентов обеспечивает разные способы доступа к БД с помощью как всем известных API (JDBC, ODBC, OLE-DB и т.д.), так и с помощью проприетарных.

Инструменты:

1) Диспетчер резервного копирования. Выполняет резервное копирование.
2) Диспетчер восстановления. Используется для восстановления когерентного состояния базы данных после сбоя.
3) Диспетчер мониторинга. Занимается протоколированием всех активностей внутри базы данных и используется для наблюдения за её состоянием.
4) Диспетчер управления. Хранит метаданные (вроде наименований и структуры таблиц) и используется для управления базами, схемами, табличными пространствами и т. д.

Диспетчер запросов:

1) Парсер запросов. Проверяет валидность запросов.
2) Рерайтер запросов. Осуществляет предварительную оптимизацию.
3) Оптимизатор запросов.
4) Исполнитель запросов. Компилирует и исполняет запросы.

Диспетчер данных:

1) Диспетчер транзакций. Занимается обработкой транзакций.
В его обязанности входит отслеживание, чтобы каждый запрос исполнялся с помощью собственной транзакции.
2) Диспетчер кэша. Используется для отправки данных перед использованием и перед записью на диск.
Вместо того, чтобы получать данные напрямую от файловой системы, исполнитель запросов обращается за ними к диспетчеру кэша.
Тот использует содержащийся в памяти буферный пул, что позволяет радикально увеличить производительность БД.
3) Диспетчер доступа к данным. Управляет доступом к данным в дисковой подсистеме.

Транзакционные модели современных СУБД.

ACID-транзакция (Atomicity, Consistency, Isolation, Durability) - элементарная операция, удовлетворяющая четырём условиям:

1) Атомарность (Atomicity). Нет никакой более мелкой операции.
В случае неудачного выполнения транзакции система возвращается в состояние «до», то есть транзакция откатывается.
2) Согласованность (Consistency). В базу данных записываются только валидные данные (с точки зрения реляционных и функциональных связей).
3) Изолированность (Isolation). Если в одно время выполняются две транзакции A и B, то их результат не должен зависеть от того, завершилась ли одна из них до,
во время или после исполнения другой.
4) Надёжность (Durability). Когда транзакция зафиксирована (commited), то есть успешно завершена,
использовавшиеся ею данные остаются в базе данных вне зависимости от возможных происшествий (ошибок, сбоев).

Многие СУБД не обеспечивают полную изолированность по умолчанию, поскольку это приводит к огромным издержкам в производительности.
В SQL используется 4 уровня изолированности:

1) Сериализуемые транзакции (Serializable). Наивысший уровень изолированности.
По умолчанию используется в SQLite. Каждая транзакция исполняется в собственной, полностью изолированной среде.

2) Повторяемое чтение (Repeatable read). По умолчанию используется в MySQL.
Каждая транзакция имеет свою среду, за исключением одной ситуации: если транзакция добавляет новые данные и успешно завершается,
то они будут видимы для других, всё ещё выполняющихся транзакций. Но если транзакция модифицирует данные и успешно завершается,
то эти изменения будут не видны для всё ещё выполняющихся транзакций. То есть для новых данных принцип изолированности нарушается.

Например, транзакция А выполняет:

```
SELECT count(1) from TABLE_X
```

Потом транзакция Б добавляет в таблицу X и коммитит новые данные. И если после этого транзакция А снова выполняет count(1),
то результат будет уже другим. Это называется фантомным чтением.

3) Чтение зафиксированных данных (Read commited). По умолчанию используется в Oracle, PostgreSQL и SQL Server.
Это то же самое, что и повторяемое чтение, но с дополнительным нарушением изолированности.
Допустим, транзакция А читает данные; затем они модифицируются или удаляются транзакцией Б, которая коммитит эти действия.
Если А снова считает эти данные, то она увидит изменения (или факт удаления), сделанные Б.

Это называется неповторяемым чтением (non-repeatable read).

4) Чтение незафиксированных данных (Read uncommited). Самый низкий уровень изолированности.
К чтению зафиксированных данных добавляется новое нарушение изолированности.
Допустим, транзакция А читает данные;
затем они модифицируются транзакцией Б (изменения не коммитятся, Б всё ещё выполняется).
Если А считает данные снова, то увидит сделанные изменения.
Если же Б будет откачена назад, то при повторном чтении А не увидит изменений, словно ничего и не было.

Это называется грязным чтением.

##
###

108. Разграничение доступа в современных СУБД.

Полномочия – это права на выполнение конкретного типа SQL-оператора или на доступ к объекту базы данных, принадлежащему другому пользователю. В базе данных Oracle необходимо явно предоставить пользователю полномочия для выполнения любых действий, включая подключение к базе данных или выборку, изменение и обновление данных в любой таблице, кроме собственной.
Существуют два основных типа полномочий Oracle: системные полномочия и объектные полномочия. Для предоставления пользователям как системных, так и объектынх полномочий служит оператор GRANT.
1) Системные полномочия позволяют пользователю выполнить конкретное действие в базе данных либо действие с любым объектом схемы, конкретного типа. Хороший пример первого типа системных полномочий – полномочия, которые позволяют подключаться к базе данных, носящие название полномочий CONNECT. Другими полномочиями этого типа являются полномоичия CREATE TABLESPACE, CREATE USER, DROP USER и ALTER USER. Второй класс системных полномоичий предоставляет пользователям право на выполнение операций, которыевлияют на объекты в любой схеме. Примерами этого типа системных полномочий служат ANALYZE ANY TABLE, GRANT ANY PRIVILEGE, INSERT ANY TABLE, DELETE ANY TABLE и т.п.
2) Объектные полномочия – это полномочия по отношению к различным типам объектов базы данных. Объектныые полномочия дают пользователю возможность выполнять действия с конкретной таблицей, представлением, материализованным представлением, последовательностью, процедурой, функцией или пакетом.

Хотя полномочиями пользователей достаточно легко управлять, непосредственно выдавая и отзывая их, эта задача может быстро стать чрезвычайно трудоемкой по мере добавления новых пользователей и увеличения количества объектов. Спустя некоторое время очень трудно отслеживать текущие полномочия каждого пользователя. Oracle решает эту проблему посредством применения ролей, представляющих собой именованные наборы прав, которые могут быть присвоены пользователям.
Роли можно считать набором прав, которые можно назначать и отзывать с помощью единственной команды GRANT или REVOKE. Роль может содержать как набор прав, так и другие роли. Роли облегчают присвоение нескольких прав пользователю. Заданная по умолчанию роль — это роль, которая автоматически вступает в действие, когда пользователь создает сеанс. Пользователю можно присваивать более одной роли по умолчанию.

##
###

109. Резервное копирование, восстановление и ремонт баз данных.

Потеря данных может быть классифицирована как:
1) Физическая потеря данных - происходит на уровне ОС и представляет собой потерю таких физических объектов, как файлов данных, управляющих файлов, журналов.
2) Логическая потеря данных - происходит на уровне объектов базы данных и представляет собой потерю логических объектов базы данных, таких как таблицы, индексы, строки в таблице.

Чтобы избежать как физической, так и логической потери данных, современные СУБД располагают средствами физического и логического резевреного копирования.

1) Физическое копирование предполагает создание копий файлов базы данных и архивных журнальных файлов. Физические резервные копии являются непереносимыми, то есть должны использоваться для восстановления базы данных на той же машине и в той же версии базы данных. Физическое копирование подразделяется на "горячее" и "холодное" копирование:

- "Горячее" копирование подразумевает создание резервной копии базы данных в тот момент, когда она находится в рабочем состоянии. Этот способ наиболее удобен, если база данных должна работать в режиме 24 на 7. Без особой необходимости пользоваться этим методом резервного копирования не рекомендуется, т. к. он не очень надежен и может привести к потере данных;
- "Холодное" копирование производится при отключенной базе данных и позволяет получить наиболее полную резервную копию. При режиме работы 24 на 7 этот способ не подходит, т. к. при «холодном» резервном копировании база данных должна быть выключена, и, следовательно, недоступна пользователям.

2) Логическое копирование предполагает создание и сохранение в файле операционной системы набора инструкций по воссозданию логических объектов базы данных, а также набора строк базы данных. Логическое копирование, как правило, применяется в тех случаях, когда требуется переместить конкретные данные в другую систему, отличающуюся архитектурой, версией операционной системы или СУБД. Логическое копирование является «горячим», т.е. выполняется при открытой базе данных.

Физическое восстановление базы данных – это применение архивных и оперативных журналов для восстановления изменений данных, произошедших с момента последнего физического резервного копирования.

1) Полное восстановление - восстановление базы данных на текущий момент времени, которое требует применения всех архивных и оперативных журналов к файлам данных, восстановленным из резервной копии. Обычно полное восстановление применяется в случаях, когда потеряны файлы данных или управляющий файл. Администратор может восстановить базу данных целиком, отдельное табличное пространство или файл данных.

2) Неполное восстановление - восстановление базы данных на момент времени, предшествующий текущему. При таком восстановлении базы данных к файлам данных применяются не все архивные и оперативные журналы. Обычно неполное восстановление базы данных требуется в следующих случаях:

- в результате сбоя носителя потеряны все или несколько оперативных журналов;
- в результате пользовательской ошибки произошла потеря данных (например, пользователь случайно удалил таблицу);
- невозможно выполнить полное восстановление базы данных, т. к. потеряны архивные журналы;
- потерян текущий управляющий файл, поэтому для открытия базы данных необходимо использовать резервную копию управляющего файла.

Существует три типа неполного восстановления:

- восстановление до отмены – журналы применяются до тех пор, пока администратор не введет команду CANCEL;
- восстановление до заданного времени – данные восстанавливаются до заданной точки во времени;
- восстановление до заданного системного номера – данные восстанавливаются до заданного системного номера.

##
###

110. Авторизация и аутентификация. Аппаратные средства идентификации пользователей.

Аутентификация - проверка принадлежности субъекту доступа предъявленного им идентификатора; подтверждение подлинности.
Авторизация — предоставление определённому лицу или группе лиц прав на выполнение определённых действий; а также процесс проверки (подтверждения) данных прав при попытке выполнения этих действий.

Один из способов аутентификации в компьютерной системе состоит во вводе пользовательского идентификатора, называемого логином, и пароля — некой конфиденциальной информации, знание которой обеспечивает владение определенным ресурсом. Получив введенный пользователем логин и пароль, компьютер сравнивает их со значением, которое хранится в специальной базе данных и, в случае совпадения, пропускает пользователя в систему. Однако более надёжным способом хранения аутентификационных данных признано использование специальных аппаратных средств (компонентов).

@Secret Net:
Система защиты информации Secret Net производится компанией «Код безопасности», входящей в группу компаний «Информзащита».
Система Secret Net предназначена для предотвращения несанкционированного доступа к рабочим станциям и серверам, работающим в гетерогенных локальных вычислительных сетях под управлением ОС MS Windows. Secret Net дополняет своими защитными механизмами стандартные защитные средства операционных систем и тем самым повышает защищенность всей автоматизированной информационной системы в целом.
Применение системы защиты информации Secret Net обеспечивает:
1) Разграничение доступа
- усиленную идентификацию и аутентификацию, которая осуществляется с помощью средств аппаратной поддержки при входе пользователя в систему; в качестве идентификаторов могут использоваться iButton и eTokenPro;
- полномочное управление доступом, осуществляемое на основе категорий конфиденциальности и прав допуска пользователей;
- разграничение доступа к устройствам с целью предотвращения несанкционированного копирования информации с защищаемого компьютера;
2) Доверенная информационная среда
- защиту от загрузки с внешних носителей, позволяющую при использовании средств аппаратной поддержки запретить пользователю загрузку ОС с внешних съемных носителей;
- замкнутую программную среду, в которой перечень программ, разрешенных для запуска, формируется для каждого пользователя компьютера (индивидуально или на уровне групп пользователей);
- контроль целостности программ и данных, обеспечивающий защиту от модификации файлов, каталогов, элементов системного реестра и секторов дисков;
- контроль аппаратной конфигурации, позволяющий заблокировать компьютера в случае обнаружения изменений;
- функциональный самоконтроль подсистем перед входом пользователя в систему;
3) Защита информации в процессе хранения
- прозрачное шифрование файлов по алгоритму ГОСТ Р 34.12-2015 "Кузнечик";
- контроль печати с возможностью маркировки листов в соответствии с принятыми в организации стандартами;

- гарантированное уничтожение данных путем записи случайной последовательности на место удаляемой информации в освобождаемую область диска;
- оперативный мониторинг и аудит, позволяющий администратору безопасности своевременно реагировать на факты и попытки НСД.

4) Удобство управления
- удобную настройку механизмов защиты на всех защищаемых компьютерах посредствам сохранения эталонов настроек для сетевого варианта;
- централизованное управление, интегрированное с доменом Microsoft Active Directory;

В Secret Net предусмотрено несколько режимов работы средств аутентификации:
- "Стандартный" — пользователь может войти в систему, выполнив ввод имени и пароля;
- "Смешанный" — пользователь может войти в систему, выполнив ввод имени и пароля, а также может использовать персональный идентификатор, поддерживаемый системой Secret Net;
- "Только по идентификатору" — каждый пользователь для входа в систему должен обязательно использовать персональный идентификатор, поддерживаемый системой Secret Net 5.0.

@Аккорд:
СЗИ НСД Аккорд-АМДЗ – это аппаратный модуль доверенной загрузки (АМДЗ) для IBM-совместимых ПК – серверов и рабочих станций локальной сети, обеспечивающий защиту устройств и информационных ресурсов от несанкционированного доступа.
«Доверенная загрузка» – это загрузка различных операционных систем только с заранее определенных постоянных носителей (например, только с жесткого диска) после успешного завершения специальных процедур: проверки целостности технических и программных средств ПК (с использованием механизма пошагового контроля целостности) и идентификации/аутентификации пользователя.
Комплекс начинает работу сразу после выполнения штатного BIOS компьютера – до загрузки операционной системы, и обеспечивает доверенную загрузку ОС, поддерживающих различные файловые системы. В Аккорд также реализована возможность отключения питания компьютера в случае, если за N секунд не начал работу BIOS АМДЗ. Аккорд-АМДЗ позволяет использовать для идентификации пользователей смарт-карты, устройства iButton, устройства считывания отпечатков пальцев, а также устройство ШИПКА.
Особенностью комплекса "Аккорд-NT/2000" v.3.0. является проведение на аппаратном уровне процедур идентификации и аутентификации до начала загрузки операционной системы. Это обеспечивается при помощи программного обеспечения, записанного в энергонезависимой флэш-памяти платы контроллера "Аккорд-АМДЗ". Встроенное ПО "Аккорд-АМДЗ" получает управление на себя во время так называемой процедуры ROM-Scan, суть которой заключается в следующем: в процессе начального старта после проверки основного оборудования BIOS компьютера начинает поиск внешних ПЗУ в некотором диапазоне. Признаком наличия ПЗУ является наличие определенного сегмента в первом слове проверяемого интервала. Если данный признак обнаружен, то в следующем байте содержится длина ПЗУ. Затем вычисляется контрольная сумма всего ПЗУ, и если она корректна - будет произведен вызов процедуры, расположенной в ПЗУ. Такая процедура обычно используется для инициализации BIOS плат расширения. СЗИ «Аккорд» в этой процедуре проводит идентификация и аутентификация пользователя, контроль целостности аппаратной части ПЭВМ, программ и данных. При любой ошибке возврат из процедуры не происходит, т.е. загрузка выполняться не будет.

@Dallas Lock:
Dallas Lock — система защиты информации от несанкционированного доступа в процессе её хранения и обработки. Представляет собой программный комплекс средств защиты информации в автоматизированных системах. Разработкой и созданием продуктов линейки Dallas Lock занимается Центр защиты информации ООО «Конфидент».
Возможности:
- СЗИ Dallas Lock запрещает посторонним лицам доступ к ресурсам ПК и позволяет разграничить права зарегистрированных в СЗИ пользователей при работе на компьютере. Разграничения касаются прав доступа к объектам файловой системы (дискам, папкам и файлам под файловой системой FAT или NTFS) и подключаемым устройствам. Для облегчения администрирования возможно объединение пользователей в группы;
- Для решения проблемы «простых паролей» СЗИ имеет гибкие настройки сложности паролей. Кроме того, в последних версиях системы имеется механизм генерации паролей, соответствующих всем установленным параметрам сложности;
- В СЗИ имеется возможность ограничения доступа пользователей к ПК по дате и времени;
- Используются два принципа контроля доступа к объектам файловой системы и подключаемым устройствам: дискреционный и мандатный;
- СЗИ позволяет настраивать замкнутую программную среду — режим, в котором пользователь может запускать только программы, определенные администратором.
- Для облегчения настройки замкнутой программной среды и мандатного доступа существуют механизмы:
  «мягкий режим» — режим, в котором, при обращении к ресурсу, доступ к которому запрещён, доступ всё равно разрешается, но в журнал событий заносится сообщение об ошибке;
  «режим обучения» — режим, в котором при обращении к ресурсу, доступ к которому запрещён, на этот ресурс автоматически назначаются выбранные администратором права;
  «неактивный режим» — режим, в котором возможно полное отключение подсистем СЗИ.
- Для опознавания пользователей служат индивидуальные пароли и аппаратные идентификаторы Touch Memory, eToken (так называемая двухфакторная аутентификация). При этом аппаратная идентификация не является обязательной: система может работать в полностью программном режиме;
- Запрос пароля и аппаратных идентификаторов происходит до начала загрузки ОС. Загрузка ОС возможна только после проверки идентификационных данных пользователя в СЗИ.

@Соболь:
Электронный замок «Соболь» - это аппаратно-программное средство защиты компьютера от несанкционированного доступа (аппаратно-программный модуль доверенной загрузки).
Для идентификации пользователей используются уникальные номера аппаратных устройств идентификации — персональных идентификаторов iButton. При аутентификации осуществляется проверка правильности указанного пользователем пароля с использованием аутентификатора пользователя. После включения компьютера комплекс "Соболь" запрашивает у пользователя его персональный идентификатор и пароль. Осуществляется проверка наличия в списке пользователей комплекса "Соболь" пользователя, которому принадлежит предъявленный персональный идентификатор. Если предъявлен персональный идентификатор, не зарегистрированный в системе:
- вход пользователя в систему запрещается;
- в журнале регистрации событий фиксируется попытка несанкционированного доступа к компьютеру.

Если указан пароль, не соответствующий предъявленному идентификатору:
- вход пользователя в систему запрещается;
- счетчик неудачных попыток входа пользователя в систему увеличивается на единицу;
- в журнале регистрации событий фиксируется попытка несанкционированного доступа к компьютеру.

Служебная информация о регистрации пользователя (имя, номер присвоенного персонального идентификатора и т.д.) хранится в энергонезависимой памяти комплекса "Соболь".

@Страж NT:
Система защиты информации «Страж NT» предназначена для комплексной защиты информационных ресурсов от несанкционированного доступа в многопользовательских автоматизированных системах на базе персональных ЭВМ, функционирующих под управлением операционных систем MS Windows.
Возможности (кратко):
- Поддержка современных операционных систем компании Microsoft;
- Контроль устройств, подключенных к компьютеру;
- Подсистема создания и применения шаблонов настроек;
- Помимо уже используемых ранее идентификаторов iButton и USB-ключей eToken, в новой версии реализована поддержка ключей Rutoken. Дополнительно реализована возможность использования в качестве идентификаторов пользователей флэш-накопителей;
- Добавлены новые функции работы с идентификаторами и возможность редактировать пользователей на удалённых рабочих станциях.

##
###

111. Контроль целостности аппаратных, программных ресурсов и гарантированное уничтожении информации.

@Secret Net:
Механизм контроля целостности осуществляет слежение за неизменностью контролируемых объектов с целью защиты их от модификации. Контроль проводится в автоматическом режиме в соответствии с некоторым заданным расписанием. Объектами контроля могут быть файлы, каталоги, элементы системного реестра и секторы дисков (последние только при использовании ПАК "Соболь"). Каждый тип объектов имеет свой набор контролируемых параметров. Так, файлы могут контролироваться на целостность содержимого, прав доступа, атрибутов, а также на их существование, т. е. на наличие файлов по заданному пути. Совокупность ресурсов, которые используются для решения определенных производственных задач, объединяется в группы ("задачи"). Для каждого из входящих в группу объектов рассчитываются эталонные значения контролируемых параметров.
В системе предусмотрена гибкая возможность выбора времени контроля. В частности, контроль может быть выполнен при загрузке ОС, при входе пользователя в систему, по заранее составленному расписанию. Для этого составляется задание на контроль.
При обнаружении несоответствия предусмотрены следующие варианты реакции на возникающие ситуации нарушения целостности:
- регистрация события в журнале Secret Net;
- блокировка компьютера;

- отклонение или принятие изменений.

Подсистема контроля аппаратной конфигурации компьютера предназначена для:

- своевременного обнаружения изменений в аппаратной конфигурации компьютера и реагирования на эти изменения;
- поддержания в актуальном состоянии списка устройств компьютера, который используется подсистемой разграничения доступа к устройствам.

Изменения аппаратной конфигурации компьютера могут быть вызваны подключением к компьютеру или отключением от него различных устройств, выходом устройств из строя и добавлением или заменой отдельных устройств.

Используются 2 метода контроля конфигурации:

- статический контроль конфигурации. Каждый раз при загрузке компьютера, а также при повторном входе пользователя подсистема получает информацию об актуальной аппаратной конфигурации и сравнивает ее с эталонной;
- динамический контроль конфигурации. Драйвер-фильтр устройств отслеживает факт подключения или изъятия устройства. При изменении конфигурации определяется класс устройства и выбирается реакция на изменение конфигурации.

Предусмотрено 2 вида реакций на изменения:

- регистрация события в журнале Secret Net;
- блокировка компьютера.

@Аккорд:

- контроль целостности системных областей диска, файлов ОС и прикладного ПО, разделов реестра Windows осуществляется на аппаратном уровне контроллером «Аккорд АМДЗ» до загрузки ОС;
- если процедура контроля из п. 1 выполнена успешно, то в момент запуска подсистемы разграничения доступа может выполняться контроль целостности файлов, по индивидуальному списку каждого пользователя. На диске хранится файл, содержащий перечень контролируемых файлов и эталонные значения хэш-функций. По этим данным определяется целостность каждого конкретного файла;
- целостность же самого этого файла на диске обеспечивается тем, что эталонное значение его хэш-функции, являющейся как бы интегральной хэш-функцией всех контролируемых файлов, вычисляется с использованием секретного ключа, который хранится в ТМ-идентификаторе пользователя. Этот ключ генерируется при регистрации ТМ-идентификатора с использованием датчика случайных чисел, установленного на плате контроллера, и для каждого пользователя является уникальным.

В комплексе "Аккорд" предусмотрен динамический контроль целостности исполняемых модулей (задач). Этот контроль выполняется при каждом запуске контролируемого модуля, независимо от того, выполняется ли эта операция пользователем, или ОС. Как и на других этапах контроля целостности, здесь применяется контроль с использованием хэш-функции.

Дополнительно в комплексе предусмотрен динамический контроль целостности собственно монитора разграничения доступа. Этот контроль выполняется периодически и обеспечивает дополнительный уровень защиты от случайных или преднамеренных покушений на отключение СЗИ.

@Dallas Lock:

Позволяет контролировать целостность файлов, папок и параметров компьютера (BIOS, CMOS, BOOT sector,MBR) до загрузки ОС, а также целостность файлов и папок при доступе. Для контроля целостности используются цифровые подписи, вычисленные по одному из алгоритмов на выбор: CRC32, MD5, ГОСТ Р 34.11-2012.

@Соболь:

Подсистема контроля целостности обеспечивает контроль целостности файлов на жестком диске и физических секторов жесткого диска.
Подсистема включает в себя следующие компоненты:
- модуль контроля целостности является программным модулем расширения BIOS комплекса "Соболь". Он обеспечивает расчет эталонных значений контрольных сумм проверяемых объектов, сохранение полученных контрольных сумм в файлах заданий на контроль целостности, а также проверку контрольных сумм проверяемых объектов при каждой загрузке компьютера. При проверке контрольных сумм файлов и секторов осуществляется сравнение текущих значений контрольных сумм с эталонными (заранее вычисленными) значениями контрольных сумм этих объектов, хранящихся в соответствующих файлах заданий на контроль целостности.
- программа формирования шаблонов для контроля целостности является дополнительным программным обеспечением, входящим в комплект поставки комплекса "Соболь". Она устанавливается на жесткий диск компьютера. Эта программа позволяет определить перечень файлов и физических секторов жестких дисков, подлежащих контролю, и создать шаблоны заданий на контроль целостности, содержащие полный путь к каждому контролируемому файлу и координаты каждого контролируемого сектора.
- задания на контроль целостности - содержат информацию о местоположении контролируемых файлов на жестком диске (полный путь к ним), координаты контролируемых секторов, а также значения контрольных сумм для каждого файла или сектора.

@Страж NT:
Обеспечивает контроль целостности файлов по следующим параметрам:
– наличие файла;
– контрольная сумма данных, содержащихся в файле;
– длина файла;
– дата и время последней модификации.

@Terrier - программа поиска и гарантированного уничтожения информации на дисках.
Порядок действия:
- чтение содержимого сектора, в котором находится найденное ключевое слово (далее – обрабатываемого сектора) в буфер в оперативной памяти;
- изменение буфера: на место стираемого слова записываются символы 0x00;
- запись содержимого буфера в обрабатываемый сектор;
- изменение буфера: на место стираемого слова записываются символы 0xFF;
- запись содержимого буфера в обрабатываемый сектор;
- изменение буфера: на место стираемого слова записываются символы 0x00;
- запись содержимого буфера в обрабатываемый сектор.

##
###

112. Управление доступом. Дискреционный и мандатный методы доступа. Изолированная программная среда.

@Secret Net:
Механизм полномочного разграничения доступа предназначен для:
- разграничения доступа пользователей к конфиденциальным документам;
- контроля потоков конфиденциальной информации в системе;

- контроля вывода конфиденциальной информации на внешние устройства;
- контроля печати конфиденциальных документов.

Механизм полномочного разграничения доступа обеспечивает управление доступом пользователей к конфиденциальной информации, хранящейся в файлах на локальных и подключенных сетевых дисках с файловой системой NTFS и NTFS5. Доступ осуществляется в соответствии с категорией конфиденциальности, присвоенной информации, и уровнем допуска пользователя к конфиденциальной информации. Для каждого пользователя компьютера устанавливается некоторый уровень допуска к конфиденциальной информации. Файлам и каталогам назначается категория конфиденциальности, которая определяется расширенным атрибутом файла или каталога. По умолчанию используются 3 категории конфиденциальности информации: "неконфиденциально" (для общедоступной информации), "конфиденциально", "строго конфиденциально".

Полномочные правила разграничения доступа действуют совместно со стандартными правилами избирательного разграничения доступа в ОС Windows. Поэтому доступ к объекту разрешен только в том случае, если он разрешен и по полномочным, и по избирательным правилам доступа.

Замкнутая программная среда:

Механизм замкнутой программной среды позволяет сформировать для любого пользователя компьютера программную среду, определив индивидуальный перечень программ, разрешенных для запуска. Перечень программ, разрешенных для запуска, может быть задан как индивидуально для каждого пользователя, так и определен на уровне групп пользователей.

На этапе настройки механизма составляется список исполняемых файлов. Список исполняемых файлов может быть сформирован автоматически по информации об установленных на компьютере программах или на основании сведений о запуске программ из журнала Secret Net или журнала безопасности ОС Windows, а также может быть задан вручную. Для файлов, входящих в этот список, можно включить режим контроля целостности. По этой причине механизм замкнутой программной среды и механизм контроля целостности используют единую модель данных.

Для пользователей можно установить один из двух дополнительных режимов работы замкнутой программной среды:

- подсистема замкнутой программной среды не контролирует запуск программ пользователями, наделенными привилегией "Замкнутая программная среда: не действует". По умолчанию этой привилегией наделяются администраторы компьютера;
- для всех пользователей компьютера можно включить "мягкий" режим работы подсистемы замкнутой программной среды. В этом режиме подсистема контролирует все попытки запуска программ пользователем, но разрешает запускать все программы — отсутствующие в списках разрешенных программ и те, целостность которых нарушена. Этот режим обычно используется на этапе настройки механизма.

@Аккорд:

В комплексе "Аккорд-NT" дискреционные правила разграничения доступа устанавливаются присвоением объектам доступа атрибутов доступа. Установленный атрибут означает, что определяемая атрибутом операция может выполняться над данным объектом. Есть специальная программа – редактор прав доступа. Существует также и "черный список". Это файлы, или каталоги, которые присутствуют в списке объектов, для которых не установлен ни один атрибут доступа. Объекты, описанные в "черном списке", становятся недоступными пользователю, даже если они расположены в каталогах, к которым пользователь имеет доступ. В "черный список" можно включать также логические имена устройств и драйверы устройств. Эти объекты после такого описания становятся недоступны пользователю. Таким образом, осуществляется сопоставление

пользователя и доступных ему устройств. Разграничение доступа с использованием мандатного механизма управления доступом комплекса “Аккорд” осуществляется путем присвоения (задания) объектам доступа категории доступа (грифа), которые характеризуются уровнем доступа от 0 (самый низкий) до 15 (максимальный). Установленный для объекта доступа гриф является его меткой конфиденциальности. Пользователям и процессам (опционально) присваиваются категории доступа (уровни допуска), также изменяющиеся от 0 до 15. Доступ пользователя или процесса возможен тогда и только тогда, когда его уровень допуска не ниже грифа объекта доступа.

@Dallas Lock:
Разграничение прав доступа к ресурсам файловой системы реализуется следующими методами:
- мискреционный: обеспечивает доступ к защищаемым объектам (дискам, каталогам, файлам) в соответствии со списками пользователей и групп пользователей, создаваемыми для данных объектов файловой системы. В соответствии с содержимым списка определяются права на доступ к объекту для каждого пользователя (открытие, запись, чтение, удаление, переименование, запуск, копирование);
- мандатный – каждому пользователю присваивается уровень доступа. Пользователь получает доступ к определенному кругу объектов, определяемому этим уровнем.

@Страж NT (дискреционный):
К защищаемым ресурсам компьютера относятся:
– локальные жесткие диски;
– папки;
– файлы;
– порты ввода-вывода;
– принтеры;
– дисководы, CD-ROM, USB-flash диски и другие отчуждаемые устройства.
К стандартным правам относятся следующие:
– синхронизация;
– изменение владельца ресурса;
– изменение списка контроля доступа;
– чтение атрибутов защиты;
– удаление.
Для файлов и папок используется следующий список прав:
– чтение информации;
– запись информации;
– добавление информации;
– чтение атрибутов файла или папки;
– запись атрибутов файла или папки;
– чтение расширенных атрибутов файла или папки;
– запись расширенных атрибутов файла или папки;
– запуск файла для выполнения.
Разрешения доступа пользователей к защищаемым ресурсам:
– нет доступа;
– чтение;
– изменение;
– полный доступ.
@Страж NT (мандатный):
В качестве меток конфиденциальности выступают:
– для защищаемых ресурсов – гриф секретности;
– для пользователей – уровень допуска;

– для прикладных программ – допуск и текущий допуск.
При контроле доступа к защищаемым ресурсам непосредственно сравнению подлежат только значения грифа секретности ресурса и текущего допуска прикладной программы.
Используются следующие значения меток конфиденциальности в порядке повышения:
– несекретно;
– секретно;
- совершенно секретно.

##
###

113. Структура законодательной базы в области разработки средств защиты информации.

Структура:
1) Законы - Федеральный закон «О лицензировании отдельных видов деятельности» (принят 28 апреля 2011 года, изменения от 2018 года)
2) Постановления правительства
- Постановление Правительства Российской Федерации от 2012 года №79 «О лицензировании деятельности по технической защите конфиденциальной информации»
- Постановление Правительства Российской Федерации от 2012 года №171 «О лицензировании деятельности по разработке и производству средств защиты конфиденциальной информации»
3) Нормативно правовые акты ФСТЭК России - СТР-К. Специальные требования и рекомендации по технической защите конфиденциальной информации
4) ГОСТы в области разработки средств защиты информации - ГОСТ Р 50922-2006 Защита информации. Основные термины и определения

Закон «О лицензировании ... » регулирует отношения между органами исполнительной власти РФ, исполнительной власти субъектов РФ, юридических лиц и ИП на:
- деятельность по распространению криптографических средств;
- деятельность по техническому обслуживанию криптографических средств;
- предоставления услуг в области шифрования информации;
- разработку, производство криптографических средств, защищенных с использованием криптографических средств информационных систем, телекоммуникационных систем;
- деятельность по выявлению электронных устройств, предназначенных для негласного получения информации в помещениях и технических средствах (исключение - для собственных нужд юридического лица или ИП);
- деятельность по разработке и (или) производству средств защиты конфиденциальной информации;
- деятельность по технической защите конфиденциальной информации.
Не распространяется на государственную тайну.

Лицензия (Л) – это специальное разрешение на осуществление конкретного вида деятельноти при обязательном соблюдении лицензионных требований и условий, выданное лицензирующим органом юридическому лицу или ИП (Л выдают ФСТЭК России, ФСБ РФ).
Лицензирующие органы осуществляют следующие полномочия: представление Л, переоформление документов (подтверждение наличия Л), приостановление действия Л (за

нарушение лицензионных требований), прекращение действия Л, ведение реестров Л, контроль за соблюдением лицензионных требований, обращение в суд с заявлением об аннулировании Л.
Деятельность, на осуществление которой предоставлена Л, может осуществляться на всей территории Российской Федерации. Можно и в других субъектах РФ, но при условии уведомления лицензиатом лицензирующего органа.
Основанием отказа в предоставлении Л являются:
- наличие в документе, представленных соискателем Л, недостаточной или искаженной информации;
- несоответствие соискателя Л, принадлежащих ему или используемых им объектов лицензирования требованиям или условиям.

Постановление «О лицензировании деятельности по технической защите конфиденциальной информации» определяет порядок лицензирования соответствующей деятельности юридическими лицами и ИП. Лицензирование осуществляет ФСТЭК.
Лицензионные требования и условия:
- наличие специалиста, специальных помещений дли осуществления деятельности;
- наличие измерительного оборудования, автоматической системы обработки информации с аттестатами;
- наличие специальных программ, нормативно-правовой документации.
Срок действия Л - 5 лет и по его окончании может быть продлен по заявлению лицензиата с переоформлением документов, подтверждающих наличие Л.

Постановление «О лицензировании деятельности по разработке и(или) производству средств защиты конфиденциальной информации» определяет порядок лицензирования соответствующей деятельности юридическими лицами и ИП. Лицензирование осуществляет ФСТЭК, но на части особо важных объектов (администрация президента) – ФСБ РФ.
Лицензионные требования и условия (ФСТЭК):
- наличие специалиста, специальных помещений дли осуществления деятельности;
- наличие измерительного оборудования;
- наличие нормативно-правовой документации и выполнение требований конструкторской, программной и технологической документации, единой системы измерений, системы разработки и запуска в производство средств защиты.

Лицензионные требования и условия (ФСБ):
- выполнение нормативных правовых актов;
- выполнение лицензиатом режима конфиденциальности при обращении со сведениями, получаемых по ходу служебной деятельности;
- наличие условий, предотвращающих НСД к средствам защиты конфиденциальной информации;
- соответствие помещений и измерительного оборудования требованиям документации;
- аттестация средств обработки информации, используемых для разработки средств защиты информации;
- выполнение требований конструкторской, программной и технологической документации, единой системы измерений, системы разработки и запуска в производство средств защиты, системы учета изменений в технической документации;
- наличие у руководителя соискателя Л образования по технической защите информации и стажа работы в этой сфере от 5 лет;
- наличие у инженерно-технического персонала соответствующего образования.

Срок действия Л - 5 лет и по его окончании может быть продлен по заявлению лицензиата с переоформлением документов, подтверждающих наличие Л.

##
###

114. Требования, на основании которых разрешается осуществлять лицензионную деятельность в области разработки средств защиты информации.

Все требования указаны в Постановлении Правительства Российской Федерации от 2012 года №171 «О лицензировании деятельности по разработке и производству средств защиты конфиденциальной информации».
Положение определяет порядок лицензирования деятельности по разработке и (или) производству СЗКИ, осуществляемой юридическими лицами и ИП.
Лицензирование деятельности по разработке и (или) производству СЗКИ осуществляет Федеральная служба по техническому и экспортному контролю, а в части разработки и (или) производства СЗКИ, устанавливаемых на объектах Администрации Президента Российской Федерации, и прочие - Федеральная служба безопасности Российской Федерации.

Лицензионными требованиями и условиями при осуществлении деятельности являются (ФСТЭК):
- наличие в штате соискателя лицензии (лицензиата) специалистов, имеющих высшее профессионально образование в области технической защиты информации либо высшее или среднее профессиональное (техническое) образование и прошедших переподготовку или повышение квалификации по вопросам технической защиты информации;
- наличие у соискателя лицензии (лицензиата) помещений для осуществления лицензируемой деятельности;
- выполнение требований конструкторской, программной и технологической документациици, единой системы измерений, системы разработки и запуска в производство СЗКИ;
- соответствие помещений, производственного, испытательного и контрольно-измерительного оборудования техническим нормам и требованиям по технической защите информации;
- наличие нормативных правовых актов, нормативно-методических и методических документов по вопросам разработки средств защиты информации в соответствии с перечнем, установленным Федеральной службой по техническому и экспортному контролю.

Лицензионными требованиями и условиями при осуществлении деятельности являются (ФСБ):
- выполнение нормативных правовых актов, относящихся к лицензируемой деятельности;
- выполнение лицензиатом режима конфиденциальности при обращении со сведениями в ходе служебной деятельности;
- наличие условий, предотвращающих НСД к СЗКИ;
- соответствие помещений требованиям технической и технологической документации на оборудование деятельности;

- соответствие производственного, технологического, испытательного и контрольно-измерительного оборудования требованиям
- аттестация средств обработки информации, используемых для разработки средств защиты конфиденциальной информации, в соответствии с требованиями по защите информации с использованием лицензионных баз данных и программного обеспечения для электронно-вычислительных машин;
- выполнение требований конструкторской, программной и технологической документации, единой системы измерений, системы разработки и запуска в производство средств защиты конфиденциальной информации;
- наличие системы учета изменений, внесенных в техническую и конструкторскую документацию на производимую продукцию, и системы учета готовой продукции;
- наличие у руководителя соискателя лицензии (лицензиата) и (или) уполномоченного им лица документа о высшем профессиональном образовании в области технической защиты информации либо документов о высшем или среднем профессиональном (техническом) образовании и о переподготовке или повышении квалификации по вопросам технической защиты информации, а также производственного стажа в области лицензируемой деятельности не менее 5 лет;
- наличие у инженерно-технического персонала, осуществляющего работы в области лицензируемой деятельности, документа о высшем образовании или профессиональной подготовке со специализацией, соответствующей выполняемым работам.

Срок действия лицензии составляет 5 лет и по его окончании может быть продлен по заявлению лицензиата в порядке, предусмотренном для переоформления документа, подтверждающего наличие лицензии, с приложением документов, предусмотренных пунктом 6 настоящего Положения.
Переоформление документа, подтверждающего наличие лицензии, приостановление, возобновление ее действия, аннулирование лицензии, а также ведение реестра лицензий и предоставление сведений, содержащихся в реестре лицензий, осуществляются в порядке, установленном Федеральным законом "О лицензировании отдельных видов деятельности".

##
###

115. Понятие эффективного коммуникативного процесса. Безопасность организационных коммуникаций.

---> docs/psychology/Бячкова Н. Б. - Основы безопасности управленческой деятельности.pdf

##
###

116. Мотивация работника в структуре политики безопасности предприятия.

---> docs/psychology/Бячкова Н. Б. - Основы безопасности управленческой деятельности.pdf

##
###

117. Роль организационной культуры в создании эффективной системы безопасности предприятия.

---> docs/psychology/Бячкова Н. Б. - Основы безопасности управленческой деятельности.pdf

##
###

118. Способы и приемы безопасной кадровой политики на предприятии.

---> docs/psychology/Бячкова Н. Б. - Основы безопасности управленческой деятельности.pdf

##
###

119. Методы и средства защиты инфраструктуры маршрутизации отказоустойчивых компьютерных сетей

Маршрутизация является одной из критически важных задач, обеспечивающей корректное функционирование,
доступность, надёжность и отказоустойчивость компьютерной сети. Выделяют следующие угрозы нарушения безопасности маршрутизации:
* угрозы, направленные на сеансы обмена маршрутной информацией: сброс TCP-сессий, исчерпание ресурсов;
* угрозы, направленные на роутеры: отказ в обслуживании, подбор паролей, переполнение буфера, повышение привилегий;
* угрозы, направленные на маршрутную информацию: внедрение ложных маршрутов, создание петель, удаление корректных маршрутов, чтение маршрутной информации, раскрытие параметров маршрутизации.

Для защиты инфраструктуры маршрутизации используются следующие основные методы:
* управление распространением маршрутной информации с применением фильтров маршрутизации,
управление обменом маршрутной информацией между узлами и процессами маршрутизации;
* ограничение множества систем, использующих протоколы маршрутизации, использование методов аутентификации,
ограничение сеансов маршрутизации только доверенными узлами;
* регистрация событий маршрутизации, регистрация изменения состояний сеансов маршрутизации со смежными или соседними узлами.

##
###

120. Методы и средства защиты информации в локальных вычислительных сетях от атак канального уровня

Для описания методов и средств защиты информации рассмотрим наиболее распространенные атаки канального уровня:

1) ARP-spoofing (ARP-poisoning)

Address Resolution Protocol (ARP) – протокол канального уровня, использующийся для установления соответствия между IP-адресом и MAC-адресом машины. Для определения MAC-адреса получателя по IP-адресу хост формирует широковещательный Ethernet-кадр, содержащий ARP-запрос (ARP-Request). Запрос содержит MAC и IP отправителя и IP получателя. Хост, обнаруживший свой IP в поле "сетевой адрес получателя", дописывает свой MAC-адрес и отправляет ARP-ответ (ARP-Reply). Получив искомый MAC-адрес, хост заносит его в ARP-кэш, и в дальнейшем для отправки запросов пользуется полученным адресом.

Недостатком данного протокола является отсутствие проверки подлинности пакетов: как запросов, так и ответов. Злоумышленник, отправляя ARP-ответы без предварительного ARP-запроса, может подменить содержимое ARP-кэша произвольным образом и перехватывать трафик между узлами в пределах одного широковещательного домена.

Способы защиты от атаки ARP-spoofing:

- использовать статическую ARP-таблицу. Необходимые соответствия адресов добавляются в ARP-таблицу и в дальнейшем не заменяются;
- использовать VLAN. В случае, когда машины злоумышленника и жертвы будут расположены в разных виртуальных сетях, атака не будет возможна;
- использовать Packet Filtering ACL на коммутаторах. Например, современные коммутаторы DLINK и CISCO поддерживают инструменты анализа пакетов и фильтрацию по конкретным параметрам. Первый вариант: фильтрация всех ARP-пакетов на всех пользовательских портах, у которых в Sender Protocol Address содержится IP-адрес шлюза, что позволяет защититься от подмены адреса шлюза. Второй вариант: фильтрация всех ARP-пакетов на каждом из пользовательских портов, у которых Sender Hardware Address и Sender Protocol Address отличаются уже известных MAC и IP-адресов, что позволяет защититься еще и от подмены адреса некоторого пользователя.

2) Атаки MAC-spoofing, MAC-flooding

MAC-spoofing – атака канального уровня, суть которой заключается в изменении MAC-адреса сетевого устройства. Благодаря этому коммутатор начинает отправлять на порт, к которому подключен злоумышленник, пакеты, которые он до этого видеть не мог.

MAC-flooding (переполнение таблицы коммутации) – атака, основанная на том, что таблица коммутации в коммутаторах имеет ограниченный размер. После заполнения таблицы, коммутатор не может более запоминать новые MAC-адреса и начинает отправлять трафик на все порты.

Для защиты от этих атак можно использовать функцию коммутатора Port Security: она позволяет указать список MAC-адресов, которым разрешено передавать данные через порт, и дополнительно позволяет ограничить количество подключений на интерфейсе.

3) Атаки на DHCP-сервер

К основным атакам этого класса относятся:

- DoS DHCP-сервера. Злоумышленник может сформировать и послать DHCP-серверу огромное количество DHCP-запросов с разными MAC-адресами. Сервер будет выделять IP-адреса из пула, который через некоторое время закончится, после чего DHCP-сервер не сможет обслуживать новых клиентов. Для защиты от этой атаки используется метод DHCP Snooping, который заключается в сравнении MAC-адреса, указанного в DHCP-

запросе, с MAC-адресом, который прописан на порту коммутатора. Если адреса не совпадают – пакет отбрасывается, иначе принимается;

- «Ложный» DHCP-сервер. Злоумышленник может развернуть свой DHCP-сервер и выдавать свои настройки пользователям сети, обеспечивая себе возможность прослушивания трафика, подделки DNS-ответов и т.д.. Для этого необходимо предварительно вывести из строя легальный DHCP-сервер (например, с помощью DoS), следовательно, защита от атаки будет происходить аналогичным образом.

4) Атака VLAN hopping

VLAN – виртуальная сеть, хосты в которой взаимодействуют друг с другом так, как если бы они были подключены к одному широковещательному домену, независимо от их физического местоположения. Порты коммутаторов, принадлежащие одной VLAN, могут обмениваться кадрами между собой, но не могут обмениваться кадрами с портами других VLAN. При этом порты, предназначенные для передачи кадров только одной виртуальной сети, называются портами доступа, а порты, предназначенные для передачи кадров нескольких VLAN – магистральными, или «транковыми».

VLAN hopping – общее название для атак, которые предполагают проникновение в VLAN, который до выполнения атаки не был доступен атакующему.

Основной атакой этого класса является атака с использованием Dynamic Trunking Protocol (DTP), когда злоумышленник через свой порт отправляет пакет DTP, в результате чего коммутатор считает этот порт магистральным.

Для создания ЛВС, защищенных от атак этого класса, используются следующие принципы:

- запретить передачу кадров собственной VLAN по магистральным каналам, а в качестве native VLAN использовать VLAN, специально выделенную для этих целей;
- не использовать стандартную VLAN 1, особенно для управления сетевым оборудованием;
- на магистральных портах использовать только необходимые VLAN – VLAN, все прочие запрещать;
- не использовать одинаковые VLAN на разных коммутаторах;
- все неиспользуемые порты коммутатора переводить в режим shutdown и определять их в отдельную изолированную VLAN.

5) Атаки на STP

Протокол Spanning Tree Protocol (STP) предназначен для предотвращения зацикливания пакетов сети при наличии дублирующих маршрутов. Для этого сначала производится обнаружение коммутаторов, которые связаны между собой. Далее среди них выбирается главный, корневой коммутатор (root bridge), после чего блокируются порты коммутатора, которые создают петли в получившейся топологии.

Для построения древовидной структуры без петель в сети должен быть определен корневой коммутатор, от которого и будет строиться это дерево. В качестве корневого коммутатора выбирается коммутатор с наименьшим значением идентификатора. Идентификатор – это число длиной 8 байт, 6 младших байтов которого составляет MAC-адрес его блока управления, а 2 старших байта конфигурируются вручную, что позволяет администратору сети влиять на процесс выбора корневого коммутатора. Если администратор не вмешается в данный процесс, в качестве корневого будет выбран коммутатор с наименьшим MAC-адресом блока управления. Такой выбор может быть далеко не рациональным, поэтому рекомендуется выбирать корневой коммутатор исходя из топологии сети и назначать ему наименьший идентификатор вручную. Далее для каждого коммутатора определяется корневой порт (root port) – порт, который имеет кратчайшее расстояние до корневого коммутатора. Для каждого логического сегмента сети выбирается назначенный мост (designated bridge), один из портов которого будет принимать пакеты от сегмента и передавать их в направлении корневого коммутатора через корневой порт данного моста.

В процессе атаки злоумышленник может притвориться коммутатором так же, как и в атаке VLAN hopping, и направить в сторону атакуемого коммутатора BDPU-пакет с подделанным приоритетом и MAC-адресом, чтобы в результате самому стать корневым коммутатором и с его помощью перехватывать сетевой трафик.
Для защиты от атак этого класса используются следующие принципы:

- использовать протоколы семейства STP с целью построения отказоустойчивых ЛВС только при необходимости. По возможности использовать механизмы и протоколы маршрутизации сетевого уровня;
- административно определять и назначать корневые коммутаторы. Использовать дополнительные механизмы и средства защиты протокола STP (RootGuard, LoopGuard, UplinkFast, UDLD) для предотвращения получения роли корневого коммутатора другими коммутаторами;
- на портах доступа коммутаторов ЛВС выполнять настройки по предотвращению возможности появления или фильтрации BPDU-пакетов протокола STP (механизмы BPDU Guard и BPDU Filter соответственно), а также выполнять настройки для быстрого включения и защиты корневого коммутатора (механизмы PortFast и RootGuard соответственно).