**Вопросы к экзамену по ПМ 04 «Сопровождение и обслуживание программного обеспечения компьютерных систем»**

1. Опишите основные процессы и взаимосвязь между документами в информационной системе согласно стандартам. ГОСТ Р ИСО/МЭК 12207.

Цель теории информационных процессов и систем состоит в том, чтобы представить имеющиеся знания в едином комплексе понятий, определений и положений, основываясь на сущности и закономерностях проектирования, внедрения и сопровождения информационных систем.

Системный подход оперирует рядом категориальных понятий. Фундаментальным понятием системного подхода является понятие системы. Согласно определению система есть сущность, которая в результате взаимодействия ее частей (компонентов) может поддерживать свое существование и функционировать как единое целое. Системы функционируют как целое, что порождает у них свойства, отличающиеся от свойств составляющих ее частей. Эти свойства известны как эмерджентные (возникающие).

Принято выделять простые и сложные системы. Сложность может проявляться двумя различными путями: статическим и динамическим. Статическая сложность (детальная) определяется в процессе детализации как количество рассматриваемых элементов. Динамическая сложность зависит от отношений между элементами. Информационные системы относятся к классу динамически сложных. Таким образом, информационная система состоит из множества элементов или подсистем, которые находятся в разных состояниях и могут изменяться, в зависимости от изменения других частей.

По виду элементов информационная система относится к системам типа «процесс» (элементами являются информационные процессы).

Отметим и другие важные определения.

Структура – совокупность элементов и их связей.

Цель – состояние, к которому стремится система.

Среда метасистема, в которой рассматриваемая система является составной частью.

Функционирование системы – работа системы в рамках заданной структуры.

Развитие системы работа системы в условии острых противоречий, которые могут вызвать изменение структуры.

Управление – целенаправленный перевод системы из одного состояния в другое желаемое.

Если целью является познание уже существующей системы, то вполне пригодным оказывается дескриптивное определение системы.

Система – совокупность объектов, свойства которой определяются отношением между этими объектами.

Объекты называют подсистемами или элементами системы. Каждый объект при самостоятельном исследовании может рассматриваться как система.

Любая техническая система создается под заранее известную цель, которая обычно является субъективной, поскольку она предлагается разработчиком, но эта цель должна исходить из объективных потребностей общества. Таким образом, можно считать, что цель формируется в процессе взаимодействия между явлениями окружающей действительности. При этом возникает ситуация, которая заставляет строить новую систему. Ситуация может стать проблемной, если она не разрешается имеющимися средствами. Могут создаваться новые недостающие средства, и в этом смысле ярким примером являются информационные технологии.

В настоящее время уже давно сформировались идеология и практика применения различных средств извлечения, передачи, хранения, обработки и представления информации. Однако разрозненное их применение или использование в ограниченной совокупности не позволяло до сих пор получить значительный системный эффект. Необходим подход к информационным технологиям как к системе. Такой подход является обоснованным ввиду того, что информационная технология обладает единой целью, а именно: необходимостью формирования информационного ресурса в обществе, имеет сопрягаемые взаимодействующие средства ее реализации, характеризуется тенденцией развития в связи с интенсивным обновлением средств вычислительной техники и техники связи. Анализ информационных технологий как системы должен выполняться на основе дескриптивного определения, разработка информационных технологий должна базироваться на конструктивном подходе. Такой подход предполагает необходимость возникновения проблемной ситуации для разработки системы. Можно считать, что возникающая проблема порождает будущую систему. Прежде всего, разработчик должен определить границы системы, полагая, что цель ее функционирования известна. Необходимо в состав системы включить те элементы, которые своим функционированием обеспечат реализацию заданной цели. Значит, конструктивное определение системы состоит в следующем: система – конечное множество функциональных элементов и отношений между ними, которые выделяются из окружающей среды в соответствии с поставленной целью в рамках определенного временного интервала ее реализации. Все то, что не вошло в состав системы, относят к окружающей среде. Очевидно, что окружающая среда включает в себя другие системы, которые реализуют свои цели функционирования. Входы и выходы системы связаны с внешней средой. На модельном уровне выделяют модель системы, модель внешней среды на входе системы, модель внешней среды на выходе системы и модели связей между системой и внешней средой на входе и выходе. Внешней средой для информационной технологии могут выступать производство, научное исследование, проектирование, обучение и т. д. Связи между информационной технологией и внешней средой носят чисто информационный характер. В процессе взаимодействия с внешней средой реализуются основные функции информационной технологии. Функции как проявление свойств системы во времени тесно связаны с ее структурой. Дескриптивный подход реализуется путем изучения функции или структуры системы. В соответствии с этим в теории систем получили применение функциональный и структурный подходы.

Учитывая, что структура отображает связи между элементами системы с учетом их взаимодействия в пространстве и во времени, можно утверждать, что структурный подход есть развитие дескриптивного подхода. Он служит для изучения (познавания) какой-то существующей системы. Функциональный подход отображает функции системы, реализуемые в соответствии с поставленной перед ней целью. Поэтому функциональный подход есть развитие конструктивного. Функции системы должны быть заданы при ее построении и должны реализовываться при функционировании системы.

Структура системы описывается на концептуальном, логическом и физическом уровнях:

– концептуальный уровень позволяет качественно определить основные подсистемы, элементы и связи между ними;

– на логическом уровне могут быть сформированы модели, описывающие структуру отдельных подсистем и взаимодействия между ними;

– физический уровень означает реализацию структуры на известных программно-аппаратных средствах.

Так как техническая система создается искусственно, то цель ее функционирования заранее субъективно известна. Можно считать, что этой цели соответствуют определенный перечень функций и некоторая оптимальная структура системы, называемая формальной структурой. Под ней понимают совокупность функциональных элементов и отношений между ними, необходимых и достаточных для достижения системой заданной цели. Формальная структура есть некоторая идеальная, не имеющая физического наполнения структура. Она реализуется различными средствами, поэтому формальной структуре может соответствовать ряд материальных физических структур. Внешняя среда, взаимодействуя с информационной технологией как с системой, может выступать как метасистема, ставя перед ней определенные задачи и формулируя цели. Внедрение информационных технологий в жизнь общества за конечный временной интервал даст эффект, если будут типизированы системы, в которые внедрены информационные технологии, и определены типовые структуры информационной технологии. Для каждой системы существует свое территориальное распределение пользователей и средств информационной технологии. Разным может быть и комплекс решаемых задач. Характер и временной интервал реализации целей информационных технологий также зависят от того, в какой области технология используется: в промышленности, научных исследованиях, проектировании, обучении и т. д. Возникает задача создания широкого набора конкретных информационных технологий, настроенных на параметры реальных систем. Таким образом, информационные технологии становятся массовым объектом разработки.

При использовании информационных технологий в системном аспекте необходимо соблюдать следующие принципы.

1. Наличие сформулированной единой цели у информационных технологий в рамках разрабатываемой системы.

2. Согласование информационных технологий по входам и выходам с окружающей средой.

3. Типизация структур информационных технологий.

4. Стандартизация и взаимная увязка средств информационной технологии.

5. Открытость информационных технологий как системы.

Информационный процесс является разновидностью технологического процесса, содержащего действия (физические, механические и др.) по изменению состояния информации.

На основе информационных технологий решается задача автоматизации информационных процессов.

Теоретической базой для информационных технологий является информатика. Цель информатики – изучение структуры и общих свойств научной информации с выявлением закономерностей процессов коммуникации.

Термин «технология» имеет множество толкований. В широком смысле под технологией понимают науку о законах производства материальных благ. Это понятие объединяет три составляющих: информационную, инструментальную и социальную. Для конкретного производства технологию понимают в узком смысле как совокупность приемов и методов, определяющих последовательность действий для реализации производственного процесса.

Методология любой технологии включает в себя: декомпозицию производственного процесса на отдельные взаимосвязанные и подчиненные составляющие (стадии, этапы, фазы, операции); реализацию определенной последовательности выполнения операций, фаз, этапов и стадий производственного процесса в соответствии с целью технологии; технологическую документацию, формализующую выполнение всех составляющих.

В развитии технологии выделяют два принципиально разных подхода.

Первый подход характеризуется непрерывным совершенствованием установившейся базисной технологии и достижением верхнего предельного уровня, когда дальнейшее улучшение является неоправданным из-за больших экономических вложений.

Второй подход отличается отказом от существующей технологии и переходом к принципиально другой, развивающейся по законам первого подхода.

**1.2. Информационная технология – составная часть информатики и основа проектирования информационных систем**

Информационная технология – совокупность методов и способов получения, обработки, представления информации, направленных на изменение ее состояния, свойств, формы, содержания и осуществляемых в интересах пользователей.

Можно выделить три уровня рассмотрения информационных технологий:

– первый уровень – теоретический. Основная задача создание комплекса взаимосвязанных моделей информационных процессов, совместимых параметрически и критериально;

– второй уровень – исследовательский. Основная задача – разработка методов, позволяющих автоматизированно конструировать оптимальные конкретные информационные технологии;

– третий уровень прикладной, который целесообразно разделить на две страты: инструментальную и предметную.

Инструментальная страта (аналог – оборудование, станки, инструмент) определяет пути и средства реализации информационных технологий, которые можно разделить на методические, информационные, математические, алгоритмические, технические и программные.

Предметная страта связана со спецификой конкретной предметной области и находит отражение в специализированных информационных технологиях, например, организационное управление, управление технологическими процессами, автоматизированное проектирование, обучение и др.

Успешное внедрение информационных технологий связано с возможностью их типизации. Конкретная информационная технология обладает комплексным составом компонентов, поэтому целесообразно определить структуру и состав информационной технологии.

Конкретная информационная технология определяется в результате компиляции и синтеза базовых технологических операций, специализированных технологий и средств реализации.

**1.3. Информационные процессы – основа функционирования информационных систем**

Функционирование информационной системы базируется на реализации информационных процессов, разнообразие которых требует выделения базовых, позволяющих осуществлять типизацию и унификацию проектных решений.

Технологический процесс – часть информационного процесса, включающая действия (физические, механические и др.) по изменению состояния информации.

Информационная технология базируется на реализации информационных процессов, разнообразие которых требует выделения базовых процессов, характерных для любой информационной технологии.

Базовый информационный процесс основан на использовании стандартных моделей и инструментальных средств и может быть использован в качестве составной части информационной технологии. К их числу можно отнести: операции извлечения, транспортировки, хранения, обработки и представления информации.

В состав базовых информационных процессов входят: извлечение информации; транспортирование информации; обработка информации; хранение информации; представление и использование информации.

Базовые информационные технологии строятся на основе базовых информационных процессов, но кроме этого включают ряд специфических моделей и инструментальных средств. Этот вид технологий ориентирован на решение определенного класса задач и используется в конкретных технологиях в виде отдельной компоненты. Среди них можно выделить технологии: мультимедиа; геоинформационные; защиты информации; CASE; телекоммуникационные; искусственного интеллекта

Специфика конкретной предметной области находит отражение в специализированных информационных технологиях, например, организационное управление, управление технологическими процессами, автоматизированное проектирование, обучение и др. Среди них наиболее продвинутыми являются информационные технологии: организационного управления (корпоративные информационные технологии); в промышленности и экономике; образовании; автоматизированного проектирования.

Аналогом инструментальной базы (оборудование, станки, инструмент) являются средства реализации информационных технологий, которые можно разделить на методические, информационные, математические, алгоритмические, технические и программные.

Методические средства определяют требования при разработке, внедрении и эксплуатации информационных технологий, обеспечивая информационную, программную и техническую совместимость. Наиболее важными из них являются требования по стандартизации.

Информационные средства обеспечивают эффективное представление предметной области, к их числу относятся информационные модели, системы классификации и кодирования информации (общероссийские, отраслевые) и др.

Математические средства включают в себя модели решения функциональных задач и модели организации информационных процессов, обеспечивающие эффективное принятие решения. Математические средства автоматически переходят в алгоритмические, обеспечивающие их реализацию.

Технические и программные средства задают уровень реализации информационных технологий, как при их создании, так и при их реализации.

Таким образом, конкретная информационная технология определяется в результате компиляции и синтеза базовых технологических операций, «отраслевых технологий» и средств реализации.

Быстрое развитие информационных технологий позволяет переместить всю информацию в киберпространство. Основными задачами, которые необходимо решать в этом случае, являются следующие:

− определение моделей данных для новых типов (например, пространственных, темпоральных, графических) и их интеграция с традиционными системами баз данных;

− масштабирование баз данных по размеру (до петабайт), пространственному размещению (распределенные) и многообразию (неоднородные);

− автоматическое обнаружение тенденций данных, структур и аномалий (добывание данных, анализ данных);

− интеграция (комбинирование) данных из нескольких источников;

− создание сценариев и управление потоком работ (процессом) и данными в организациях;

− автоматизация проектирования и администрирования базами данных.

1. Опишите основные методы внедрения и анализа функционирования программного обеспечения: Виды внедрения, план внедрения, стратегии, цели и сценарии внедрения.

Первым, самым важным шагом, предшествующим внедрению новой стра­тегии, должен быть анализ системы менеджмента компании, ее резервов, дефектов и их устранение до начала процесса внедрения стратегии. Для это­го следует провести управленческую диагностику фирмы, определить силь­ные и слабые стороны системы управления.

Процесс внедрения стратегии организации включает ряд этапов: **Подготовительный этап:**

• Анализ существующей системы менеджмента компании, определе­ние ее соответствия внедряемой стратегии. Разработка мер по устранению выявленных дефектов.

• Решение проблемы мотивации менеджеров и специалистов, внедряю­щих стратегию.

• Разработка системы управления внедрением стратегии.

• Разработка системы стратегического контроля.

• Подбор управленческого персонала и принятие решений о расстанов­ке менеджеров, связанных с внедрением стратегии.

• Формирование рабочих групп и команд по решению проблем внедре­ния стратегии.

• Изменение организационной структуры и приведение ее в соответ­ствие с новой стратегией.

• Подготовка персонала к внедрению стратегии, определение необ­ходимых изменений в организационной культуре, прежде всего, обуче­ние и повышение квалификации персонала для работы в изменившихся условиях, развитие обучения и самообучения, трансфера (распростра­нения и передачи знаний), применение новых систем мотивации и сти­мулирования.

• Разработка плана внедрения (этапы внедрения, результаты) и форми­рование временных графиков внедрения стратегии, распределение функций власти и ответственности между конкретными менеджерами, участвующи­ми во внедрении стратегий.

**Внедрение стратегии:**

• Проведение рабочих совещаний групп и команд по внедрению стра­тегии.

• Установление прямых коммуникаций и взаимодействия между работника­ми, внедряющими стратегию, в том числе по каналам телекоммуникаций.

• Финансирование внедрения стратегии, рабочих групп и команд.

• Контроль результатов внедрения стратегии.

• Определение экономической эффективности и результатов стратегии. Сравнение их с планом.

• Сравнение достигнутых результатов с результатами главных конку­рентов.

• Проведение совещаний руководства компании и руководителей рабо­чих групп и команд.

• Разработка стратегических установок по внедрению стратегии и рас­пространение ее в компании.

• Анализ текущих результатов внедрения стратегии и их сопоставление с планами, определение причин отклонений и принятие решений о про­должении работ, коррекции или прекращении.

**Этап завершения реализации основных этапов стратегии:**

• Оценка общих результатов внедрения стратегии в плановых периодах (месяцы, кварталы, годы).

• Определение причин отклонений и принятие корректирующих ре­шений.

• Анализ перспектив развития стратегических процессов.

Для быстрого и эффективного внедрения стратегии необходимы:

— развитие стратегического мышления менеджеров организации;

— разработка и применение систем управления внедрением стратегии стратегического контроля;

— создание прогрессивной, инновационной культуры организации;

— расширение и интеграция функций менеджеров в единый комплекс;

— реорганизация структуры управления на основе применения сетевых и проектных структур.

Управление в процессе реализации стратегии проблем значительно уп­рощается, если используются механизмы политик, процедур и правил.

Политика определяет общие ориентиры действий. Процедуры указыва­ют обязательную последовательность действий при возникновении проблем. Правила определяют обязательные действия менеджеров в условиях стан­дартных и часто возникающих ситуаций.

Среди процессных, непрерывно повторяющихся действий менеджеров, следует выделить виды деятельности, существенно влияющие на внедрение стратегии:

— применение систем мотивации и стимулирования успешной разра­ботки и реализации стратегии;

— планирование и прогнозирование результатов стратегии на основе количественно измеримых показателей;

— обеспечение научной организации труда менеджеров и рабочих групп; —переход к экономичной и гибкой структуре управления фирмой, име­ющей минимум уровней управления;

— координация и контроль стратегических задач, закрепление за ними не подразделений, а конкретных менеджеров;

— организация внедрения стратегии на основе сети рабочих групп и команд;

— стимулирование инноваций;

— рост квалификации и мотивация развития личностей работников, непрерывное обучение и самообучение;

— развитие организационной культуры и создание благоприятных усло­вий труда в организации.

Эффективность внедрения стратегии сильно зависит от управленческо­го контроля, который должен действовать на всех стадиях процесса внедре­ния стратегии и охватывать все звенья организации.

Применяются следующие виды контроля:

административный контроль выполнения управленческих решений и регламентов, соблюдения правовых норм и законодательства, расстановки кадров, выполнения планов и заданий, взаимодействия организации с ок­ружающей средой (экологический контроль, соблюдения прав потребите­лей, этическое поведение по отношению к конкурентам);

финансовый контроль—учет и анализ расходования финансовых ресурсов, обеспечение финансовых потребностей организации. В современных систе­мах управления нередко принципы финансового аудита распространяются на сферу маркетинга, контроля качества, процессы производства;

бюджетный контроль, включающий разработку системы бюджетов под­разделений организации и их объединение в главный бюджет организации. Главное назначение этого контроля — связь финансовых ресурсов с результатами производства. В российской практике бюджетный менеджмент час­то вообще отсутствует;

контроль эффективности систем стимулирования и мотивации, оцени­вающий степень заинтересованности работников и менеджеров в решении задач организации;

маркетинговый контроль, обеспечивающий руководство информацией об изменениях спроса на рынке, предпочтений клиентуры, реакции рынка на поведение организации;

контроль качества, включая оценку уровня качества, соблюдение стан­дартов качества, причин отклонений от них.

Для подготовки системы менеджмента к внедрению стратегии нужно обеспечить:

— расширение функций менеджеров с учетом новых видов функций — интегральных (объединяющих ряд базовых функций) и объектно-ориенти­рованных (специализированных функций, направленных на конкретный объект, например, маркетинговый менеджмент);

— обеспечение взаимодействия всех основных функций в процессе уп­равления и интеграция всех функций управления в единый процесс;

— перенос центра тяжести в принятии и реализации управленческих решений от формальных административных структур к сети команд менед­жеров и специалистов и изменение приоритетов.

Профессиональные функции современного менеджера целесообразно разделить на следующие виды [37]:

— базовые функции общего менеджмента, универсальные для всех менед­жеров и всех видов деятельности, независимо от подразделения, в котором менеджеры работают, образующие каркас содержания труда каждого менед­жера. К ним относятся планирование, организация, контроль, координа­ции и активизация персонала. Эти функции дополняются функциями, вхо­дящими в две другие группы;

— интегральные функции менеджмента. Эти функции образуют об­щую платформу для всех других функций, тесно связаны со всеми други­ми функциями и друг с другом (рис. 7.**1).**Их особенность в обобщенном характере и влиянии на все функции и все виды действий менеджеров. Они включают функции дебюрократизации, руководства на основе ли­дерства, непрерывные инновации, развитие самоуправления, новой куль­туры менеджмента, непрерывное обучение и формирование интеллекту­ального потенциала. На основе этих функций менеджерами производится разработка концепций управления своими подразделениями и работни­ками, развитие нестандартного мышления и быстрой реакции на изме­нения. Создание позитивной культуры менеджмента основывается на развитии взаимного доверия, уважения, открытости, честности, добро­желательности и справедливости;

— специальные (объектно-ориентированные) функции менеджмента. Эти функции как самостоятельные возникли сравнительно недавно в связи с углублением и расширением специализации управленческого труда. Рост конкуренции, появление новых видов деятельности, рост сложности техно­логий и создаваемых продуктов, обладающих различными потребительны­ми качествами, вызвали необходимость перехода от общего руководства спе­циализированными процессами к ориентации на конкретные объекты управления, например на маркетинг, новые продукты, инвестиции. Так, место системы маркетинга все больше занимает маркетинговый менедж­мент, содержащий систему маркетинга, средства и методы управления этой системой.

Чем выше конкурентоспособность компании, тем полнее набор функ­ций менеджмента. Во многих современных организациях они реализуются частично. Их развитие сдерживается внутренней бюрократией, в условиях которой отсутствуют стимулы эффективной работы, практикуется конфор­мизм или соглашательство с любыми указаниями руководства, безразличие и беспринципность, отказ от инициативных инноваций, стремление избе­жать риска.

1. Опишите типовые функции инструментария для автоматизации процесса внедрения информационной системы.

**Информационные системы**

Информационная система есть совокупность технического, программного и организационного обеспечения, а также персонала, предназначенная для того, чтобы своевременно обеспечивать надлежащих людей надлежащей информацией

Программное обеспечение за полвека своего существования претерпело огромные изменения, пройдя путь от программ, способных выполнять только простейшие логические и арифметические операции, до сложных систем управления предприятиями. В программном обеспечении всегда можно было выделить два основных направления развития:

• выполнение вычислений;

• накопление и обработка информации.

Интуиции, личного опыта руководителя и размеров капитала уже мало для того, чтобы быть первым. Для принятия любого грамотного управленческого решения в условиях неопределенности и риска необходимо постоянно держать под контролем различные аспекты финансово-хозяйственной деятельности, будь то торговля, производство или предоставление каких-либо услуг.

В условиях жесткой конкуренции, динамического рынка даже самые консервативные или небогатые предприятия не могут позволить себе отказаться от такого мощного средства, как автоматизация. Выгода от использования современных компьютерных технологий в промышленности столь велика, что эпоха агитации за автоматизацию давно прошла.

В настоящее время понятие информационной системы настолько размыто, что под информационной системой может быть определено любое понятие от компьютерной программы, помогающей автоматизировать какой-то процесс, до сложившегося набора правил и процедур, регламентирующих действия сотрудников компании по организации процессов создания и использования информации в нужном для компании виде. Современный бизнес крайне чувствителен к ошибкам в управлении, и для принятия грамотного управленческого решения в условиях неопределенности и риска необходимо постоянно держать под контролем различные аспекты финансово-хозяйственной деятельности предприятия (независимо от профиля его деятельности). Теория управления предприятием представляет собой довольно обширный предмет для изучения и совершенствования.

**• Управление производственным процессом**

Оптимальное управление производственным процессом представляет собой очень трудоемкую задачу. Основным механизмом здесь является планирование. Автоматизированное решение подобной задачи дает возможность грамотно планировать, учитывать затраты, проводить техническую подготовку производства, оперативно управлять процессом выпуска продукции в соответствии с производственной программой и технологией. Очевидно, что чем крупнее производство, тем большее число процессов участвует в создании прибыли, а значит, использование информационных систем жизненно необходимо.

**•Документооборот**

Документооборот является очень важным процессом деятельности любого предприятия. Хорошо отлаженная система учетного документооборота отражает реально происходящую на предприятии текущую производственную деятельность и дает управленцам возможность воздействовать на нее. Поэтому автоматизация документооборота позволяет повысить эффективность управления.

**•Оперативное управление предприятием**

Информационная система, решающая задачи оперативного управления предприятием, строится на основе базы данных, в которой фиксируется вся возможная информация о предприятии. Такая информационная система является инструментом для управления бизнесом и обычно называется корпоративной информационной системой. Информационная система оперативного управления включает в себя массу программных решений по автоматизации бизнес-процессов, имеющих место на конкретном предприятии.

**Назначение информационных систем.**

«Идеальная» информационная система управления предприятием должна автоматизировать все или, по крайней мере, большинство из видов деятельности предприятия. При чем, автоматизация должна быть выполнена не ради автоматизации, а с учетом затрат на нее, и дать реальный эффект в результатах финансово-хозяйственной деятельности предприятия.

В зависимости от предметной области информационные системы могут весьма значительно различаться по своим функциям, архитектуре, реализации. Однако можно выделить ряд свойств, которые являются общими.

• Информационные системы предназначены для сбора, хранения и обработки информации, поэтому в основе любой из них лежит среда хранения и доступа к данным.

• Информационные системы ориентированы на конечного пользователя, не обладающего высокой квалификацией в области вычислительной техники. Поэтому клиентские приложения информационной системы должны обладать простым, удобным, легко осваиваемым интерфейсом, который предоставляет конечному пользователю все необходимые для работы функции и в то же время не дает ему возможность выполнять какие-либо лишние действия.

На предприятии должна быть создана база данных, которая обеспечивает хранение информации и доступность ее для всех составляющих системы управления.

Наличие такой базы данных позволяет сформировать информацию для принятия решений. Сама по себе информационная система не является инструментом для принятия управленческих решений. Решения принимаются людьми. Но система управления в состоянии представить или «подготовить» информацию в таком виде, чтобы обеспечить принятие решения. Системы поддержки принятия решений в состоянии обеспечить, например:

• отслеживание эффективности работы различных участков и служб для выявления и устранения слабых звеньев, а также для совершенствования бизнес-процессов и организационных единиц (т.е. анализ информации может привести к изменению правил выполнения тех или иных управленческих процессов и даже к изменению организационной структуры предприятия);

• анализ деятельности отдельных подразделений;

• обобщение данных из различных подразделений;

• анализ показателей различных направлений финансово-хозяйственной деятельности предприятия для выделения перспективных и убыточных направлений бизнеса;

• выявление тенденций, развивающихся на предприятии, так и на рынке.

Не следует забывать и о том, что работать с системой придется обычным людям, являющимся специалистами в своей предметной области, но зачастую обладающими весьма средними навыками в работе с компьютерами. Интерфейс информационных систем должен быть им интуитивно понятен.

**Технология внедрения информационных систем.**

- Технология построения системы по моделям "как надо", без попыток программирования действующих сейчас алгоритмов. Практика создания систем по модели "как есть" показала, что автоматизация без проведения реинжиниринга бизнес процессов и модернизации существующей системы управления не приносит желаемых результатов и не эффективна. Ведь использование в работе программных приложений - это не просто сокращение бумажных документов и рутинных операций, но и переход на новые формы ведения документооборота, учета и отчетности.

- Технология построения систем с подходом "сверху вниз". Если решение об автоматизации принято и одобрено высшим руководством, то внедрение программных модулей осуществляется с головных предприятий и подразделений, а процесс построения корпоративной системы проходит гораздо быстрее и эффективнее, чем при внедрении системы первоначально в низовые подразделения. Только при внедрении "сверху вниз" и активном содействии руководства можно изначально правильно оценить и провести весь комплекс работ без незапланированных издержек.

- Технология поэтапного внедрения. Поскольку комплексная автоматизация - это процесс, в который вовлекаются практически все структурные подразделения предприятия, технология поэтапного внедрения является наиболее предпочтительной. Первыми объектами автоматизации становятся те участки, на которых в первую очередь необходимо наладить процесс учета и формирования отчетных документов для вышестоящих органов и смежных подразделений.

- Привлечение к разработке будущих пользователей. При выполнении работ по комплексной автоматизации фирмой-интегратором меняются функции отделов информационных технологий фирмы-заказчика, и возрастает их роль в общем процессе перехода предприятия на прогрессивные методы управления.

Во время реализации проекта сотрудники отделов вместе с разработчиками работают с информацией и моделями, участвуют в принятии решения по выбору технологических решений и, самое главное, организуют взаимодействие поставщиков решения и сотрудников предприятия. При эксплуатации информационной системы на плечи сотрудников автоматизированной системы управления ложится обслуживание и сопровождение системы (если не заключен договор на сопровождение с фирмой-поставщиком). Специалисты заказчика являются инициаторами и исполнителями подготовки предложений по совершенствованию и развитию существующей системы. Это позволяет им лучше приспособить ее к своим требованиям, поэтому эти требования должны быть основательно продуманы, чтобы информационные технологии не использовались там, где легко можно справиться с задачами управления с помощью карандаша и листа бумаги.

Система должна поддерживать такую схему взаимодействия между модулями и автоматизированными рабочими местами, которая отвечала бы требованиям и техническим возможностям пользователя. Важнейшими параметрами информационной системы являются надежность, масштабируемость, безопасность, поэтому при создании таких систем используется архитектура клиент-сервер. Эта архитектура позволяет распределить работу между клиентской и серверной частями системы, предусматривает развитие и совершенствование в соответствии с особенностями решаемых задач. В последние годы наблюдается устойчивая тенденция увеличения спроса на клиент-серверные приложения, которые обладают в области учета и управления большими возможностями, чем файл-серверные системы при обработке больших объемов данных, возможностью создания распределенных систем, а также достаточной интеграцией с другими системами.

Разработка методик подготовки и ведения нормативно-справочной информации.

Разработка классификации объектов нормативно-справочной информации, их определение и детальное описание их свойств. Подготовка образцов описания данных объектов.

Базовый состав объектов нормативно-справочной информации включает:

ü производственную структуру предприятия (рабочие центры и их группировки, их идентификация и классификация);

ü территориальную структуру предприятия (площадки и места хранения запасов и их группировки, их идентификация и классификация);

ü финансовую структуру предприятия (центры финансовой ответственности и их группировки, их идентификация и классификация);

ü номенклатурные позиции, их классификация и группировки;

ü спецификации номенклатурных позиций (структуры продуктов);

ü технологические маршруты (в том числе учетные точки в нём для построения системы производственного учета);

ü другие данные.

Формирование рекомендаций по устранению выявленного дефицита данных об объектах нормативно-справочной информации в существующей информационной системе.

Аудит процесса подготовки и ведения справочников нормативно-справочной информации на предмет соответствия задачам предприятия и принципам формирования ИС.

Выделение категорий затрат, изучение и определение методик расчета себестоимости продукции (в части прямых затрат и переменных косвенных затрат).

Подготовка бизнес-процессов.

Анализ и формирование рекомендаций по совершенствованию бизнес-процессов планирования операционной деятельности, ее исполнения, а также ведения нормативных данных для поддержки операционной деятельности.

Анализ и формирование рекомендаций по достижению соответствия бизнес-процессов рекомендациям методики ИС

Разработка моделей бизнес-процессов сбыта, производства, закупок, планирования и других, в соответствии с предметной областью проекта, на различных уровнях иерархии плановых решений, необходимых предприятию Заказчика бизнес-процессов, которые будут поддерживаться системой

Выбор программной системы автоматизации планирования и учета на производстве.

Анализ рынка программного обеспечения.

Разработка системы аналитической отчетности, которую необходимо будет получать средствами системы.

Разработка требований к информационной системе.

Подготовка технического задания на выбор и внедрение информационной системы.

Организация проведения конкурса по выбору программного обеспечения для информационной системы.

Необходимо учитывать уровень подготовки специалистов, которым предстоит работать с приложением, а также назначение приложения. Если пользователи имеют большой опыт работы с программными приложениями, то можно использовать многооконный интерфейс, выпадающие меню и т. д.

Если же речь идет о сотрудниках, для которых сложно “двумя руками три кнопки нажать”, то интерфейс системы должен быть как можно более простым, а последовательность действий - очевидной. Аналогично, если в режиме использования критичен быстрый ввод данных, то на первое место выходит удобство интерфейса. Имеет смысл еще до сдачи информационной системы в эксплуатацию предоставить разработчикам возможность попробовать себя в роли конечных пользователей.

Виды информационных систем в организации

Так как имеются различные интересы, особенности и уровни в организации, существуют различные виды информационных систем. Никакая единственная система не может полностью обеспечивать потребности организации во всей информации. Организацию можно разделить на уровни: стратегический, управленческий, знания и эксплуатационный; и на функциональные области типа продажи и маркетинга, производства, финансов, бухгалтерского учета и человеческих ресурсов. Системы создаются чтобы обслужить эти различные организационные интересы. Различные организационные уровни обслуживают четыре главных типа информационных систем: системы с эксплуатационным уровнем, системы уровня знания, системы уровня управления и системы со стратегическим уровнем.

Системы эксплуатационного уровня поддерживают управляющих операциями, следят за элементарными действиями организации типа продажи, платежей, обналичивают депозиты, платежную ведомость. Основная цель системы на этом уровне состоит в том, чтобы ответить на обычные вопросы и проводить потоки транзакций через организацию. Чтобы отвечать на эти виды вопросов, информация вообще должна быть легко доступна, оперативна и точна.

Системы уровня знания поддерживают работников знания и обработчиков данных в организации. Цель систем уровня знания состоит в том, чтобы помочь интегрировать новое знание в бизнес и помогать организации управлять потоком документов. Системы уровня знания, особенно в форме рабочих станций и офисных систем, сегодня являются наиболее быстрорастущими приложениями в бизнесе.

Системы уровня управления разработаны, чтобы обслуживать контроль, управление, принятие решений и административные действия средних менеджеров. Они определяют, хорошо ли работают объекты, и периодически извещают об этом. Например, система управления перемещениями сообщает о перемещении общего количества товара, равномерности работы торгового отдела и отдела, финансирующего затраты для служащих во всех разделах компании, отмечая, где фактические издержки превышают бюджеты.

Некоторые системы уровня управления поддерживают необычное принятие решений. Они имеют тенденцию сосредоточиться на менее структурных решениях, для которых информационные требования не всегда ясны. Системы стратегического уровня – это инструмент помощи руководителям высшего уровня, которые подготавливают стратегические исследования и длительные тренды в фирме и в деловом окружении. Их основное назначение – приводить в соответствие изменения в условиях эксплуатации с существующей организационной возможностью.

Информационные системы могут также быть дифференцированы функциональным образом. Главные организационные функции типа продажи и маркетинга, производства, финансов, бухгалтерского учета и человеческих ресурсов обслуживаются собственными информационными системами. В больших организациях подфункции каждой из этих главных функций также имеют собственные информационные системы. Например, функция производства могла бы иметь системы для управления запасами, управления процессом, обслуживания завода, автоматизированной разработки и материального планирования требований.

Типичная организация имеет системы различных уровней: эксплуатационную, управленческую, знания и стратегическую для каждой функциональной области. Например, коммерческая функция имеет коммерческую систему на эксплуатационном уровне, чтобы делать запись ежедневных коммерческих данных и обрабатывать заказы. Система уровня знания создает соответствующие дисплеи для демонстрации изделий фирмы. Системы уровня управления отслеживают ежемесячные коммерческие данные всех коммерческих территорий и докладывают о территориях, где продажа превышает ожидаемый уровень или падает ниже ожидаемого уровня. Система прогноза предсказывает коммерческие тренды в течение пятилетнего периода – обслуживает стратегический уровень

Внедрение информационных систем.Основные проблемы и задачи

Отсутствие постановки задачи менеджмента на предприятии.

Большинство руководителей управляют своим предприятием только исходя из своего опыта, своей интуиции, своего видения и весьма неструктурированных данных о его состоянии и динамике. Как правило, если руководителя попросить описать в каком-либо виде структуру деятельности своего предприятия или набор положений, исходя из которых он принимает управленческие решения, дело достаточно быстро заходит в тупик. Грамотная постановка задач менеджмента является важнейшим фактором, влияющим как и на успех деятельности предприятия в целом, так и на успех проекта автоматизации. Поэтому, первое, что необходимо сделать для того, чтобы проект внедрения информационной системы управления предприятием оказался удачным - максимально формализовать все те контуры управления, которые собственно Вы планируете автоматизировать.В большинстве случаев, для осуществления этого не обойтись без привлечения профессиональных консультантов, но по опыту, затраты на консультантов просто не сопоставимы с убытками от проваленного проекта автоматизации.

Необходимость в частичной или полной реорганизации структуры предприятия.

Прежде чем приступать к внедрению информационной системы управления на предприятии обычно необходимо произвести частичную реорганизацию его структуры и технологий ведения бизнеса. Поэтому, одним из важнейших этапов проекта внедрения,является полное и достоверное обследование предприятия во всех аспектах его деятельности. На основе заключения, полученного в результате обследования,строится вся дальнейшая схема построения корпоративной информационной системы. Несомненно, можно автоматизировать все, про принципу "как есть", однако, этого не следует делать по ряду причин. Дело в том, что в результате обследования обычно фиксируется большое количество мест возникновения необоснованных дополнительных затрат, а также противоречий в организационной структуре, устранение которых позволило бы уменьшить производственные и логистические издержки, а также существенно сократить время исполнения различных этапов основных бизнес-процессов. Нельзя автоматизировать хаос, ибо в результате этого получится автоматизированный хаос. Реорганизация может быть проведена в ряде локальных точек, где она объективно необходима, что не повлечет за собой ощутимый спад активности текущей коммерческой деятельности.

Необходимость в изменении технологии работы с информацией, и принципов ведения бизнеса

Эффективно построенная информационная система не может не внести изменений в существующую технологию планирования и контроля, а так же управления процессами. Oдними из самых важных для руководителя особенностей корпоративной информационной системы, являются модули управленческого учета и финансового контроллинга. Теперь каждое функциональное подразделение может быть определено как центр учета, с соответствующим уровнем ответственности его руководителя. Это в свою очередь повышает ответственность каждого из таких руководителей, и предоставляет в руки высших менеджеров эффективный инструментарий для чёткого контроля исполнения отдельных планов и бюджетов.

При наличии информационной системы управления предприятием, руководитель способен получать актуальную и достоверную информацию обо всех срезах деятельности компании, без временных задержек и излишних передаточных звеньев. Кроме того, информация подаётся руководителю в удобном виде "с листа" при отсутствии человеческих факторов, которые могут предвзято или субъективно трактовать информацию при передаче. Однако справедливо было бы заметить, что некоторые руководители не привыкли принимать управленческие решения по информации в чистом виде, если к ней не приложено мнение человека, который ее доставил. Такой подход в принципе имеет право на жизнь и при наличии информационной системы управления предприятием, однако часто он негативно отражается на объективности менеджмента. Внедрение информационной системы управления предприятием вносит существенные изменения в управление бизнес-процессами. Каждый документ, отображающий в информационном поле течение или завершение того или иного сквозного процесса, в интегрированной системе создается автоматически, на основании первичного документа, открывшего процесс. Сотрудники, ответственные за этот процесс лишь контролируют и, при необходимости, вносят изменения в позиции построенных системой документов. Например, заказчик разместил заказ на продукцию, который должен быть исполнен к определенному числу месяца. Заказ вводится в систему, на основании его системой автоматически создается счет (на основе существующих алгоритмов ценообразования),счет пересылается заказчику, а заказ направляется в производственный модуль, где происходит разузлование заказанного вида продукции на отдельные комплектующие. На основе списка комплектующих в модуле закупок системой создаются заказы на их закупку, а производственный модуль соответствующим образом оптимизирует производственную программу, чтобы заказ был исполнен точно к сроку. Естественно, в реальной жизни возможны различные варианты неустранимых срывов поставок комплектующих, поломки оборудования и т.д., поэтому каждый этап выполнения заказа должен строго контролироваться ответственным за него кругом сотрудников, которые, в случае необходимости, должны создать управленческое воздействие на систему, чтобы избежать нежелательных последствий или уменьшить их. Не стоит полагать, что работать при наличии информационной системы управления предприятием станет проще. Наоборот, существенное сокращение бумажной волокиты ускоряет процесс и повышает качество обработки заказов, поднимает конкуренто способность и рентабельность предприятия в целом, а все это требует большей собранности, компетенции и ответственности исполнителей. Возможно, что существующая производственная база не будет справляться с новым потоком заказов, и в нее тоже нужно будет вносить организационные и технологические реформы, которые впоследствии положительно скажутся на процветании предприятия.

Сопротивление сотрудников предприятия

Сложности машиностроительных предприятий очень похожи на сложности большинства российских предприятий и написано об этом уже очень много.

Однако можно попробовать выделить несколько довольно интересных и редко упоминаемых в прессе проблем, характерных именно для машиностроения.

1. Кому из директоров система больше нужна, тот пусть и занимается ее внедрением. Часто решение о внедрении информационной системы является не решением компании или хотя бы консолидированным решением топ-менеджеров и акционеров, а решением одного из функциональных руководителей, например, финансового директора или директора по производству. В таком случае информационная система внедряется в интересах именно данного руководителя, большинство же топ-менеджеров предприятия не принимает участие в процессе внедрения и, как следствие, может неоднозначно оценивать как сам процесс, так и результаты внедрения системы в целом.

Более того, зачастую при внедрении системы акценты могут смещаться настолько, что для дальнейшего комплексного внедрения уже сделанные работы приходится серьезно переделывать. Примеры такой автоматизации встречаются. На вопрос "Почему?" «нам надо было попробовать сделать неправильно, чтобы все руководство осознало, как не следует проводить автоматизацию управления. Зато теперь все руководители понимают необходимость личного участия в работах по комплексной автоматизации предприятия.»

Сложно сказать, насколько такой подход себя оправдывает, но и оспаривать инициаторов внедрения информационных систем на предприятии сложно, поскольку иначе "предприятие вообще бы не обратило внимания на вопросы автоматизации". Фактически в описанном случае компания на собственных ошибках проходит обучение тому, как правильно внедрять информационные системы, и, как следствие, при повторных проектах руководство предприятия уже значительно точнее понимает, каких целей хочет добиться в результате внедрения системы, какие усилия необходимо организовать для ведения проекта, для чего нужно приглашать профессиональных консультантов и сколько реально может стоить проект внедрения комплексной информационной системы управления.

2. При внедрении информационной систем управления предприятием в большинстве случаев возникает активное сопротивление сотрудников на местах, которое является серьезным препятствием для консультантов и вполне способно сорвать или существенно затянуть проект внедрения Если система не нравится сотрудникам, значит она плохая.

Удивительно, но даже на крупных предприятиях иногда мнение сотрудника низшего звена (что-то не нравится, не подходит, не удобно, мелкий шрифт, "вообще старая система была понятнее" и т.п.) вполне способно притормозить внедрение системы.

Часто руководители предприятия, особенно, если они не уделяют достаточного внимания процессу внедрения, судят о качестве системы по отзывам персонала, интересы которого зачастую расходятся с интересами руководителя. В результате, вместо оптимального достижения целей внедрения системы управления значительное время тратится на переделку интерфейсов, реализацию всяческих "рюшечек" и "бантиков", притом, что реальной ценности такие доработки не представляют, но значительно влияют на бюджет и главное - сроки внедрения. Руководители предприятия, принявшие решение автоматизировать свой бизнес , в таких случаях должны всячески содействовать ответственной группе специалистов, проводящей внедрение информационной системы управления предприятием ,вести разъяснительную работу с кадрами, и, кроме того:

• Создать у сотрудников всех уровней твёрдое ощущение неизбежности внедрения;

• Наделить руководителя проекта внедрения достаточными полномочиями, поскольку сопротивление иногда (часто подсознательно, или в результате неоправданных амбиций) возникает даже на уровне топ-менеджеров (Нередко сложности могут возникнуть в тот самый момент, когда руководитель вдруг осознает, что с внедрением системы он попадает в своего рода ловушку: рост информированности, который обеспечивает система, сводит к минимуму неопределенность текущей производственной ситуации и тем самым ограничивает возможность принятия решений исходя лишь из субъективного мнения. Новая технология управления также способствует выявлению некомпетентности руководства.);

• Всегда подкреплять все организационные решения по вопросам внедрения изданием соответствующих приказов и письменных распоряжений.

Во время проведения опытной эксплуатации и при переходе к промышленной эксплуатации системы в течение некоторого времени приходится вести дела, как и в новой системе, так и продолжать ведение их традиционными способами (поддерживать бумажный документооборот и существовавшие ранее системы). В связи с этим, отдельные этапы проекта внедрения системы могут затягиваться под предлогом того, что у сотрудников и так хватает срочной работы по прямому назначению, а освоение системы является второстепенным и отвлекающим занятием. В таких случаях руководителю предприятия, помимо ведения разъяснительной работы с уклоняющимися от освоения новых технологий сотрудниками необходимо:

1. Повысить уровень мотивации сотрудников к освоению системы в форме поощрений и благодарностей;

2. Принять организационные меры к сокращению срока параллельного ведения дел.

•Необходимость в формировании квалифицированной группы внедрения и сопровождения системы, выбор сильного руководителя группы. Внедрение большинства крупных систем автоматизации управления предприятием производитсяпо следующей технологии: на предприятии формируется небольшая (3-6 человек)рабочая группа, которая проходит максимально полное обучение работе с системой, затем на эту группу ложится значительная часть работы по внедрению системы и дальнейшему ее сопровождению. Применение подобной технологии вызвано двумя факторами: во-первых, тем, что предприятие обычно заинтересовано в том, чтобы у него под рукой были специалисты, которые могут оперативно решать большинство рабочих вопросов при настройке и эксплуатации системы, а во-вторых, обучение своих сотрудников и их использование, всегда существенно дешевле аутсорсинга. Таким образом, формирование сильной рабочей группы является залогом успешной реализации проекта внедрения.

Особенно важным вопросом является выбор руководителя такой группы и администратора системы. Руководитель, помимо знаний базовых компьютерных технологий, должен обладать глубокими знаниями в области ведения бизнеса и управления. В отечественной практике, при внедрении систем такую роль, как правило, играет начальник отдела АСУП или ему аналогичного. Основными правилами организации рабочей группы являются следующие принципы:

• Специалистов рабочей группы необходимо назначать с учетом следующих требований: знание современных компьютерных технологий (и желание осваивать их в дальнейшем), коммуникабельность, ответственность, дисциплинированность;

• С особой ответственностью следует подходить к выбору и назначению администратора системы, так как ему будет доступна практически вся корпоративная информация;

• Возможное увольнение специалистов из группы внедрения в процессе внедрения проекта может крайне негативно отразиться на его результатах. Поэтому членов группы следует выбирать из преданных и надежных сотрудников и выработать систему поддержки этой преданности в течение всего проекта;

• После определения сотрудников, входящих в группу внедрения, руководитель проекта должен четко расписать круг решаемых каждым из них задач, формы планов и отчетов, а также длину отчетного периода. В наилучшем случае, отчетным периодом должен быть один день.

Подведем итог

Можно сделать вывод, что общей стратегической целью создания корпоративной информационной системы является повышение управляемости, что позволяет увеличить экономическую эффективность и качественно улучшить производственные показатели предприятия. Достижение этой цели в рамках создания корпоративной информационной системы должно быть реализовано за счёт применения на предприятии всего комплекса информационных технологий, а именно:

- сбора достоверной информации;

- оперативной обработки данных о фактах производственно-хозяйственной деятельности;

- аналитической поддержки принятия управленческих решений;

В структуре предприятия имеет смысл запланировать отдел поддержки внедренных систем, чтобы после опытной эксплуатации эта рутинная работа не висела на сильных и талантливых программистах, которых целесообразно направить на реализацию новых проектов. Цель компьютерных систем поддержки принятия стратегических решений состоит в том, чтобы обеспечить высшее руководство непосредственным и свободным доступом к информации относительно ключевых факторов, являющихся критическими при реализации стратегических целей фирмы.

Следовательно, ИС должны быть просты в эксплуатации и понимании. Они обеспечивают доступ к множеству внутренних и внешних баз данных, активно используя графическое представление данных.

- своевременного информирования персонала о результатах производственного планирования и ресурсном обеспечении.

Перед тем, как осуществлять проект внедрения максимально формализуйте его цели;

Установить высокий приоритет процессу внедрения системы, среди остальных организационных и коммерческих процессов. Наделите высокими полномочиями руководителя проекта;

Создать среди всех сотрудников предприятия атмосферу неотвратимости внедрения и старайтесь организационными мерами повысить темп освоения новых технологий;

Внедрение информационной системы управления предприятием как ремонт - его невозможно закончить, можно лишь прекратить. Так что внедрение по сути никогда не закончится, система должна все время совершенствоваться в процессе своей промышленной эксплуатациями вместе с прогрессом информационных технологий и методологий управления деятельностью вашего предприятия.

1. Опишите процесс оценки качества функционирования информационной системы.

Понятие *качества* ИС соответствует понятию о том, что система успешно справляется со всеми возлагаемыми на нее задачами, имеет хорошие показатели надежности, имеет приемлемую стоимость, удобна в эксплуатации и обслуживании, легко сопрягается с другими системами и, в случае необходимости, может быть модифицирована.

Разные группы пользователей имеют различные точки зрения на характеристики качества ИС. Например, если задать вопрос о том, какой должна быть хорошая ИС, то от пользователя можно, скорее всего, получить следующие варианты ответов:

* • должна иметь хорошую производительность;
* • должна иметь широкие функциональные возможности;
* • должна быть удобна в эксплуатации;
* • должна быть надежна.

Менеджер даст, скорее всего, другие варианты ответов:

* • не должна быть очень дорогой;
* • не должна быть очень дорогой в эксплуатации;
* • не должна морально устаревать в течение возможно более длительного промежутка времени, а, в случае необходимости, может быть легко модифицирована.

Для системного администратора наиболее важными могут оказаться такие характеристики системы, как:

* • должна быть надежна и стабильна в работе;
* • должна быть проста в администрировании;
* • должна иметь хорошую эксплуатационную документацию;
* • должна иметь хорошую поддержку изготовителем.

Другие заинтересованные лица могут иметь свои точки зрения на то, какой должна быть качественная система.

Качество ИС связано с *дефектами,* заложенными на этапе проектирования и проявляющиеся в процессе эксплуатации. Любые свойства ИС, в том числе и дефектологические, могут проявляться лишь во взаимодействии с внешней средой, включающей технические средства, персонал, информационное и программное окружение.

В зависимости от целей исследования и этапов жизненного цикла ИС дефектологические свойства разделяют на дефектогенность, дефек- табельность и дефектоскопичность.

*Дефектогенность* определяется влиянием следующих факторов:

* • количество разработчиков ИС, их профессиональные и психофизиологические характеристики;
* • условия и организация процесса разработки ИС;
* • характеристики инструментальных средств и компонент ИС;
* • сложность задач, решаемых ИС;
* • степень агрессивности внешней среды (потенциальная возможность внешней среды вносить преднамеренные дефекты, например, воздействие вирусов).

*Дефектабелъность* характеризует наличие дефектов ИС и определяется их количеством, местонахождением и др. факторами, влияющими на дефектабельность, являются:

* • структурно-конструктивные особенности ИС;
* • интенсивность и характеристики ошибок, приводящих к дефектам.

*Дефектоскопичностъ* характеризует возможность проявления дефектов в виде отказов и сбоев в процессе отладки, испытаний или эксплуатации. На дефектоскопичность влияют:

* • количество, типы и характер распределения дефектов в ИС;
* • устойчивость ИС к проявлению дефектов;
* • характеристики средств контроля и диагностики дефектов;
* • квалификация обслуживающего персонала.

Оценка качества ИС является крайне сложной задачей ввиду многообразия интересов пользователей. Поэтому невозможно предложить одну универсальную меру качества и приходится использовать ряд характеристик, охватывающих весь спектр предъявляемых требований. Наиболее близки к задачам оценки качества ИС модели качества программного обеспечения, являющегося одной из важных составных частей ИС. В настоящее время используется несколько абстрактных моделей качества программного обеспечения, основанных на определениях характеристики качества, показателя качества, критерия и метрики.

Критерий может быть определен как независимый атрибут ИС или процесса ее создания. С помощью такого критерия может быть измерена характеристика качества ИС на основе той или иной метрики. Совокупность нескольких критериев определяет показатель качества, формируемый исходя из требований, предъявляемых к ИС. В настоящее время наибольшее распространение получила иерархическая модель взаимосвязи компонент качества ИС. Вначале определяется характеристика качества, в числе которых могут быть, например, общая полезность, исходная полезность, удобство эксплуатации. Далее формируются показатели, к числу которых могут быть отнесены: практичность, целостность, корректность, удобство обслуживания, оцениваемость, гибкость, адаптируемость, мобильность, возможность взаимодействия. Каждому показателю качества ставится в соотвествие группа критериев. Для указанных выше показателей ниже приведены возможные критерии. Надо отметить, что один и тот же критерий может характеризовать несколько показателей.

С помощью метрик можно дать количественную или качественную оценку качества ИС. Различают следующие виды метрик и шкал для измерения критериев.

*Первый тип* — метрики, которые используют интервальную шкалу, характеризуемую относительными величинами или реально измеряемыми физическими показателями, например, временем наработки на отказ, вероятностью ошибки, объемом информации и др.

*Второй тип* — метрики, которым соответствует порядковая шкала, позволяющая ранжировать характеристики путем сравнения с опорными значениями.

*Третий тип* — метрики, которым соответствуют номинальная или категорированная шкала, определяющая наличие рассматриваемого свойства или признака у рассматриваемого объекта без учета градаций по этому признаку. Так, например, интерфейс может быть «простым для понимания», «умеренно простым», «сложным для понимания».

Развитием иерархического подхода является представленная на рис. 6.11 модель классификации критериев качества информационных систем. С помощью функциональных критериев оценивается степень выполнения ИС основных целей или задач. Конструктивные критерии предназначены для оценки компонент ИС, не зависящих от целевого назначения.

Поскольку в современных ИС ключевой компонентой является программная компонента, а пользователи, работающие с системой, в подавляющем большинстве случаев взаимодействуют непосредственно с программной компонентой, поэтому показатели качества информационных и программных систем в значительной степени совпадают.

Качество программного обеспечения определяется в стандарте *ISO*9126 как вся совокупность его характеристик, относящихся к возможности удовлетворять высказанные или подразумеваемые потребности всех заинтересованных лиц.

Различаются понятия внутреннего качества, связанного с характеристиками ПО самого по себе, без учета его поведения; внешнего качества, характеризующего ПО с точки зрения его поведения; и *качества*ПО при использовании в различных контекстах — того качества, которое ощущается пользователями при конкретных сценариях работы ПО. Для всех этих аспектов качества введены метрики, позволяющие оценить их. Кроме того, для создания добротного ПО существенно качество технологических процессов его разработки.

*ISO* 9126 — это международный стандарт, определяющий оценочные характеристики качества программного обеспечения. Российский аналог стандарта ГОСТ 28195. Стандарт разделяется на четыре части, описывающие следующие вопросы: модель качества; внешние метрики качества; внутренние метрики качества; метрики качества в использовании.

Вторая и третья части стандарта *ISO* 9126-2,3 посвящены формализации соответственно внешних и внутренних метрик характеристик качества сложных программных систем. В ней изложены содержание и общие рекомендации по использованию соответствующих метрик и взаимосвязей между типами метрик.

Четвертая часть стандарта *ISO* 9126-4 предназначена для покупателей, поставщиков, разработчиков, сопровождающих, пользователей и менеджеров качества программные системы (ПС). В ней повторена концепция трех типов метрик, а также аннотированы рекомендуемые виды измерений характеристик. Модель качества, установленная в первой части стандарта *ISO* 9126-1, классифицирует качество ПО в шести структурных наборах характеристик:

* • функциональность;
* • надежность;
* • эффективность (производительность);
* • удобство использования;
* • удобство сопровождения;
* • переносимость.

Перечисленные характеристики, в свою очередь, детализированы подхарактеристиками (субхарактеристиками) (рис. 6.12). Ниже приведены определения этих характеристик и атрибутов по стандарту *ISO*9126:2001.

Функциональность (*functionality)* определяется как способность ПО в определенных условиях решать задачи, нужные пользователям.

Для данной характеристики выделяются следующие субхарактеристики:

* • функциональная пригодность;
* • точность;
* • способность к взаимодействию;
* • защищенность;
* • соответствие стандартам и правилам.

*Функциональная пригодность* *(suitability*) — способность решать нужный набор задач.

*Точность* (*accuracy*) — способность выдавать нужные результаты.

*Способность к взаимодействию* (*interoperability*) — это способность взаимодействовать с нужным набором других систем.

*Соответствие стандартам и правилам* (*compliance*) — соответствие ПО имеющимся индустриальным стандартам, нормативным и законодательным актам, другим регулирующим нормам.

*Защищенность* *(security*) — способность предотвращать неавтори- зированный, т.е. без указания лица, пытающегося его осуществить, и неразрешенный доступ к данным и программам.

Надежность (*reliability*) — это способность ПО поддерживать определенную работоспособность в заданных условиях.

Для данной характеристики выделяются следующие субхарактеристики:

* • зрелость;
* • устойчивость к отказам;
* • способность к восстановлению;
* • соответствие стандартам.

*Зрелость, завершенность* *(maturity)* — величина, обратная частоте отказов ПО. Обычно измеряется средним временем работы без сбоев и величиной, обратной вероятности возникновения отказа за данный период времени.

*Устойчивость к отказам (fault tolerance*) — способность поддерживать заданный уровень работоспособности при отказах и нарушениях правил взаимодействия с окружением.

*Способность к восстановлению* *(recoverability*) — способность восстанавливать определенный уровень работоспособности и целостность данных после отказа, необходимые для этого время и ресурсы.

*Соответствие стандартам* надежности *(reliability compliance).*

Удобство использования (*usability*), или практичность, определяется как способность ПО быть удобным в обучении и использовании, а также привлекательным для пользователей.

Для данной характеристики выделяются следующие субхарактеристики:

* • понятность;
* • удобство обучения;
* • удобство работы;
* • привлекательность;
* • соответствие стандартам.

*Понятность (understandability)* — это показатель, обратный к усилиям, которые затрачиваются пользователями на восприятие основных понятий ПО и осознание их применимости для решения своих задач.

*Удобство обучения (learnability*) — показатель, обратный усилиям, затрачиваемым пользователями на обучение работе с ПО.

*Удобство работы (operability)* — это показатель, обратный усилиям, предпринимаемым пользователями для решения своих задач с помощью ПО.

*Привлекательность (attractiveness)* — это способность ПО быть привлекательным для пользователей. Этот атрибут добавлен в 2001 г.

*Соответствие стандартам* определяется как удобства использования (*usability compliance*).

Производительность *(efficiency),* или эффективность, — это способность ПО при заданных условиях обеспечивать необходимую работоспособность по отношению к выделяемым для этого ресурсам. Можно определить ее и как отношение получаемых с помощью ПО результатов к затрачиваемым на это ресурсам всех типов.

Для данной характеристики выделяются следующие субхарактеристики:

* • временная эффективность;
* • эффективность использования ресурсов;
* • соответствие стандартам.

*Временная эффективность* *(time behaviour*) — способность ПО выдавать ожидаемые результаты, а также обеспечивать передачу необходимого объема данных за отведенное время.

*Эффективность использования ресурсов* *(resource utilisation*) — способность решать нужные задачи с использованием определенных объемов ресурсов определенных видов. Имеются в виду такие ресурсы, как оперативная и долговременная память, сетевые соединения, устройства ввода и вывода и пр.

*Соответствие стандартам* производительности (*efficiency compliance).*

Удобство сопровождения (*maintainability)* определяется как удобство проведения всех видов деятельности, связанных с сопровождение программ.

Для данной характеристики выделяются следующие субхарактеристики:

* • анализируемость;
* • удобство внесения изменений;
* • стабильность;
* • удобство проверки;
* • соответствие стандартам.

*Анализируемость (analyzability)*, или удобство проведения анализа, определяется как удобство проведения анализа ошибок, дефектов и недостатков, а также удобство анализа необходимости изменений и их возможных последствий.

*Удобство внесения изменений* *(changeability*) — показатель, обратный трудозатратам на выполнение необходимых изменений.

*Стабильность* *(stability*) — это показатель, обратный риску возникновения неожиданных эффектов при внесении необходимых изменений.

*Удобство проверки* *(testability*) — это показатель, обратный трудозатратам на проведение *тестирования* и других видов проверки того, что внесенные изменения привели к нужным результатам.

*Соответствие стандартам* удобства сопровождения (*maintainability compliance').*

Переносимость (*portability*) определяется как способность ПО сохранять работоспособность при переносе из одного окружения в другое, включая организационные, аппаратные и программные аспекты окружения.

Иногда эта характеристика называется в русскоязычной литературе мобильностью. Однако термин «мобильность» стоит зарезервировать для перевода *«mobility»* — способности ПО и компьютерной системы в целом сохранять работоспособность при ее физическом перемещении в пространстве.

Для данной характеристики выделяются следующие субхарактеристики:

* • адаптируемость;
* • удобство установки;
* • способность к сосуществованию;
* • удобство замены;
* • соответствие стандартам;

*Адаптируемость* (*adaptability*) — способность ПО приспосабливаться различным окружениям без проведения для этого действий, помимо заранее предусмотренных.

*Удобство установки* (*installability*) — это способность ПО быть установленным или развернутым в определенном окружении.

Способность к сосуществованию (*coexistence)* — это способность ПО сосуществовать с другими программами в общем окружении, деля с ними ресурсы.

Удобство замены (*replaceability*) другого ПО данным определяется как возможность применения данного ПО вместо других программных систем для решения тех же задач в определенном окружении.

Соответствие стандартам *переносимости* (*portability compliance).*

Перечисленные атрибуты относятся к внутреннему и внешнему *качеству ПО* согласно *ISO* 9126. Для описания *качества ПО* при использовании стандарт *ISO* 9126-4 предлагает другой, более узкий набор характеристик:

* • эффективность;
* • продуктивность;
* • безопасность;
* • удовлетворение пользователей;
* • эффективность (*effectiveness).*

Способность ПО предоставлять пользователям возможность решать их задачи с необходимой точностью при использовании в заданном контексте.

*Продуктивность* *(productivity*) — это способность ПО предоставлять пользователям определенные результаты в рамках ожидаемых затрат ресурсов.

*Безопасность* *(safety*) — это способность ПО обеспечивать необходимо низкий уровень риска нанесения ущерба жизни и здоровью людей, бизнесу, собственности или окружающей среде.

*Удовлетворение пользователей* *(satisfaction*) — это способность ПО приносить удовлетворение пользователям при использовании в заданном контексте.

Одним из путей обеспечения качества ИС является сертификация. В США Радиотехническая комиссия по аэронавтике в своем руководящем документе определяет процесс сертификации следующим образом: «Сертификация — процесс официального утверждения государственным полномочным органом выполняемой функции системы путем удостоверения, что функция удовлетворяет всем требованиям заказчика, а также государственным нормативным документам». К сожалению, в настоящее время не существует стандартов, полностью удовлетворяющим оценке качества ИС. В западно-европейских странах имеется ряд стандартов, определяющих основы сертификации программных систем. Стандарт Великобритании (BS750) описывает структурные построения программных систем, при соблюдении которых может быть документ — гарантия качества на государственном уровне. Имеется международный аналог указанного стандарта (7S09000) и аналог для стран членов НАТО (AQAP1). Существующая в нашей стране система нормативно-технических документов относит программное обеспечение к «продукции производственно-технического назначения», которая рассматривается как материальный объект. Однако программное обеспечение является, скорее, абстрактной нематериальной сферой. Существующие ГОСТы, например ГОСТ 28195—89. «Оценка качества программных средств. Общие положения», явно устарели и являются неполными.

1. Раскрыть определение понятия «CALS-технологии», опишите основную задачу, показатели CALS-технологий.

CALS-технологии призваны служить средством, интегрирующим промышленные автоматизированные системы в единую многофункциональную систему. Целью интеграции автоматизированных систем проектирования и управления является повышение эффективности создания и использования сложной техники.

В чем выражается повышение эффективности?

Во-первых, повышается качество изделий за счет более полного учета имеющейся информации при проектировании и принятии управленческих решений. Так, обоснованность решений, принимаемых в автоматизированной системе управления предприятием (АСУП), будет выше, если ЛПР (лицо, принимающее решение) и соответствующие программы АСУП имеют оперативный доступ не только к базе данных АСУП, но и к базам данных других автоматизированных систем (САПР, АСТПП и АСУТП) и, следовательно, могут оптимизировать планы работ, содержание заявок, распределение исполнителей, выделение финансов и т.п. При этом под оперативным доступом следует понимать не просто возможность считывания данных из БД, но и легкость их правильной интерпретации, т.е. согласованность по синтаксису и семантике с протоколами, принятыми в АСУП. То же относится и к другим системам, например, технологические подсистемы должны с необходимостью воспринимать и правильно интерпретировать данные, поступающие от подсистем автоматизированного конструирования. Этого не так легко добиться, если основное предприятие и организации-смежники работают с разными автоматизированными системами.

Во-вторых, сокращаются материальные и временные затраты на проектирование и изготовление продукции. Применение CALS-технологий позволяет существенно сократить объемы проектных работ, так как описания ранее выполненных удачных разработок компонентов и устройств, многих составных частей оборудования, машин и систем, проектировавшихся ранее, хранятся в базах данных сетевых серверов, доступных любому пользователю CALS-технологии. Доступность опять же обеспечивается согласованностью форматов, способов, руководств в разных частях общей интегрированной системы. Кроме того, появляются более широкие возможности для специализации предприятий, вплоть до создания виртуальных предприятий, что также способствует снижению затрат.

В-третьих, существенно снижаются затраты на эксплуатацию, благодаря реализации функций интегрированной логистической поддержки. Существенно облегчается решение проблем ремонтопригодности, интеграции продукции в различного рода системы и среды, адаптации к меняющимся условиям эксплуатации и т.п.

Эти преимущества интеграции данных достигаются применением современных CALS-технологий.

Промышленные автоматизированные системы могут работать автономно, и в настоящее время так обычно и происходит. Однако эффективность автоматизации будет заметно выше, если данные, генерируемые в одной из систем, будут доступны в других системах, поскольку принимаемые в них решения станут более обоснованными.

Чтобы достичь должного уровня взаимодействия промышленных автоматизированных систем требуется создание единого информационного пространства в рамках как отдельных предприятий, так и, что более важно, в рамках объединения предприятий. Единое информационное пространство обеспечивается благодаря унификации как формы, так и содержания информации о конкретных изделиях на различных этапах их жизненного цикла.

Унификация формы достигается использованием стандартных форматов и языков представления информации в межпрограммных обменах и при документировании.

Унификация содержания, понимаемая как однозначная правильная интерпретация данных о конкретном изделии на всех этапах его жизненного цикла, обеспечивается разработкой онтологий (метаописаний) приложений, закрепляемых в прикладных протоколах CALS.

Унификация перечней и наименований сущностей, атрибутов и отношений в определенных предметных областях является основой для единого электронного описания изделия в CALS-пространстве.

1. Опишите организацию процесса обновления в информационной системе, регламенты обновления.

Любая операционная система, как и программные продукты, через какое-то время после установки должна обновляться. Обновления выпускаются для:

1. устранения в системе безопасности;
2. обеспечения совместимости со вновь появившимися на рынке комплектующими компьютеров;
3. оптимизации программного кода;
4. повышения производительности всей системы;

Если служба «Центр обновления Windows» включена, и некоторые программные компоненты системы, которые связанны с работой службы обновления, нуждаются в обновлении для ее функционирования, то эти обновления должны устанавливаться перед проверкой, загрузкой и установкой любых других обновлений. Эти обязательные обновления исправляют ошибки, а также обеспечивают усовершенствования и поддерживают совместимость с серверами корпорации Майкрософт, поддерживающими работу службы. Если служба обновления отключена, то получать обновления для операционной системы будет невозможно.

Обновления представляют собой дополнения к программному обеспечению, предназначенные для предотвращения или устранения проблем и улучшения работы компьютера. Обновления безопасности для Windows способствуют защите от новых и существующих угроз для конфиденциальности и устойчивой работы компьютера. Оптимальный способ получения обновлений безопасности - включить автоматическое обновление Windows и всегда оставаться в курсе последних проблем, связанных с безопасностью и предоставить операционной системе самостоятельно заботиться о своей безопасности. В этой статье речь пойдет именно о Центре обновления Windows.

Желательно обновлять компьютер как можно чаще. В этом случае использования автоматического обновления, операционная система Windows устанавливает новые обновления, как только они становятся доступными. Если не устанавливать обновления, то компьютер может подвергнуться риску в плане безопасности или же могут возникнуть нежелательные неполадки в работе Windows или программ. Каждый день появляется все больше и больше новых вредоносных программ, использующих уязвимости Windows и другого программного обеспечения для нанесения ущерба, и получения доступа к компьютеру и данным. Обновления Windows и другого программного обеспечения позволяют устранить уязвимости вскоре после их обнаружения. Если отложить установку обновлений, компьютер может стать уязвимым для таких угроз.

Обновления и программное обеспечение от Microsoft для продуктов Microsoft являются бесплатным предложением от службы поддержки, так что можно не волноваться за то, что с вас будет взиматься дополнительная плата за обеспечение надежности вашей системы. Чтобы узнать, являются ли обновления других программ бесплатными, обращайтесь к соответствующему издателю или изготовителю. При загрузке и установке обновлений различных программ в зависимости от типа подключения к Интернету может взиматься стандартная плата за местные или междугородные телефонные переговоры, а также плата за пользование Интернетом. В связи с тем, что обновления применяются к Windows и установленным на компьютере программам независимо от того, кто ими пользуется, после установки обновлений они будут доступны для всех пользователей компьютера.

Все обновления подразделяются на: важные, рекомендуемые, необязательные и основные. Далее приводится их описание:

1. Важные обновления обеспечивают существенные преимущества в безопасности, конфиденциальности и надежности. Их следует устанавливать сразу же, как только они становятся доступны, и можно выполнять установку автоматически с помощью **«Центра обновления Windows»**.
2. Рекомендуемые обновления могут устранять менее существенные проблемы или делать использование компьютера более удобным. Хотя эти обновления не предназначены для устранения существенных недостатков в работе компьютера или программного обеспечения Windows, их установка может привести к заметным улучшениям. Их можно устанавливать автоматически.
3. К необязательным обновлениям относятся обновления, драйверы или новое программное обеспечение Майкрософт, делающее использование компьютера более удобным. Их можно устанавливать только вручную.
4. К остальным обновлениям можно отнести все обновления, которые не входят в состав важных, рекомендуемых или необязательных обновлений.

В зависимости от типа обновления в **«Центре обновления Windows»** предлагаются следующие возможности.

1. Обновления безопасности. Это открыто распространяемые исправления уязвимостей определенных продуктов. Уязвимости различаются по уровню серьезности и указаны в бюллетене по безопасности Майкрософт как критические, важные, средние или низкие.
2. Критические обновления. Это открыто распространяемые исправления определенных проблем, которые связаны с критическими ошибками, не относящимися к безопасности.
3. Пакеты обновления. Протестированные наборы программных средств, включающие в себя исправления, обновления безопасности, критические и обычные обновления, а также дополнительные исправления проблем, обнаруженных при внутреннем тестировании после выпуска продукта. Пакеты обновления могут содержать небольшое количество изменений оформления или функций, запрошенных пользователями.
4. Охарактеризуйте процесс тестирования программного обеспечения в процессе внедрения и эксплуатации.

Любая программа может содержать в себе ошибки. Компилятор способен выявлять только синтаксические ошибки, но не способен отслеживать семантику. Большинство ошибок проявляется в ходе работы программы, при этом они могут возникать не всегда, а лишь при определенных условиях. Таким образом, успешная компиляция программы и выполнение этой программы в одних и тех же условиях не гарантируют отсутствие ошибок.

Для выявления ошибок в программах ЖЦ разработки ПО предусматривает процесс тестирования, который является достаточно трудоемким и занимает больше времени, чем кодирование. (Г. Майерс дает оценку 1/3 для тестирования, при том, что кодированиезанимает примерно 1/6.) Тестируемое ПО обычно называют SUT - Software Under Test. Цель тестирования - не убедиться в безошибочной работоспособности программы, а наоборот - найти ошибки. Поэтому в первую очередь возникает вопрос: а что есть ошибка в программе?

Заметим, что к этому моменту программа уже представляет собой выполнимый процессором набор команд, т.е. с точки зрения процессора она корректна. Даже если при каких-то условиях программа аварийно завершает свое выполнение или "портит" другие процессы, сразу нельзя сказать, что это ошибка в программе - возможно, так было задумано. Таким образом, ошибки необходимо рассматривать с точки зрения пользователя, основываясь на дополнительной информации, т.е. неком описании того, что должна делать программа (это же описание может включать в себя требование о том, чтобы программа никогда не завершалась аварийно и др.).

При рассмотрении вопросов анализа программного кода порой удобнее применять ранее рассмотренную в разделе 1 модель жизненного цикла. Ее часто называют V-образной из-за расположения блоков на рисунке (рис. 1).

Нисходящая левая ветвь модели отражает поэтапную последовательность преобразования одних программных документов в другие: SYS - системных требований в SRD - требования к программному обеспечению, проектированию и формированию DDD - описания архитектуры системы и, наконец, разработке CODE - кода программ. Восходящая правая ветвь отражает процесс верификации разработанного программного обеспечения.

На первом этапе путем тестирования производится модульная верификация (MV), при которой поведение исполняемого программного кода проверяется на соответствие его DDD-описанию. Это наиболее трудоемкая и скрупулезная часть исследования. Она часто требует написания драйверов - моделей модулей, вызывающих процедуры тестируемого модуля, и заглушек - моделей процедур других модулей, вызываемых из тестируемого. Часто в MV отдельно выделяют процесс тестирования межмодульных связей, описанных в DDD.

На втором этапе производится комплексная верификация (CV) реализованного программного обеспечения по отношению к требованиям. Наконец, производится комплексная интеграция (CI) и проверка всей системы: пользователь, аппаратура и программное обеспечение. При грамотном процессе разработки уже на этапах нисходящей ветви для каждого требования определяется, на каком уровне верификации должна будет проводиться проверка его соблюдения.

При этом следует исходить из предположения, что ошибки всегда есть. Тестирование можно считать успешным, если найдены ошибки, а не наоборот. В достаточно сложном ПО все ошибки могут не обнаруживаться даже после длительного тестирования, однако чем тщательнее ведется тестирование, тем меньше ошибок остается и тем менее вероятно возникновение невыявленных ошибок.

#### Тестовый план

Тестирование обычно проводится снизу вверх, т.е. сначала тести-руются отдельные функции, затем целые модули и далее проводится комплексное тестирование всей программы или комплекса программ. Для проведения тестирования разрабатывается тест-план (test-plan) - совокупность тестовых наборов {примеров} (test-case). В каждом тестовом примере производится выполнение тестируемого программного элемента SUT при заданных Input - условиях и входных данных и проверяются все Output - выходные данные на соответствия заданным значениям.

Тестовый пример (набор) должен включать в себя как минимум:

* входы (конкретные значения всех выходных параметров, все необходимые свойства и установки окружения);
* действия (что надо выполнить и в какой последовательности);
* ожидаемый выход (конкретные величины всех возвращаемых значений, все выводы в потоки, сигналы, все изменяемые свойства и установки окружения).

Кроме указанных данных удобно, если каждый тестовый пример имеет дополнительно:

* номер (уникальный номер каждого тестового примера, чтобы на него можно было ссылаться);
* ссылку на требование (если для тестирования используются требования, то указание ссылок на конкретные требования, которые проверяет данный тестовый пример, упростит локализацию ошибок и обеспечит возможность проверки полноты тестирования);
* краткое описание (что проверяет данный тестовый пример).

Для проведения тестирования разрабатывается программа-драйвер (тест), выполняющая все тестовые примеры и сравнивающая выходные значения с ожидаемыми. В результате выполнения теста получается не только общий результат - есть или нет ошибки, но еще и список пройденных и непройденных тестовых примеров, который помогает локализовать ошибки в SUT.

Для упрощения локализации ошибок и последующей модификации тест-плана нужно, чтобы тестовые примеры были независимы друг от друга, т.е. чтобы каждый последующий тестовый пример никак не использовал результаты работы предыдущего. Для этого необходимо провести установки всех начальных условий перед выполнением каждого тестового примера.

#### Проблема полноты тестирования

Основная проблема тестирования ПО заключается в том, что проверить программу при всех возможных условиях функционирования в большинстве случаев невозможно. Это происходит либо в силу ограниченности ресурсов, либо в силу бесконечного количества возможных условий. Например, если рассмотреть функцию умножения двух рациональных чисел, варьируемых от -1000 до +1000, то в интервале от минимального возможного числа до максимального содержится бесконечное количество чисел. Т.е. все возможные значения входов проверить нельзя. Если же учесть, что машина оперирует невсеми этими числами, а различает только 10 знаков после запятой (т.е. множество чисел в интервале дискретно, минимальное отличие двух чисел 0,0000000001), то для проверки всех комбинаций из заданного диапазона понадобится https://webnvpks.github.io/files/vnedrenie_i_podderzhka_kompyuternyh_sistem/lectures/osnovnye_metody_vnedreniya_i_analiza_funkcionirovaniya_programmnogo_obespecheniya7_pic_02.png степени тестовых примера, что является достаточно большим числом для такой простой функции. Если проверяются не все возможные комбинации входных условий, то тестирование является неполным.

В основном для сложных программ тестирование является неполным, но даже неполное тестирование может выявить большинство ошибок, если выработать грамотную стратегию их поиска. Часто используют метод деления входных значений на области эквивалентности, так чтобы внутри каждой области для всех значений программа "вела себя" похоже. Тогда при написании тестовых примеров рассматриваются все значения на границах областей и по одному произвольному значению из каждой области (области определяются для каждого входного параметра).

Этот подход называют методом трех точек. В нашем примере для функции умножения двух чисел можно рассмотреть области [-1000; 0] и [0; +1000]. Деление образовано путем выявления трех особых точек (-1000, 0 и +1000). Такие точки называют критическими точками, в них тестируемая функция может менять свое поведение или потенциально вести себя особо. Т.е. для тестирования функции методом трех точек достаточно проверить https://webnvpks.github.io/files/vnedrenie_i_podderzhka_kompyuternyh_sistem/lectures/osnovnye_metody_vnedreniya_i_analiza_funkcionirovaniya_programmnogo_obespecheniya7_pic_03.png случаев (для каждого входа это точки -1000; 0; 1000 и, например, -500 и 500), что значительно меньше полного перебора. Конечно, при таком подходе возможно, что какие-то ошибки останутся, но вероятность этого будет невелика и зависит от выбора критических точек.

Функции, выполняющие различные сравнения, могут неверно их проводить, поэтому имеет смысл проверять их работу в непосредственной близости к критическим точкам. Для этого берутся значения, отстоящие от критических точек на величину дискретизации значений. Т.е. для примера функции умножения двух чисел, кроме значений метода трех точек, стоит рассмотреть значения -999,9999999999; -0,0000000001; 0,0000000001 и 999,9999999999. Этот подход называют методом пяти точек.

Иная сторона тестирования связана с типизацией переменных, при помощи которых задаются входные данные. Если для входных значений функции используются переменные типа float, а максимальное значение входа ограничено как +1000, то теоретически можно передать на вход и число +1001. Зачастую реакция функции на такое число не будет даже описана. Однако существуют приложения, чье поведение критично даже при передаче им входных значений, выходящих за пределы допустимых (например, авиационные программы, программы управления ядерными реакторами). В этом случае подразумевается, что программа должна вести себя корректно, т.е. не "зависнуть", не "повесить" систему, хотя выходное значение предсказать нельзя. Тестовые примеры, проверяющие поведение программы, в таких случаях, называются тестами на устойчивость (robustness). Если при тестировании методом пяти точек проверять еще и значения, выходящие за пределы допустимых диапазонов, то такой метод будет называться методом семи точек. В примере функции умножения двух чисел кроме значений -1000; -500; -999,9999999999; -0,0000000001; 0,0; 0,0000000001; 500,0; 999,9999999999; 1000 для каждого входа следует взять, например, еще значения -1001 и 100,0000000001.

Как уже было сказано, для тестирования ПО необходимо обладать информацией о том, что оно должно делать. Это может быть либо подробное описание (требования), либо просто сам код программы (в этом случае подразумевается, что программа должна работать корректно, не "портить память", не завершаться аварийно, не мешать другим процессам). В зависимости от исходной информации о ПО различают два подхода к тестированию - тестирование по требованиям и тестирование по коду.

#### Тестирование. Метод "черного ящика"

Тестирование по требованиям или тестирование "черного ящика" подразумевает неиспользование сведений о структуре исходного кода. Все тестовые примеры составляются только на основе требований, т.е. мы не видим, что находится внутри SUT и как оно работает. Единственное, что доступно - это то, что SUT должно делать, внешние эффекты поведения программного обеспечения.

При тестировании "черного ящика" удобно использовать методы трех, пяти или семи точек. При тестировании по требованиям используется понятие "покрытие" требований тестами. Тест покрывает требования, если он полностью проверяет выполнение каждого отдельного требования. При этом возможны как случаи, когда для проверки одного требования необходимо несколько тестовых примеров, так и случаи, когда достаточно одного тестового примера для проверки нескольких требований.

Рассмотрим, например, следующее требование к функции умножения двух целых чисел.

Если входное значение хотя бы одного множителя выходит за гра-ницы диапазона [15 ... 1500], то функция должна вернуть значение 0, в противном случае функция должна вернуть значение произведения двух множителей.

Для тестирования этого требования (если быть точнее, то двух требований) необходимо проверить значения множителей как из диа-пазона [15 ... 1500], так и вне его.

Тесты для проверки значений из диапазона могут выглядеть, например, следующим образом.

В табл. 1 представлены все комбинации двух входов, каждый из которых принимает по три различных значения (метод трех точек), так как диапазон [15 ... 1500] с точки зрения сложения чисел не имеет критических точек, кроме своих концов.

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица 1. Часть тест-плана | | | |
| Входы | Действия | Ожидаемый выход |  |
| Множитель 1 | Множитель 2 |  |  |
| 15 | 15 | Вызов функции | 225 |
| 15 | 700 | Вызов функции | 10500 |
| 15 | 1500 | Вызов функции | 22500 |
| 700 | 15 | Вызов функции | 10500 |
| 700 | 700 | Вызов функции | 490000 |
| 700 | 1500 | Вызов функции | 1050000 |
| 1500 | 15 | Вызов функции | 22500 |
| 1500 | 700 | Вызов функции | 1050000 |
| 1500 | 1500 | Вызов функции | 2250000 |
| 14 | 14 | Вызов функции | 0 |
| 1501 | 14 | Вызов функции | 0 |
| 14 | 1501 | Вызов функции | 0 |
| 14 | 700 | Вызов функции | 0 |
| 700 | 14 | Вызов функции | 0 |
| 1501 | 700 | Вызов функции | 0 |
| 700 | 1501 | Вызов функции | 0 |

Для проверки поведения функции за границами диапазона стоит проверить, например, значения 14 и 1501 для каждого входа.

Необходимо отметить, что не всегда можно покрыть все требования по тем же причинам, по которым невозможно добиться полноты тестирования, поэтому не всегда возможно доказать правильность работы программы. Кроме того, некоторые требования могут быть так сформулированы, что их нельзя протестировать. Например, требования могут касаться алгоритмов работы (а алгоритмы не видны при тестировании "черного ящика") или каких-нибудь внутренних переменных, не доступных извне. Некоторые требования вообще могут оказаться некорректными при тестировании, хотя могли быть интуитивно понятны при разработке, например требование "программа должна иметь дружественный интерфейс" протестировать невозможно. Для исключения таких ситуаций необходимо уже на этапе составления требований формулировать их как можно более конкретно и однозначно.

Метод тестирования "черного ящика" выявляет все несоответствия между требованиями к ПО и поведением самого ПО.

#### Тестирование. Метод "белого ящика"

Тестирование по коду или тестирование "белого ящика" основывается на проверке кода SUT, когда в ходе выполнения SUT проверяется выполнение каждого блока кода. При тестировании "белого ящика" основная задача - это выполнение всего кода для проверки работоспособности всех его ветвей. При этом ставится задача покрытия кода тестами и рассматриваются разные уровни покрытия. Один из уровней покрытия (покрытие операторов) - это выполнение всех операторов, т.е. при выполнении всех тестовых примеров в итоге должны выполниться все операторы (не в каждом тестовом примере, а по результатам выполнения всех тестовых примеров).

Например, для написанного ниже фрагмента программы, где A, B и C рассматриваются как входные значения:

X = 0;

if ((A<=B) || (A>C)) X = 5;

достаточно одного тестового примера (ТП1: A=1, B=2, C=3). В этом случае выполнятся все операторы. Но если программист допустил ошибку и неверно написал условие, например так:

if ((A<=B) || (A>B)) X = 5;

то тогда код будет работать неверно (переменной X всегда будет присваиваться значение 5), хотя показанный выше тестовый пример приведет к выполнению всех операторов и не выявит ошибки.

Для выявления таких ошибок требуется выполнить другой уровень покрытия - по условиям.

Покрытие по условиям требует проверок всех условий на TRUE/FALSE, т.е. каждое условие в ходе тестирования должно проверяться на оба возможных значения. Для покрытия по условиям приведенного примера кода необходимо уже два тестовых примера:

ТП1: A=1, B=2, C=3;

ТП2: A=3, B=2, C=3.

Эти тестовые примеры позволяют найти ошибку.

Не все блоки кода всегда удается покрыть тестами. Это может быть связано с защитным программированием (когда входные значения функции проверяются на корректность, но передаются ей только корректные данные, так как передающая функция тоже проверяет их корректность); операторами выхода (закрывающая скобка "}" после оператора exit); мертвым кодом (код, который заведомо никогда не выполняется).

Проверка таких блоков кода и анализ их "безопасности" может происходить без выполнения самого кода, в этом случае группа экспертов читает и анализирует программный код, делая выводы о существовании или несуществовании ошибок. Результаты подобного анализа могут установить причины появления непокрытого кода.

Одной из причин может быть несовершенство тест-плана. Другой - пробелы в требованиях. И в том и в другом случае сам тестируемый код может не требовать изменений, но должны быть дописаны новые требования или изменены уже существующие и/или расширен тест-план. В обоих случаях процесс верификации повторяется.

#### Заглушки

Часто возникает необходимость тестировать модули, использующие процедуры других модулей, которые могут влиять на результат тестирования или значительно усложнять его получение. Допустим, имеется функция climatControl, обрабатывающая информацию с датчика температуры и управляющая нагревателем. Значение температуры она получает при помощи функции getTemperature, которая в свою очередь опрашивает температурный датчик. Пусть имеется следующее требование: "Если температура становится ниже 23 °C, то функция climatControl должна включить нагреватель". Для тестирования этого требования необходимо передать функции некое значение температуры, меньшее 23, допустим 20. Но эта величина не является входным значением функции climatControl, она задается возвращаемым значением функции getTemperature. Один из путей - "заставить" функцию getTemperature возвратить нужное значение, например подключив реальный датчик температуры и охладив его до 20°С. Но при этом нет гарантии, что датчик и сама функция getTemperature работает правильно. А что если она возвращает значение температуры в фаренгейтах (20°С равно 68°F, т.е. функция вернет 68 вместо ожидаемых 20)? Это приведет к тому, что функция climatControl не включит нагреватель, даже если не содержит в себе ошибок.

В таких ситуациях реальная функция getTemperature заменяется новой функцией, которая разрабатывается специально для конкретного (конкретных) тестового примера, и выполняет только то, что необходимо для теста, возвращая нужное значение. Т.е. в данном примере функция getTemperature заменяется на функцию, которая не будет опрашивать датчик, а сразу вернет значение 20. Функции или программы, которыми заменяются используемые при тестировании SUT функции и программы, называют заглушками.

Стоит отметить, что если SUT использует какие-либо функции, которые не заменяются заглушками в ходе тестирования, то такие функции также должны быть предварительно протестированы. Это приводит к так называемому "восходящему" тестированию. Что, в свою очередь, предполагает такую же последовательность реализации модулей системы. Чаще как раз подобные getTemperature процедуры реализуются в последний момент, так как зависят от периферийного оборудования, которое не изготавливается, а закупается или характеристики которого специально подгоняются под условия эксплуатации системы. В этих случаях написание моделей (заглушек) программ неизбежно.

#### Процесс тестирования

Как уже говорилось, тест состоит из набора тестовых примеров. Перед написанием теста необходимо подготовить сценарий каждого тестового примера. Такие сценарии описываются в тест-плане. Тест-план представляет собой документ, последовательно определяющий всю совокупность тест-примеров с указанием конкретных значений всех входных данных, действий, а также ожидаемых значений выход-ных данных (см. выше про то, что включает в себя тестовый пример).

В тест-плане отражается основная логика тестирования, а сам тест (набор тестовых примеров) является реализацией тест-плана. Вполне возможно построить универсальный тест-драйвер, который будет использовать в качестве входных данных тест-план, создавать на его основе все необходимые условия для каждого тестового примера и выполнять заданные сценарии, сравнивая реальные выходные значения с ожидаемыми. Этот подход часто применяется для автоматизации процесса тестирования.

Если в тест-плане для каждого тестового примера указываются ссылки на тестируемые им требования, то можно говорить о трассировке или о соответствии тестового примера тест-плана требованию. Аналогично при кодировании теста для каждого тест-примера удобно указывать ссылку на номер его описания в тест-плане.

Такая трассировка позволит при выявлении ошибки каким-либо тестовым примером определить, перейдя к тест-плану, при каких условиях проявилась ошибка, и далее, перейдя к требованиям, выяснить, какие требования нарушены. Так как в ходе разработки документация может меняться, то при трассировке необходимо сохранять соответствие версий трассируемых документов. Это, в свою очередь, позво-ляет определять область влияния внесенных изменений, например определить, какие тесты должны быть модифицированы или перепроверены при изменении части требований к системе.

Результатом выполнения теста является отчет о прогоне теста, в котором отражается, сколько всего было выполнено тестовых примеров, сколько ошибок найдено, какие тестовые примеры выявили ошибки. Удобно помещать в отчет о прогоне теста ожидаемые и ре-альные значения всех выходных данных после выполнения каждого тестового примера, чтобы в случае выявления ошибки можно было их сравнить.

#### Пример верификации

##### Третий этап работы

Тестирование программной реализации

**Соображения**

Общие требования к тест-плану состоят в определении входных воздействий на программу достаточных для контроля всех требований по проверке функций программной реализации как "черного ящика" (полнота покрытия требований). Дополнительно желательно определить входные воздействия, обеспечивающие выполнение всех ветвей программной реализации (программа проверяется как "белый ящик"). Часто удобно представлять тест-план в виде таблицы.

1. Охарактеризуйте процесс создания эксплуатационной документации при внедрении и анализе функционирования программного обеспечения.

К эксплуатационным документам относят руководоство по эксплуатации и техничесическое обслуживание, паспорт.

РЭ должна содержать подробное изложение правил пользования с указа­нием последовательности действия оператора при включении средства измерения (СИ), его подготовке к работе (настройка, наладка, калибровка и т. д.), в процессе измерений, о за­мене сменных элементов, подстройке.

Для проведения ремонта СИ в инструкции по эксплуатации (ИЭ) целесообразно привести монтажные карты, принципиальные схемы со спецификацией, таблицу возможных неисправностей и способы их устранения; карты режимов полупроводнико­вых приборов или электронных ламп (если такие элементы имеются); намоточные данные дросселей, трансформаторов и проволочных резисторов, кинематические схемы, отдельные рекомендации по ремонту.

Содержание руководства по эксплуатации (РЭ) регламентируется ГОСТ 2.601-2006

ГОСТ 2.610-2006 содержит следующую типовую структуру руководства по эксплуатации.

1. Введение.

2. Описание и работа.

- Описание и работа изделия.

- Назначение изделия.

- Характеристики (свойства).

- Состав изделия.

- Устройство и работа.

- Средства измерения, инструмент и принадлежности.

- Маркировка и пломбирование.

- Упаковка.

- Описание и работа составных частей изделия.

3. Использование по назначению.

- Эксплуатационные ограничения.

- Подготовка изделия к использованию.

- Меры безопасности при подготовке изделия.

- Правила и порядок заправки изделия ГСМ.

- Объем и последовательность внешнего осмотра изделия.

- Правила и порядок осмотра рабочих мест.

- Правила и порядок осмотра и проверки готовности изделия к использованию.

- Описание положений органов управления и настройки после подготовки изделия к работе и перед включением.

- Указания об ориентировании изделия.

- Особенности подготовки изделия к использованию из различных степеней готовности.

- Указания о взаимосвязи (соединении) данного изделия с другими изделиями.

- Указания по включению и опробованию работы изделия.

- Перечень возможных неисправностей изделия в процессе его подготовки и рекомендации по действиям при их возникновении.

- Использование изделия.

- Порядок действия обслуживающего персонала при выполнении задач применения изделия.

- Порядок контроля работоспособности изделия.

- Перечень возможных неисправностей в процессе использования изделия по назначению и рекомендации по действиям при их возникновении.

- Перечень режимов работы изделия, а также характеристики основных режимов работы.

- Порядок и правила перевода изделия с одного режима работы на другой с указанием необходимого для этого времени.

- Порядок приведения изделия в исходное положение.

- Порядок выключения изделия, содержание и последовательность осмотра изделия после окончания работы.

- Порядок замены, пополнения и контроля качества ГСМ.

- Меры безопасности при использовании изделия по назначению.

- Действия в экстремальных условиях.

- Особенности использования доработанного изделия.

4. Техническое обслуживание.

- Техническое обслуживание изделия.

- Общие указания.

- Меры безопасности.

- Порядок технического обслуживания изделия.

- Проверка работоспособности изделия.

- Техническое освидетельствование.

- Консервация (расконсервация, переконсервация).

- Техническое обслуживание составных частей изделия.

5. Текущий ремонт.

6. Хранение.

7. Транспортирование.

8. Утилизация.

1. **Введение**излагают без заголовка. Оно содержит:

* назначение и состав РЭ;
* требуемый уровень специальной подготовки обслуживающего персонала;
* распространение РЭ на модификации изделия;
* другие сведения (при необходимости).
* Для изделий, которые при определенных условиях могут представлять опасность для жизни и здоровья человека, во введении должна быть приведена информация о видах опасных воздействий

2. Часть «**Описание и работа**» состоит из разделов:

* описание и работа изделия;
* описание и работа составных частей изделия.

Раздел «**Описание и работа изделия**» содержит:

* назначение изделия;
* технические характеристики (свойства);
* состав изделия;
* устройство и работа;
* средства измерения, инструмент и принадлежности;
* маркировка и пломбирование;
* упаковка.

Подраздел «Назначение изделия» содержит наименование изделия, его обозначение, назначение, область применения, параметры, размеры, характеризующие условия эксплуатации

Подраздел «Технические характеристики» содержит технические данные, основные параметры и характеристики (свойства), необходимые для изучения и правильнойтехнической эксплуатации изделия. При изложении сведений о контролируемых (измеряемых) параметрах необходимо указывать: наименование параметра; номинальное значение, допуск (доверительный интервал); применяемое средство измерения

Подраздел «Состав изделия» содержит наименования, обозначения и места расположения основных составных частей изделия и установленных для изделиякомплектов ЗИП. Здесь же указывают общие отличия в конструкции различныхмодификаций изделий от базового изделия и друг от друга и особенности их комплектации. Допускается приводить схему деления изделия на составные части

Подраздел «Устройство и работа» содержит общие сведения о принципе действия, устройстве и режимах работы изделия в целом, взаимодействии составных частей изделия. Здесь же указывают, при необходимости, взаимодействие данного изделия с другими изделиями.

Подраздел «Средства измерения, инструмент и принадлежности» содержит назначение, перечень, места расположения и краткие основные технические (в том числе метрологические) характеристики, а также устройство и принцип действия специальных средств измерения, испытательного и другого оборудования, инструмента и принадлежностей, которые необходимы для контроля, регулирования (настройки), выполнения работ по техническому обслуживанию и текущему ремонту изделия и его составных частей.

Подраздел «Маркировка и пломбирование» содержит сведения для всего изделия в целом о маркировании и пломбировании изделия, тары и упаковочных материалов.

Подраздел «Упаковка» содержит для всего изделия в целом описание конструкции и порядка использования тары, упаковочных материалов и т. п., порядок пломбирования и распломбирования.

Раздел «Описание и работа составных частей изделия» содержит общие сведения о составных частях изделия и состоит из подразделов:

* общие сведения;
* описание;
* работа;
* маркировка и пломбирование;
* упаковка.

Подраздел «Общие сведения» содержит в общем виде назначение и описание составных частей изделия, из каких основных составных частей более мелкого уровня деления состоит описываемая составная часть изделия, где они расположены, какие выполняют функции, их взаимосвязь и др.

Подраздел «Работа» содержит описание работы составных частей изделия.

Содержание подразделов «Маркировка и пломбирование» и «Упаковка» составных частей изделия аналогично содержанию подразделов для изделия в целом.

3. Часть «**Использование по назначению**» состоит из разделов:

* эксплуатационные ограничения;
* подготовка изделия к использованию;
* использование изделия;
* действия в экстремальных условиях;
* особенности использования доработанного изделия.

Раздел «Эксплуатационные ограничения» содержит те технические характеристикиизделия, несоблюдение которых недопустимо по условиям безопасности и которые могут привести к выходу изделия из строя. Эти характеристики, с указанием их количественных значений, рекомендуется излагать в виде таблиц в порядке, соответствующем последовательности этапа использования изделия по назначению.

Все ограничения, помещаемые в данном разделе, должны обеспечивать возможность их контроля обслуживающим персоналом

Раздел «Подготовка изделия к использованию» содержит указания по проверке и приведению изделия к использованию по назначению.

Раздел, как правило, содержит подразделы:

* меры безопасности при подготовке изделия;
* правила и порядок заправки изделия топливом, маслами, смазками, газами, жидкостями и другими материалами (далее - ГСМ) с указанием их количества и марки, а также условия и порядок заправки дублирующими (резервными) ГСМ и, при необходимости, зарубежными ГСМ;
* объем и последовательность внешнего осмотра изделия;
* правила и порядок осмотра рабочих мест;
* правила и порядок осмотра и проверки готовности изделия к использованию;
* описание положений органов управления и настройки после подготовки изделия к работе и перед включением;
* указания об ориентировании изделия (с приложением схем при необходимости);
* особенности подготовки изделия к использованию из различных степеней готовности;
* при необходимости, указания о взаимосвязи (соединении) данного изделия с другими изделиями;
* указания по включению и опробованию работы изделия с описанием операций по проверке изделия в работе, в том числе с помощью средств измерения, входящих в состав изделия (приводятся значения показаний средств измерений, соответствующие установленным режимам работы, и допустимые отклонения от этих значений);
* перечень возможных неисправностей изделия в процессе его подготовки и рекомендации по действиям при их возникновении.

Раздел «Использование изделия» содержит, как правило, подразделы:

* порядок действия обслуживающего персонала при выполнении задач применения изделия;
* порядок контроля работоспособности изделия в целом с описанием методик выполнения измерений, регулирования (настройки), наладки изделия, а также схем соединения изделия со средствами измерений и вспомогательными устройствами, используемых для измерений;
* перечень возможных неисправностей в процессе использования изделия по назначению и рекомендации по действиям при их возникновении;
* перечень режимов работы изделия, а также характеристики основных режимов работы;
* порядок и правила перевода изделия с одного режима работы на другой с указанием необходимого для этого времени;
* порядок приведения изделия в исходное положение;
* порядок выключения изделия, содержание и последовательность осмотра изделия после окончания работы;
* порядок замены, пополнения и контроля качества (при необходимости) ГСМ;
* меры безопасности при использовании изделия по назначению. При этом должны быть отражены требования, обеспечивающие безопасность обслуживающего персонала, техники и экологическая безопасность проводимых работ.

Раздел «Действия в экстремальных условиях» содержит случаи отказа изделия в экстремальных условиях и условия, которые могут привести к аварийной ситуации. Раздел содержит, как правило, действия в следующих случаях:

* при пожаре на изделии на различных этапах использования изделия;
* при отказах систем изделия, способных привести к возникновению опасных аварийных ситуаций;
* при попадании в аварийные условия эксплуатации;
* при экстренной эвакуации обслуживающего персонала.

Раздел «Особенности использования доработанного изделия» содержит:

* основные конструктивные отличия данного изделия от базового изделия и обусловленные ими изменения в эксплуатационных ограничениях и рекомендациях по эксплуатации;
* особенности выполнения операций на всех этапах подготовки и использования по назначению модифицированного изделия.

Допускается эти особенности приводить в тексте РЭ, не выделяя в отдельный раздел

4. Часть «**Техническое обслуживание**» содержит сведения по техническому обслуживанию(ТО) изделия и его составных частей и состоит из разделов:

* техническое обслуживание изделия;
* техническое обслуживание составных частей изделия. Изделие и его составные части, на которых проводят работы по техническому обслуживанию (далее - объекты ТО), виды и объемы работ и периодичность их выполнения зависят от уровня надежности объектов ТО при условии оптимальных сроков проведения ТО и расходов материальных средств и трудовых ресурсов на ТО.

Раздел «**Техническое обслуживание изделия**» состоит из подразделов:

* общие указания;
* меры безопасности;
* порядок технического обслуживания изделия;
* проверка работоспособности изделия;
* техническое освидетельствование;
* консервация (расконсервация, переконсервация).

Подраздел «**Общие указания**» содержит:

* характеристику принятой системы ТО: виды, объемы и периодичность ТО, особенности организации ТО изделия и его составных частей в зависимости от этапов его эксплуатации (использование по назначению, хранение,транспортирование и т. д.) и условий эксплуатации (климатические, временные и т. д.), указания по организации ТО;
* требования к составу и квалификации обслуживающего персонала;
* требования к изделию, направляемому на ТО;

Подраздел «Меры безопасности» содержит правила, которые необходимо соблюдать в соответствии с особенностями конструкции изделия и его эксплуатации, действующими положениями нормативных документов, а также перечень обязательных требований по техническому обслуживанию и (или) ремонту, невыполнение которых может привести к опасным последствиям для жизни, здоровья человека или окружающей среды. Здесь же излагают правила пожарной безопасности, взрывобезопасности и т. п.

Подраздел «Порядок технического обслуживания изделия» содержит характеристику каждого вида ТО изделия и его составных частей, в том числе замена смазки, заправка специальными жидкостями, кислородом и др., дренаж трубопроводов и агрегатов и т. д. в зависимости от особенностей и условий эксплуатации, периодичность видов ТО, в том числе и при хранении, сведения по всем видам ТО, принятым для эксплуатируемого изделия.

Содержание подраздела рекомендуется излагать в виде таблицы 1.

Таблица 1 - Порядок технического обслуживания

|  |  |  |  |
| --- | --- | --- | --- |
| Пункт РЭ | Наименование объекта ТО и работы | Виды ТО | Примечание |
|  |  |  |  |

В графе «Пункт РЭ» указывают порядковый номер пункта (работы), под ним номер раздела, подраздела, пункта РЭ.

В графе «Наименование объекта ТО и работа» приводят наименование объекта ТО и перечень работ, проводимых при ТО.

В графе «Виды ТО» приводят условное обозначение вида ТО или периода выполнения видов ТО, а также условное обозначение выполняемой («+») или невыполняемой («-») работы. Графа может состоять из одной или нескольких колонок

Подраздел «Проверка работоспособности изделия» содержит последовательность выполнения работ по проверке работоспособности изделия.

Содержание подраздела рекомендуется излагать в виде таблицы 2.

Таблица 2 - Проверка работоспособности

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование работы | Кто выполняет | Средства измерений вспомогательные технические устройства и материалы | Контрольные значения параметров |
|  |  |  |  |

В графе «Наименование работы» приводят наименование выполняемой работы в последовательности их выполнения.

В графе «Кто выполняет» указывают в сокращенном виде, кто выполняет работу, например М - механик, О - оператор и т. д.

В графе «Средства измерений, вспомогательные технические устройства и материалы» указывают измерительные и вспомогательные устройства, а также материалы, не входящие в изделие, но которые необходимо использовать.

В графе «Контрольные значения параметров» указывают значения, в пределах которых должны находиться параметры, контролируемые при проверке исправности изделия, и значения параметров, при которых изделие отправляют в ремонт. При изложении сведений о контролируемых (измеряемых) параметрах необходимо указывать: наименование параметра; номинальное значение; допуск (доверительный интервал); применяемое средство измерения.

В подразделе также приводят указания о порядке проведения пред ремонтной дефектации изделия с целью оценки его технического состояния и определения необходимости отправки изделия в капитальный (средний) ремонт.

Подраздел «Техническое освидетельствование» содержит порядок и периодичность освидетельствования изделия (и) или его составных частей органами инспекции и надзора, а также указывают, в каком месте формуляра или паспорта приведен перечень поверяемых средств измерения, освидетельствованных сосудов, работающих под высоким давлением, грузоподъемных средств, входящих в изделие и его комплекты. Здесь же указывают требования по подготовке средств измерений к поверке и методики поверки встроенных средств измерений без демонтажа их с изделия

Подраздел «Консервация (расконсервация, переконсервация)» содержит сведения о средствах и методах наружной и внутренней консервации, расконсервации, переконсервации (далее - консервации) изделия в целом, периодичности консервации при хранении, порядок приведения изделия в состояние готовности к использованию по назначению из состояния консервации, перечень используемых инструментов, приспособлений и материалов

Раздел «Техническое обслуживание составных частей изделия», как правило, содержит подразделы:

* обслуживание;
* демонтаж и монтаж;
* регулирование и испытание;
* осмотр и проверка;
* очистка и окраска;
* консервация.

Подраздел «Обслуживание» содержит правила и порядок обслуживания

Подраздел «Демонтаж и монтаж» содержит порядок работ по демонтажу и монтажу, перечень приспособлений и инструментов, необходимых для отсоединения, снятия, обратной установки и присоединения сборочных единиц (деталей), меры предосторожности, перечень регулировочных работ после монтажа. Указание «Установку проводить в обратной последовательности» приводить не разрешается.

Подраздел «Регулирование и испытание» содержит порядок работ, необходимых для регулирования (настройки) составной части изделия для получения требуемых технических характеристик и параметров

Подраздел «Осмотр и проверка» содержит порядок работ, необходимых для осуществления доступа к осматриваемой части изделия; виды и методы ее осмотра и проверки; порядок работ, необходимых для проведения технического освидетельствования составных частей изделия органами инспекции и надзора, а также оценки технического состояния составных частей изделия при определении необходимости отправки их в ремонт

Подраздел «Очистка и окраска» содержит порядок работ по очистке и подкраске составных частей изделия, условий их выполнения и перечень используемых инструментов, приспособлений и материалов

5. Часть «**Текущий ремонт**» содержит сведения, необходимые для организации и проведения текущего ремонта изделия и его составных частей в условиях эксплуатации, состоит из разделов:

* текущий ремонт изделия;
* текущий ремонт составных частей изделия.

Раздел «Текущий ремонт изделия» содержит подразделы:

* общие указания;
* меры безопасности.

Подраздел «Общие указания» содержит требования по проведению ремонта, методы ремонта, требования к квалификации персонала, описание и характеристики диагностических возможностей систем встроенного контроля, а также переченьсоставных частей изделия, текущий ремонт которых может быть осуществлен только в условиях ремонтных органов и описание и характеристики диагностических возможностей внешних средств диагностирования. При необходимости приводят схемы поиска последствий отказов и повреждений.

Подраздел «Меры безопасности» содержит правила предосторожности, которые в соответствии с действующими нормативами должны быть соблюдены при проведении работ.

Раздел «Текущий ремонт составных частей изделия» содержит указания по поиску и устранению последствий отказов и повреждений и применительно к каждой составной части изделия, текущий ремонт которых возможен при эксплуатации, состоит из подразделов:

* поиск последствий отказов и повреждений;
* устранение последствий отказов и повреждений.

Подраздел «Поиск последствий отказов и повреждений» содержит указания по последовательности и объему работ, необходимых для отыскания последствий отказов и повреждений

Подраздел «Устранение последствий отказов и повреждений» содержит указания о методах устранения последствий отказов и повреждений, а также перечень необходимых для этого средств измерения, инструмента и приспособлений. Подраздел рекомендуется оформлять в виде карты (см. приложение А).

Раздел «Текущий ремонт составных частей изделия» допускается на подразделы не разделять, а сведения излагать в виде таблицы 3.

Таблица 3 - Текущий ремонт

|  |  |  |  |
| --- | --- | --- | --- |
| Описание последствий отказов и повреждений | Возможные причины | Указания по установлению последствий отказов и повреждений сборочной единицы (детали) | Указания по устранению последствий отказов и повреждений |
|  |  |  |  |

В графе «Описание последствий отказов и повреждений» приводят описание последствий отказов и повреждений, записанных в порядке вероятности их появления, и, при необходимости, указывают внешние проявления и другие дополнительные признаки последствий отказов и повреждений.

В графе «Возможные причины» указывают, какая из составных частей изделия может отказать и быть повреждена и возможные причины отказов и повреждений. Причины отказов и повреждений перечисляют в порядке вероятности появления.

В графе «Указания по установлению последствий отказов и повреждений сборочной единицы (детали)» приводят последовательность действий и другие указания, необходимые для установления (отыскания) последствий отказов и повреждений сборочной единицы (детали).

В графе «Указания по устранению последствий отказов и повреждений» перечисляют указания по устранению последствий отказов и повреждений или приводят ссылки на другие документы, по которым проводят работы по их устранению.

При необходимости перечень наиболее вероятных последствий отказов и повреждений может быть выделен в самостоятельную таблицу

6. Часть «**Хранение**» содержит:

* правила постановки изделия на хранение и снятия его с хранения;
* перечень составных частей изделия с ограниченными сроками хранения;
* перечень работ, правила их проведения, меры безопасности при подготовке изделия к хранению, при кратковременном и длительном хранении изделия, при снятии изделия с хранения;
* условия хранения изделия (вид хранилищ, температура, влажность, освещенность и т. п.) для определенных сроков хранения;
* способы утилизации (если изделие представляет опасность для жизни, здоровья людей или окружающей среды после окончания срока эксплуатации;
* предельные сроки хранения в различных климатических условиях.

7. Часть «**Транспортирование**» содержит:

* требования к транспортированию изделия и условиям, при которых оно должно осуществляться;
* порядок подготовки изделия для транспортирования различными видами транспорта;
* способы крепления изделия для транспортирования его различными видами транспорта с приведением необходимых схем крепления;
* порядок погрузки и выгрузки изделия и меры предосторожности.

Одновременно в разделе приводят транспортные характеристики изделия (масса, габаритные размеры, положение центра тяжести и т. п.), а также схему изделия применительно к расположению его на транспортном средстве с указанием основных размеров изделия. При необходимости указывают сведения по буксированию изделия и эвакуации [из п. 5.1.8 ГОСТ 2.601-95]

8. Часть «**Утилизация**» содержит:

* меры безопасности;
* сведения и проводимые мероприятия по подготовке и отправке изделия наутилизацию;
* перечень утилизируемых составных частей (расчетный);
* перечень утилизируемых составных частей, выявляемых по результатамтекущего ремонта, технического обслуживания и хранения (при необходимости);
* методы утилизации, если изделие представляет опасность для жизни, здоровья людей и окружающей среды после окончания срока службы (эксплуатации).

1. Дать определения понятиям: совместимость программного обеспечения, аппаратная и программная совместимость, совместимость драйверов. Опишите причины возникновения проблем совместимости.

**Аппаратное обеспечение.**К аппаратному обеспечению вычислительных систем относятся устройства и приборы, образующие аппаратную конфигурацию. Современные компьютеры и вычислительные комплексы имеют блочно-модульный конструктив — аппаратную конфигурацию, которую можно собирать из готовых узлов и блоков в зависимости от конкретных видов работ.

По способу расположения устройств относительно центрального процессорного устройства (ЦПУ, Central Processing Unit — CPU) различают внутренние и внешние устройства. Внешними, как правило, являются большинство устройств ввода-вывода данных (их также называют периферийными устройствами) и некоторые устройства, предназначенные для длительного хранения данных.

Согласование между отдельными узлами и блоками выполняют с помощью переходных аппаратно-логических устройств, называемых аппаратными интерфейсами. Стандарты на аппаратные интерфейсы в вычислительной технике называют протоколами. Таким образом, протокол — это совокупность технических условий, которые необходимо обеспечить для успешного согласования функционирования устройств.

Многочисленные интерфейсы, присутствующие в архитектуре любой вычислительной системы, условно подразделяют на две большие группы: последовательные и параллельные. Через последовательный интерфейс данные передаются последовательно, бит за битом, а через параллельный — одновременно группами битов. Количество битов, участвующих в одной посылке, определяется разрядностью интерфейса, например восьмиразрядные параллельные интерфейсы передают 1 байт (8 бит) информации за один цикл.

Параллельные интерфейсы устроены сложнее, чем последовательные, но обеспечивают более высокую производительность. Их применяют там, где важна скорость передачи данных: для подключения печатающих устройств, устройств ввода графической информации, устройств записи данных на внешний носитель и т. п. Производительность параллельных интерфейсов измеряют байтами в секунду (байт/с; Кб/с; Мб/с).

Последовательные интерфейсы конструктивно проще параллельных; как правило, для них не надо синхронизировать работу передающего и принимающего устройств (поэтому их часто называют асинхронными интерфейсами), но пропускная способность их меньше и коэффициент полезного действия ниже, так как из-за отсутствия синхронизации посылок полезные данные предваряют и завершают посылками служебных данных, т. е. на 1 байт полезных данных приходятся 1—3 служебных бита (состав и структуру посылки определяет конкретный протокол).

Поскольку обмен данными через последовательные устройства производится не байтами, а битами их производительность измеряют битами в секунду (бит/с, Кбит/с, Мбит/с). Несмотря на кажущуюся простоту перевода единиц измерения скорости последовательной передачи в единицы измерения скорости параллельной передачи данных путем механического деления на восемь, такой пересчет не выполняют, поскольку он не корректен из-за наличия служебных данных. В крайнем случае с поправкой на служебные данные иногда скорость последовательных устройств выражают в знаках в секунду или в символах в секунду (с/с), но эта величина имеет не технический, а справочный, потребительский характер.

Последовательные интерфейсы применяют для подключения «медленных» устройств (простейших устройств печати низкого качества, устройств ввода-вывода знаковой и сигнальной информации, контрольных датчиков, малопроизводительных устройств связи и т. п.), а также в тех случаях, когда нет существенных ограничений по продолжительности обмена данными, к примеру, в большинстве цифровых фотокамер.

**Программное обеспечение.**Программа — это упорядоченная последовательность команд. Конечная цель любой компьютерной программы — управление аппаратными средствами. Даже если на первый взгляд программа не взаимодействует с оборудованием, не требует ввода данных с устройств ввода и не осуществляет вывод данных на устройства вывода, ее работа основана на управлении аппаратными устройствами компьютера.

Программное и аппаратное обеспечение в компьютере работают в неразрывной связи и в непрерывном взаимодействии. Несмотря на то, что эти две категории рассматриваются отдельно, нельзя забывать, что между ними существует диалектическая связь.

Состав программного обеспечения вычислительной системы называют программной конфигурацией. Между программами, как и между физическими узлами и блоками, существует взаимосвязь — многие программы работают, опираясь на другие программы более низкого уровня, т. е. можно говорить о так называемом межпрограммном интерфейсе. Возможность существования такого интерфейса тоже основана на соблюдении определенных технических условий и протоколов взаимодействия, что на практике обеспечивается распределением программного обеспечения на несколько взаимодействующих в виде пирамиды. Каждый следующий уровень опирается на программное обеспечение предшествующих уровней. Такое членение удобно для всех этапов работы с вычислительной системой, начиная с установки программ и заканчивая практической эксплуатацией и техническим обслуживанием. Следует отметить, что каждый вышележащий уровень повышает функциональность всей системы. Так, вычислительная система с программным обеспечением базового уровня не способна выполнять большинство функций, но позволяет установить системное программное обеспечение.

Базовый уровень — это самый низкий уровень программного обеспечения; он отвечает за взаимодействие с базовыми аппаратными средствами. Как правило, базовые программные средства непосредственно входят в состав базового оборудования и хранятся в специальных микросхемах — постоянных запоминающих устройствах (ПЗУ). Программы и данные записываются («прошиваются») в микросхемы ПЗУ на этапе производства и не могут быть изменены в процессе эксплуатации.

В тех случаях, когда изменение базовых программных средств во время эксплуатации является технически целесообразным, вместо микросхем ПЗУ применяют перепрограммируемые постоянные запоминающие устройства (ППЗУ). В этом случае изменение содержания ПЗУ можно выполнять как непосредственно в составе вычислительной системы (флэш-технология), так и вне ее, на специальных устройствах, называемых программаторами.

Системный уровень. Этот уровень — переходный; он обеспечивает взаимодействие программ базового уровня и непосредственно аппаратного обеспечения с другими программами компьютерной системы, т. е. выполняет «посреднические» функции.

От программного обеспечения этого уровня во многом зависят эксплуатационные показатели всей вычислительной системы в целом. Так, при подключении к вычислительной системе нового оборудования на системном уровне должна быть установлена программа, обеспечивающая взаимосвязь с этим оборудованием. Конкретные программы, отвечающие за взаимодействие с конкретными устройствами, называются драйверами устройств — они входят в состав программного обеспечения системного уровня.

Другой класс программ системного уровня отвечает за взаимодействие с пользователем. Эти программные средства называют средствами обеспечения пользовательского интерфейса. От них напрямую зависит удобство работы с компьютером и производительность труда на рабочем месте.

Совокупность программного обеспечения системного уровня образует ядро операционной системы компьютера. Понятие «операционная система» рассматривается далее, здесь следует отметить, что, если компьютер оснащен программным обеспечением системного уровня, то он уже подготовлен к установке программ более высоких уровней, к взаимодействию программных средств с оборудованием и, самое главное, к взаимодействию с пользователем. Иными словами, наличие ядра операционной системы — непременное условие для возможности практической работы человека с вычислительной системой.

Служебный уровень. Необходим для обеспечения взаимодействия как с программами базового уровня, так и с программами системного уровня. Основное назначение служебных программ (их также называют утилитами) состоит в автоматизации работ по проверке, наладке и настройке компьютерной системы. Во многих случаях они используются для расширения или улучшения функций системных программ. Некоторые служебные программы (как правило, это программы обслуживания) изначально включают в состав операционной системы, но большинство служебных программ являются для операционной системы внешними и предназначены для расширения ее функций.

В разработке и эксплуатации служебных программ существует два альтернативных направления: интеграция с операционной системой и автономное функционирование. В первом случае служебные программы способны изменять потребительские свойства системных программ, делая их более удобными для практической работы. Во втором случае они слабо связаны с системным программным обеспечением, но предоставляют пользователю больше возможностей для персональной настройки их взаимодействия с аппаратным и программным обеспечением.

Прикладной уровень. Программное обеспечение данного уровня представляет собой комплекс прикладных программ, с помощью которых в предметной области выполняется конкретная деятельность. Огромный функциональный диапазон возможных приложений средств вычислительной техники обусловлен наличием прикладных программ для разных видов деятельности — от производственной до творческой и развлекательно-обучающей.

Поскольку между прикладным программным обеспечением и системным существует непосредственная взаимосвязь (первое опирается на второе), то можно утверждать, что универсальность вычислительной системы, доступность прикладного программного обеспечения и широта функциональных возможностей компьютера напрямую зависят от типа используемой операционной системы, от того, какие системные средства содержит ее ядро, как она обеспечивает триединое взаимодействие: человек — программа — оборудование.

**Классификация прикладных программных средств.**Текстовые редакторы. Основные функции этого класса прикладных программ заключаются в вводе и редактировании текстовых данных. Дополнительные функции состоят в автоматизации процессов ввода и редактирования. Для операций ввода, вывода и сохранения данных текстовые редакторы вызывают и используют системное программное обеспечение.

С этого класса прикладных программ обычно начинают знакомство с программным обеспечением и на нем отрабатывают первичные навыки взаимодействия с компьютерной системой.

Текстовые процессоры. Отличие текстовых процессоров от текстовых редакторов состоит в том, что они позволяют не только вводить и редактировать текст, но и форматировать его, т. е. оформлять. Соответственно к основным средствам текстовых процессоров относятся средства обеспечения взаимодействия текста, графики, таблиц и других объектов, составляющих итоговый документ, а к дополнительным — средства автоматизации процесса форматирования.

Современный стиль работы с документами подразумевает два альтернативных подхода — работу с бумажными документами и работу с электронными документами (по безбумажной технологии), поэтому, говоря о форматировании документов средствами текстовых процессоров, надо иметь в виду два принципиально разных направления — форматирование документов, предназначенных для печати, и форматирование электронных документов, предназначенных для отображения на экране. Приемы и методы в этих случаях существенно различаются; соответственно различаются и текстовые процессоры, хотя многие из них успешно сочетают оба подхода.

Графические редакторы — широкий класс программ, предназначенных для создания и (или) обработки графических изображений. В данном классе различают следующие категории: растровые редакторы, векторные редакторы и программные средства для создания и обработки трехмерной графики (3D-редакторы).

Растровые редакторы применяют в тех случаях, когда графический объект представлен в виде комбинации точек, образующих растр и обладающих свойствами яркости и цвета. Такой подход эффективен в тех случаях, когда графическое изображение имеет много полутонов и информация о цвете элементов, составляющих объект, важнее, чем информация об их форме. Это характерно для фотографических и полиграфических изображений. Растровые редакторы широко применяются для обработки изображений, их ретуши, создания фотоэффектов и художественных композиций (коллажей).

Возможности создания изображений средствами растровых редакторов ограничены и не всегда удобны. В большинстве случаев художники предпочитают пользоваться традиционными инструментами, после чего вводить рисунок в компьютер с помощью сканеров и завершать работу путем применения спецэффектов растровым редактором.

Векторные редакторы отличаются от растровых способом представления данных об изображении. Элементарным объектом векторного изображения является не точка, а линия. Такой подход характерен для чертежно-графических работ, в которых форма линий имеет большее значение, чем информация о цвете отдельных точек, составляющих ее. В векторных редакторах каждая линия рассматривается как математическая кривая третьего порядка и представляется не комбинацией точек, а математической формулой (в компьютере хранятся числовые коэффициенты этой формулы). Такое представление намного компактнее, чем растровое, в результате данные занимают меньший объем, однако построение любого объекта выполняется не простым отображением точек на экране, а сопровождается непрерывным пересчетом параметров кривой в координаты экранного или печатного изображения. Соответственно работа с векторной графикой требует более производительных вычислительных систем.

Простейшие геометрические объекты (примитивы) создаются из элементарных объектов (линий), из которых в свою очередь составляются законченные композиции. Художественная иллюстрация, выполненная средствами векторной графики, может содержать десятки тысяч простейших объектов, взаимодействующих друг с другом.

Векторные редакторы удобны для создания изображений, но практически не используются для обработки готовых рисунков. Они нашли широкое применение в рекламном бизнесе: при оформлении обложек полиграфических изданий всюду, где стиль художественной работы близок к чертежному.

Редакторы трехмерной графики предназначены для создания трехмерных композиций и характеризуются следующими особенностями. Во-первых, позволяют гибко управлять взаимодействием свойств поверхности изображаемых объектов со свойствами источников освещения, во-вторых, позволяют создавать трехмерную анимацию. Поэтому редакторы трехмерной графики нередко называют также ЗО-аниматорами.

Системы управления базами данных (СУБД). Основными функциями системы управления базами данных являются:

* • создание пустой (незаполненной) структуры базы данных;
* • предоставление средств ее заполнения или импорта данных из таблиц другой базы;
* • обеспечение возможности доступа к данным, а также предоставление среды поиска и фильтрации.

Многие СУБД дополнительно предоставляют возможности проведения простейшего анализа данных и их обработки. В результате можно создавать новые таблицы баз данных на основе имеющихся. В связи с широким распространением сетевых технологий к современным СУБД предъявляется также требование обеспечения взаимодействия с распределенными ресурсами, находящимися на серверах Интернета.

Электронные таблицы. Электронные таблицы предоставляют комплексные средства для хранения различных типов данных и для их обработки. В некоторой степени они аналогичны СУБД, но основной акцент смещен на преобразование данных, причем в соответствии с их внутренним содержанием.

В отличие от баз данных, которые характеризуются широким спектром типов данных (от числовых и текстовых до мультимедийных), для электронных таблиц типичными являются числовые данные. Правда, электронные таблицы предоставляют широкий спектр методов для работы с данными числового типа. Так, при изменении содержания любых ячеек таблицы происходит автоматическое изменение содержания всех ячеек, связанных с измененными ячейками соотношением, заданным математическими или логическими выражениями (формулами). Простота и удобство работы с электронными таблицами обусловили их популярность в сфере бухгалтерского учета, в качестве универсальных инструментов анализа финансовых, сырьевых и товарных рынков, доступных средств обработки результатов технических испытаний, т. е. всюду, где необходимо автоматизировать регулярно повторяющиеся вычисления достаточно больших объемов числовых данных.

Системы автоматизированного проектирования (CAD-систе- мы). Предназначены для автоматизации проектно-конструкторских работ в машиностроении, приборостроении и архитектуре. Кроме чертежно-графических работ эти системы позволяют проводить простейшие расчеты (например, расчеты прочности деталей и т. д.) и выбор готовых конструктивных элементов из обширных баз данных.

Отличительная особенность CAD-систем состоит в автоматическом обеспечении на всех этапах проектирования технических условий, норм и правил, что освобождает конструктора (или архитектора) от рутинной работы. Так, в машиностроении CAD-системы способны на базе сборочного чертежа изделия автоматически выполнить рабочие чертежи деталей, подготовить регламентированную технологическую документацию с указанием последовательности переходов механической обработки, назначить необходимые инструменты, станочные и контрольные приспособления, а также подготовить управляющие программы для станков с числовым программным управлением (ЧПУ), промышленных роботов и гибких автоматизированных линий.

Настольные издательские системы. Обеспечивают автоматизацию процесса верстки полиграфических изданий; как программное обеспечение занимают промежуточное положение между текстовыми процессорами и системами автоматизированного проектирования.

Теоретически текстовые процессоры предоставляют средства для внедрения в текстовый документ объектов другой природы, например объектов векторной и растровой графики, а также обеспечивают взаимодействие между параметрами текста и внедренных объектов. Однако на практике для изготовления полиграфической продукции эти средства либо функционально недостаточны с точки зрения требований полиграфии, либо неудобны для производительной работы.

От текстовых процессоров настольные издательские системы отличаются, с одной стороны, расширенными средствами управления взаимодействием текста с параметрами страницы и с графическими объектами. С другой стороны, указанные системы характеризуются пониженными функциональными возможностями по автоматизации ввода и редактирования текста. Настольные издательские системы в основном применяют к документам, прошедшим предварительную обработку в текстовых процессорах и в графических редакторах.

Экспертные системы. Предназначены для анализа данных, содержащихся в базах знаний, и выдачи рекомендаций по запросу пользователя. Такие системы применяют в тех случаях, когда исходные данные хорошо формализуются, но для принятия решения требуются специальные знания. Характерными областями использования экспертных систем являются юриспруденция, медицина, фармакология, химия. По совокупности симптомов медицинские экспертные системы помогают установить диагноз и назначить лекарства, дозировку и программу лечебного курса. По совокупности признаков события юридические экспертные системы могут дать правовую оценку и предложить порядок действий как для обвиняющей, так и для защищающейся стороны.

Характерной особенностью экспертных систем является их способность к саморазвитию. Исходные данные хранятся в базе знаний в виде фактов, между которыми с помощью специали- стов-экспертов устанавливается определенная система отношений. Если на этапе тестирования экспертная система выдает некорректные рекомендации и заключения по конкретным вопросам или не дает их вообще, это означает либо отсутствие важных фактов в ее базе, либо нарушения в логической системе отношений. И в том и в другом случае экспертная система сама способна сгенерировать достаточный набор запросов к эксперту и автоматически повысить свое качество.

С использованием экспертных систем связана особая область научно-технической деятельности, называемая инженерией знаний. Инженеры знаний — это специалисты высокой квалификации, выступающие в качестве промежуточного звена между разработчиками экспертной системы (программистами) и ведущими специалистами в конкретных областях науки и техники (экспертами).

Редакторы HTML (веб-редакторы). Это особый класс редакторов, объединяющих свойства текстовых и графических редакторов; предназначены для создания и редактирования так называемых веб-документов (веб-страниц Интернета). Веб-докумен- ты — это электронные документы, при подготовке которых следует учитывать ряд особенностей, связанных с приемом/пе- редачей информации в Интернете.

Теоретически для создания веб-документов подходят обычные текстовые редакторы и процессоры, а также некоторые из графических редакторов векторной графики, но веб-редакторы обладают рядом полезных функций, повышающих производительность труда веб-дизайнеров. Программы этого класса можно также эффективно использовать для подготовки электронных документов и мультимедийных изданий.

Браузеры (обозреватели, средства просмотра веб-страниц). К этой категории относятся программные средства, предназначенные для просмотра электронных документов, выполненных в формате HTML (документы этого формата используются в качестве веб-документов). Современные браузеры воспроизводят не только текст и графику, но и музыку, человеческую речь, обеспечивают прослушивание радиопередач в Интернете, просмотр видеоконференций, работу со службами электронной почты, с системой телеконференций (групп новостей) и многое другое.

Интегрированные системы делопроизводства. Представляют собой программные средства автоматизации рабочего места руководителя. Среди функций подобных систем следует отметить создание, редактирование и форматирование простейших документов, централизацию функций электронной почты, факсимильной и телефонной связи, диспетчеризацию и мониторинг документооборота предприятия, координацию деятельности подразделений, оптимизацию административно-хозяйственной деятельности и поставки по запросу оперативной и справочной информации.

Бухгалтерские системы. Это специализированные программные системы, сочетающие функции текстовых и табличных редакторов, электронных таблиц и систем управления базами данных. Предназначены для автоматизации подготовки первичных бухгалтерских документов предприятия и их учета, для ведения счетов плана бухгалтерского учета, а также для автоматической подготовки регулярных отчетов по итогам производственной, хозяйственной и финансовой деятельности в форме, принятой для предоставления в налоговые органы, внебюджетные фонды и органы статистического учета. Несмотря на то, что теоретически все функции бухгалтерских систем способны исполнять неспециализированные программные средства, использование описываемых систем удобно благодаря интеграции разных средств в одной системе.

При внедрении на предприятии автоматизированной системы бухгалтерского учета следует обращать внимание на наличие в ней средств адаптации к изменениям нормативно-правовой базы. Поскольку нормативно-правовая база в России отличается крайней нестабильностью и подвержена частым изменениям, возможность гибкой перенастройки системы является обязательной функцией, хотя это требует от пользователей системы повышенной квалификации.

Финансовые аналитические системы. Используются в банковских и биржевых структурах; позволяют контролировать и прогнозировать ситуацию на финансовых, товарных и сырьевых рынках, проводить анализ текущих событий, готовить сводки и отчеты.

Геоинформационные системы (ГИС). Предназначены для автоматизации картографических и геодезических работ на основе информации, полученной топографическими или аэрокосмическими методами.

Системы видеомонтажа. Предназначены для цифровой обработки видеоматериалов, их монтажа, создания видеоэффектов, устранения дефектов, наложения звука, титров и субтитров.

Отдельные категории прикладных программных средств, обладающих развитыми внутренними системами классификации, представляют обучающие, развивающие, справочные и развлекательные системы и программы, характерной особенностью которых являются повышенные требования мультимедийной составляющей (использование музыкальных композиций, средств графической анимации и видеоматериалов).

Классификация служебных программных средств. Диспетчеры файлов (файловые менеджеры). Выполняют большинство операций, связанных с обслуживанием файловой структуры: копирование, перемещение и переименование файлов, создание каталогов (папок), удаление файлов и каталогов, поиск файлов и навигацию в файловой структуре. Базовые программные средства, предназначенные для этой цели, обычно входят в комплекс программ системного уровня и устанавливаются вместе с операционной системой, однако для повышения удобства работы большинство пользователей устанавливают дополнительные служебные программы.

Средства сжатия данных (архиваторы). Предназначены для создания архивов. Архивирование данных упрощает их хранение за счет того, что большие группы файлов и каталогов сводятся в один архивный файл. При этом повышается и эффективность использования носителя за счет того, что архивные файлы обычно имеют увеличенную плотность записи информации. Архиваторы часто применяют для создания резервных копий ценных данных.

Средства просмотра и воспроизведения. Для работы с файлами данных их загружают в прикладную систему, с помощью которой они были созданы. В результате появляется возможность просматривать документы и вносить в них изменения, но в тех случаях, когда требуется только просмотр без редактирования, используют более простые универсальные средства, позволяющие просматривать документы разных типов.

В случаях звукозаписи или видеозаписи вместо термина «просмотр» применяют термин «воспроизведение документов».

Средства диагностики. Предназначены для автоматизации диагностики программного и аппаратного обеспечения; выполняют необходимые проверки и выдают собранную информацию в удобном и наглядном виде. Их используют не только для устранения неполадок, но и для оптимизации работы компьютерной системы.

Средства контроля (мониторинга). Программные средства контроля иногда называют мониторами. Они позволяют следить за процессами, происходящими в компьютерной системе. При этом возможны два подхода: наблюдение в реальном режиме времени или контроль с записью результатов в специальном протокольном файле. Первый подход обычно используют при поиске путей оптимизации работы вычислительной системы и повышения ее эффективности; второй — в тех случаях, когда мониторинг выполняется автоматически и (или) дистанционно.

Зафиксированные результаты мониторинга передаются удаленной службе технической поддержки для установления причин конфликтов в работе программного и аппаратного обеспечения. Результаты работы в режиме реального времени особенно полезны для практического изучения функционирования компьютера, поскольку позволяют наглядно отображать его внутренние процессы.

Мониторы установки. Предназначены для контроля за установкой программного обеспечения. Необходимость в данном программном обеспечении связана с тем, что между различными категориями программного обеспечения возникают связи: вертикальные (между уровнями), которые являются необходимым условием функционирования всех компьютеров; горизонтальные (внутри уровней), характерные для компьютеров, работающих с операционными системами, поддерживающими принцип совместного использования одних и тех же ресурсов разными программными средствами. И в тех и в других случаях при установке или удалении программного обеспечения могут происходить нарушения работоспособности прочих программ.

Мониторы установки следят за состоянием и изменением окружающей программной среды; протоколируют образование новых связей и позволяют восстанавливать связи, утраченные в результате удаления ранее установленных программ.

Простейшие средства управления установкой и удалением программ обычно входят в состав операционной системы и размещаются на системном уровне, однако их почти всегда недостаточно. Поэтому в вычислительных системах, требующих повышенной надежности, используют дополнительные служебные программы.

Средства коммуникации (коммуникационные программы).

Приобрели очень большое значение с появлением электронной связи и компьютерных сетей. Они позволяют устанавливать соединения с удаленными компьютерами, обслуживают передачу сообщений электронной почты, работу с телеконференциями (группами новостей), обеспечивают пересылку факсимильных сообщений и выполняют множество других операций в компьютерных сетях.

Средства обеспечения компьютерной безопасности. Включают средства пассивной и активной защиты данных от повреждения, а также средства защиты от несанкционированного доступа, просмотра и изменения данных.

В качестве средств пассивной защиты используют служебные программы, предназначенные для резервного копирования. Нередко они обладают и базовыми свойствами диспетчеров архивов (архиваторов). В качестве средств активной защиты применяют антивирусное программное обеспечение. Для защиты данных от несанкционированного доступа, их просмотра и изменения служат специальные системы, основанные на криптографии.

Понятие об информационном и математическом обеспечении вычислительных систем. Наряду с аппаратным и программным обеспечением средств вычислительной техники в некоторых случаях целесообразно рассматривать информационное обеспечение, под которым понимают совокупность программ и предварительно подготовленных данных, необходимых для работы программ.

Рассмотрим, например, систему автоматической проверки орфографии в редактируемом тексте. Ее функция заключается в том, что лексические единицы исходного текста сравниваются с заранее заготовленным эталонным массивом данных (словарем), тогда для успешной работы системы необходимо иметь, кроме аппаратного и программного обеспечения, специальные наборы словарей, подключаемых извне. Это пример информационного обеспечения вычислительной техники.

В специализированных компьютерных системах (бортовых компьютерах автомобилей, судов, ракет, самолетов, космических летательных аппаратов и т. п.) совокупность программного и информационного обеспечения называют математическим обеспечением. Как правило, оно записано в микросхемы ПЗУ и может быть изменено только путем замены ПЗУ или его перепрограммирования на специальном оборудовании.

Совместимость и мобильность программного обеспечения. Концепция программной совместимости впервые в широких масштабах реализована разработчиками системы IBM/360. Основная задача при проектировании всего ряда моделей этой системы заключалась в создании такой архитектуры, которая была бы одинаковой с точки зрения пользователя для всех моделей системы независимо от цены и производительности каждой из них. Огромные преимущества такого подхода, позволяющего сохранять существующее программное обеспечение при переходе на новые (как правило, более производительные) модели, быстро оценили как производители компьютеров, так и пользователи; практически все фирмы-поставщики компьютерного оборудования взяли на вооружение эти принципы, поставляя серии совместимых компьютеров. Следует заметить, однако, что со временем даже самая передовая архитектура неизбежно устаревает и возникает потребность внесения радикальных изменений в архитектуру и в способы организации вычислительных систем.

В настоящее время одним из наиболее важных факторов, определяющих современные тенденции в развитии информационных технологий, является ориентация компаний-поставщиков компьютерного оборудования на рынок прикладных программных средств. Это объясняется прежде всего тем, что для конечного пользователя важно программное обеспечение, позволяющее решить его задачи, а не выбор той или иной аппаратной платформы. Переход от однородных сетей программно совместимых компьютеров к построению неоднородных сетей, включающих компьютеры разных фирм-производителей, в корне изменил и точку зрения на саму сеть: из сравнительно простого средства обмена информацией она превратилась в средство интеграции отдельных ресурсов — мощную распределенную вычислительную систему, каждый элемент которой (сервер или рабочая станция) лучше всего соответствует требованиям конкретной прикладной задачи.

Этот переход выдвинул ряд новых требований. Во-первых, такая вычислительная среда должна позволять гибко менять количество и состав аппаратных средств и программного обеспечения в соответствии с меняющимися требованиями решаемых задач. Во-вторых, она должна обеспечивать возможность запуска одних и тех же программных систем на различных аппаратных платформах, т. е. обеспечивать мобильность программного обеспечения. В-третьих, эта среда должна гарантировать возможность применения одних и тех же человеко-машинных интерфейсов на всех компьютерах, входящих в неоднородную сеть. В условиях жесткой конкуренции производителей аппаратных платформ и программного обеспечения сформировалась концепция открытых систем, представляющая собой совокупность стандартов на различные компоненты вычислительной среды, предназначенные для обеспечения мобильности программных средств в рамках неоднородной распределенной вычислительной системы.

Одним из вариантов моделей открытой среды является модель OSE (Open System Environment), предложенная комитетом IEEE POSIX. На основе этой модели Национальный институт стандартов и технологии США выпустил документ «Application Portability Profile (АРР). The U.S. Government's Open System Environment Profile OSE/1 Version 2.0», который определяет рекомендуемые для федеральных учреждений США спецификации в области информационных технологий, обеспечивающие мобильность системного и прикладного программного обеспечения. В настоящее время все ведущие производители компьютеров и программного обеспечения придерживаются требований этого документа.

## Как установить драйвер в режиме совместимости?

Установка драйвера в режиме совместимости нужна, если вы подключаете к компьютеру с обновленной операционной системой старое оборудование. Например, принтер или МФУ сняты с производства, а производитель официально объявил, что продукт больше не поддерживается и ПО обновляться не будет.

В этом случае другого выхода нет и нужно устанавливать последнюю выпущенную версию. Однако в первую очередь все-таки попробуйте просто установить драйвер как будто он подходит к вашей ОС. Если корректной установки не получилось, переходим к решению возникших проблем.

### Установка в режиме совместимости

Кликните правой кнопкой мыши по установщику драйвера (.exe). Откройте Свойства и найдите вкладку Совместимость:

1. Запустите средство устранения проблем с совместимостью и выберите из двух предложенных вариантов первый: установку по тем параметрам, которые рекомендуются. Выполняйте подсказки мастера установки программы. Если установка прошла успешно, параметры нужно сохранить.
2. В противном случае на той же вкладке поставьте маркер в строке Запустить программу в режиме совместимости с, и выберите версию системы, для которой и был разработан драйвер, например, Windows 7. Еще раз запустите установку программы.
3. Если проблемы остаются, можно ввести дополнительные параметры совместимости вручную. Например для старых программ кликнуть строку Режим пониженной цветности и выбрать 8 или 16-разрядный цвет.

### Устранение проблем с совместимостью

В свойствах файла для установки драйвера не всегда будет вкладка Совместимость, где можно вручную задать некоторые параметры установки. Но в Панели управления (на примере Windows 10) есть целый раздел под названием Устранение неполадок. Если предыдущие действия не помогли или в работе драйвера замечены лаги:

1. Откройте раздел Устранение неполадок , выберите строку Программы/Устранение проблем с совместимостью.
2. В открывшемся окне запустите диагностику от имени администратора.
3. Немного подождите и посмотрите список программ. После клика по строке Нет в списке можно выбрать файл на компьютере. Или выберите нужную программу из перечня.
4. Поскольку установка с рекомендованными параметрами не помогла, выбираем диагностику программы и одну из драйвер работал (или для которой был разработан).
5. Нажмите Проверить программу. Если проблема решена, сохраните параметры и завершите диагностику
6. Опишите причины возникновения проблем совместимости. Охарактеризуйте методы выявления проблем совместимости ПО.

Программные неисправности компьютера встречаются намного чаще, чем аппаратные. Конечно, ничего хорошего в их появлении нет, однако радует то, что такие неисправности можно без труда и риска исправить в домашних условиях.

Размышлять над причинами возникновения программных неисправностей особого смысла не имеет, поскольку не существует программного обеспечения, которое бы стабильно работало в любых условиях.

Мало того, его никогда не будет существовать. Поэтому не очень удивляйтесь, если обнаружите неисправности такого плана: рано или поздно они обязательно появляются.

Рассмотрим основные причины возникновения подобного рода ошибок.

**Несовершенство программного обеспечения**. Не все люди, которые занимаются написанием программного обеспечения, являются высококвалифицированными программистами. В основном это самоучки, начитавшиеся умных книг, знакомые с основами программирования. Написав программу, которая работает у него на компьютере, такой «программист» решает, что она будет работать и у других, совершенно не учитывая тот факт, что на компьютерах пользователей могут быть установлены разные операционные системы, компьютеры могут обладать разными ресурсами, устройствами, использовать разное программное обеспечение и т. д.

Это, конечно, совсем не означает, что профессионалы пишут программы без проблем: их преследуют те же ошибки. Однако квалифицированные программисты умеют адаптировать программу для работы в разных условиях, что делает ее менее уязвимой и, соответственно, более корректной в работе.

Очень часто бывает так, что человек садится за написание программы, не думая о том, что при работе она будет захватывать различные ресурсы, контролировать которые в одиночку не удастся. В результате появляется программное обеспечение, имеющее множество недочетов и постоянно выдающее ошибки.

Достаточно большой проблемой является и то, как программа распоряжается установленными в операционной системе системными библиотеками. Иногда системные библиотеки заменяются улучшенными версиями, с которыми отказываются работать другие программы, что, в свою очередь, также приводит к появлению ошибок.

**Несовершенство операционной системы**. Какими бы совершенными ни были операционные системы, они никогда не смогут создать нормальные условия для работы всего существующего программного обеспечения. Кроме того, совместимость операционных систем с выпуском каждой новой системы ухудшается. В таких условиях разработчики вынуждены писать программы, ориентированные на конкретную операционную систему.

Пользователю остается или работать с сертифицированными программами, или мириться со сложившейся ситуацией. Иногда и выбирать не приходится, ведь многие программы распространяются бесплатно (можно догадаться, какое у большой части из них качество работы).

**Отсутствие нужных ресурсов.** Часто даже небольшая программа требует для нормальной работы большее количество ресурсов, чем ей может предоставить операционная система. В таком случае для обеспечения работы программы система увеличивает файл подкачки, отбирающий оперативную память у других программ. Это приводит не только к снижению производительности компьютера, но и к появлению разнообразных ошибок.

Обычно такую проблему можно решить простым увеличением объема оперативной памяти. Однако это можно сделать и с помощью специальных утилит.

**Ошибки в реестре.** Реестр — мозг операционной системы. Ошибки в нем негативно сказываются на всех процессах, происходящих в компьютере. Причиной возникновения сбоев в реестре являются все те же непрофессионально написанные программы, прописывающие свои файлы и ссылки в самых различных местах. Не стоит также забывать о троянских конях и «червях», которые только и мечтают о том, чтобы что-нибудь подпортить.

Для «лечения» реестра предназначены специальные утилиты, умеющие анализировать записи и удалять ошибочные и не используемые данные. При этом не следует забывать об элементарном сохранении рабочей версии файлов реестра.

**Конфликты между устройствами**. Аппаратное обеспечение, как и программное, также не всегда бывает полностью совместимым. При использовании устройств доморощенных производителей вы обязательно получите целый букет подобных конфликтов. Например, перестанет работать звуковая карта или пропадет один из приводов компакт-дисков.

Прежде всего в такой ситуации следует проанализировать используемые конфликтующими устройствами прерывания и ресурсы. Для этого воспользуйтесь Диспетчером устройств или просмотрите отчеты об ошибках работы системы.

Достаточно часто в инструкции к устройству производители указывают несовместимые с ним комплектующие других производителей. Поэтому при сборке или модернизации компьютера обязательно внимательно изучите документацию к каждому из приобретенных устройств.

Часто подобные конфликты можно решить переустановкой платы расширения в другой слот или вручную, назначив им свободные ресурсы (с помощью Диспетчера устройств в свойствах конкретного устройства можно назначить прерывания и диапазон адресов). В случае с Windows Vista еще один действенный способ — замена проблемного устройства более новой моделью.

**Вирусы, троянские кони и «черви».** Если вы пользуетесь Интернетом, то вам не избежать неприятных сюрпризов, таких как различные вирусы, трояны, «черви», программы-шпионы и т. д.

Трудно сказать, когда Интернет наполнился вирусами, однако на сегодняшний день ни один пользователь Интернета не в состоянии полностью обезопасить свой компьютер от этой напасти. По данным некоторых исследований, на компьютерах, часто работающих с Интернетом, может находиться одновременно до 30 различных модулей описанного характера. Порой разного рода троянские кони и «черви» проникают даже сквозь, казалось бы, непробиваемую защиту антивирусной программы и брандмауэра.

Однако в любом случае необходимо использовать хорошую антивирусную программу, а также программу поиска модулей-шпионов и блокирования программных портов (брандмауэр). В противном случае вы обречете себя на частую переустановку операционной системы.

**Ограничения операционной системы**. Чтобы обезопасить свои системные файлы и вашу работу за компьютером в целом, Windows Vista часто не разрешает выполнять те или иные действия: устанавливать новые программы, удалять файлы, просматривать веб-ресурсы и т. д.

Как результат — появление «невинных» и не очень ошибок, которые могут нарушить работу программ. В подобном случае можно только отказаться от задуманного либо найти новую версию используемой программы.

Однако бывают и такие ограничения, которые не убрать установкой новой версии программы. В частности, если вы хотите получить от

Windows Vista Starter такую же функциональность, какой обладает Windows Vista Ultimate, о подобной затее можете забыть навсегда. Единственный выход — перейти на более продвинутый выпуск Windows Vista, например Windows Vista Home Premium.

**Использование устаревшего оборудования**. Не секрет, что Windows Vista очень требовательна к используемому оборудованию, поэтому часто случается так, что при переходе с Windows XP на Vista вы в добавок к новой операционной системе получаете целый букет ошибок.

Даже если вы сможете установить драйвер для устаревшего устройства, появления разного рода ошибок, скорее всего, не избежать. Операционная система будет постоянно «напрягать» вас сообщениями о том, что «такое-то устройство не удалось инициализировать», и оно либо вообще не будет работать, либо будет работать с ошибками.

Единственный выход из такой ситуации (если вы, конечно, по-прежнему хотите использовать Windows Vista) — поискать новые драйверы для подобных устройств либо найти им замену среди современных моделей.

**Неверные настройки операционной системы**. Настройкой операционной системы можно решить самые распространенные неполадки: неверное разрешение или мерцание экрана, исчезновение языков, отсутствие или искажение звука, низкую скорость модема, проблемы с локальной сетью и т. п.

Не забывайте: операционная система будет работать так, как вы ее настроите. Поэтому к изменению значений любых параметров следует относиться внимательно. Если вы не знаете точно, как новые настройки отразятся на работе системы, лучше ничего не менять.

Обычно для настройки операционной системы используется Панель управления Windows, реже — менеджеры устройств. Чтобы настроить многие скрытые возможности системы, часто применяют специализированные программы.

## Общие методы уменьшения проблемы с совместимостью

Совместимы ли Ваши приложения с Windows 7, поможет определить подключение Application Compatibility Toolkit (ACT) 5.5. ACT также помогает определить, как будут влиять на Ваши приложения апгрейды. Так же Вы функции ACT могут использоваться для:

* Проверки своих приложений, устройств и компьютера на совместимость с новой версией операционной системы Windows
* Проверки совместимости обновления Windows
* Подключения в сообщество ACT и совместной оценки риска с другими пользователями ACT
* Тестирования своих Веб-приложений и Веб-сайтов на возможность проблем совместимости с новыми выпусками и обновлениями системы защиты Internet Explorer.

### Методы уменьшения проблем с совместимостью

Уменьшение **проблем с совместимостью** приложения обычно зависит от различных факторов, таких как тип приложения и текущей поддержки приложения. Некоторые из общих методов включают следующее:

* Изменение конфигурации существующего приложения: Вы можете использовать инструменты, Compatibility Administrator или Standard User Analyzer (устанавливается с ACT), для обнаружения проблемы и создания исправления данного приложения, что решит проблему совместимости.
* Применение обновлений или пакетов обновлений к приложению: обновления или пакеты обновлений могут помочь решить многие из проблем с совместимостью и дать возможность приложению работать в новой среде операционной системы.
* Апгрейд приложения до совместимого релиза: если более новая, совместимая версия приложения существует, лучшее решение — обновить до более новой версии.
* Изменение конфигурации безопасности: как пример, Защищенный режим Internet Explorer может быть смягчен, добавив сайт в список надежных сайтов или выключив Защищенный режим (что не рекомендуется).
* Запуск приложения в виртуализированной среде: если все другие методы недоступны, для решения проблем Вы можете запустить приложение в более раннем релизе Windows, используя инструменты виртуализации, такие как PC Microsoft Virtual и Microsoft Virtual Server.
* Использование функций совместимости приложения: проблемы приложения, такие как управление версиями операционной системы, могут быть смягчены, запуском приложения в режиме эмуляции. К этому режиму можно получить доступ, щелкнув правой кнопкой по ярлыку или .exe файлу и применяя режим эмуляции более ранней версии Windows на вкладки «**Совместимость**» (**Свойства -> Совместимость**). Так же, чтобы помочь в конфигурировании режима эмуляции с приложением, Вы можете использовать "**Мастер Совместимости Программ**". Эту функцию можно найти так: «**Панель управления» -> «Программы» -> «Выполнение программ, созданных для предыдущих версий Windows**».
* Выбор другого приложения, которое выполняет ту же самую функцию, но не имеет проблем с совместимостью: если другое совместимое приложение доступно, Вы можете использовать его.

1. Дать определение понятию чистая загрузка. Опишите процесс выявления причин возникновения проблем совместимости ПО.

**Выполнение чистой загрузки**

Если компьютер не запускается в нормальном режиме или при его запуске появляются непонятные ошибки, возможно, необходимо применить «чистую» загрузку. При данной операции компьютер загружается с минимальным набором драйверов и программ из автозагрузки. Это может помочь в решении проблемы.

"Чистая загрузка" - это безопасный режим, в котором компьютер запускается с минимальным количеством работающих программ и служб. Если при запуске компьютера в безопасном режиме ошибок не возникает, это означает, что проблема вызвана одной из программ или служб автозагрузки. В режиме "чистой загрузки" компьютер запускается с отключенными программами и службами автозагрузки, которые затем включаются и отключаются по одной до тех пор, пока источник проблемы не будет локализован.

Как настроить Windows XP для чистой загрузки и как затем вернуть Windows XP к нормальной загрузке. После настройки Windows XP на "чистую" загрузку некоторые программы могут работать не так, как ожидалось. При возвращении к обычному типу загрузки Windows XP программы работают снова.

Существует два способа запуска Windows XP с использованием чистой загрузки.

· Способ 1. Использование интерактивной помощи

· Способ 2. Ручной запуск Windows XP в режиме чистой загрузки (только для опытных пользователей)

Начните с первого способа. Если он не поможет и вы готовы применить более продвинутый способ, переходите ко второму способу.

Способ 1. Интерактивная помощь

**Подготовка**

Перед запуском интерактивной помощи убедитесь в выполнении перечисленных ниже условий.

· Интерактивная помощь может использоваться только на компьютерах с ОС Windows XP Home Edition, Windows XP Professional, Windows XP Media Center Edition или Windows XP Tablet PC Edition.

· Для настройки Windows XP на чистую загрузку необходимо войти в систему с учетной записью администратора или члена группы «Администраторы».

Для получения дополнительных сведений о том, как войти в Windows, щелкните следующий номер статьи базы знаний Майкрософт:

Как войти в систему Windows XP в случае потери или истечения срока действия пароля

Способ 2. Ручной запуск Windows XP в режиме чистой загрузки

Этот раздел предназначен для опытных пользователей. Если продвинутые способы устранения неполадок кажутся слишком сложными, можно попробовать обратиться за помощью к специалисту или в службу поддержки. Сведения о том, как обратиться в службу технической поддержки Майкрософт, см. на следующем веб-узле корпорации Майкрософт:

http://support.microsoft.com/contactus/?ln=ru

Для запуска Windows XP в режиме чистой загрузки вручную выполните указанные ниже действия.

**Действие 1. Запустите программу «Настройка системы»**

1. В меню **Пуск** выберите пункт **Выполнить**, введите команду **msconfig** и нажмите кнопку **ОК**.

2. Откроется диалоговое окно **Настройка системы**.

**Действие 2. Настройка выборочной загрузки**

1. В диалоговом окне **Программа настройки системы** выберите вкладку **Общие**, затем отметьте **Выборочный запуск**.

2. Снимите флажок **Обрабатывать файл SYSTEM.INI**.

3. Снимите флажок **Обрабатывать файл WIN.INI**.

4. Снимите флажок **Загружать элементы автозагрузки**. Убедитесь, что выбраны пункты **Загружать системные службы** и **Использовать оригинальный BOOT.INI**.

5. Перейдите на вкладку **Службы**.

6. Установите флажок **Не отображать службы Microsoft**.

7. Нажмите кнопку **Отключить все**, затем нажмите кнопку **ОК**.

8. При запросе на перезагрузку компьютера нажмите кнопку **Перезагрузка**.

## Методы выявления проблем совместимости

При переходе на новую версию операционной системы неизбежно возникает проблема совместимости приложений, которая должна быть решена в ходе обслуживания компьютеров.

Проблема совместимости чаще всего возникает, когда приложение некорректно обращается к функциям операционной системы.

Как правило, ведущие производители программного обеспечения стараются своевременно выпускать новые версии программных продуктов, совместимые с актуальными операционными системами, но нередко предприятия для решения некоторых специфических задач вынуждены использовать ПО, совместимость которого с новой версией операционной системы находится под вопросом.

С учетом возможных проблем совместимости используемых приложений и новой версии операционной системы рекомендуется в обязательном порядке включить в подготовку к обновлению ОС на компьютерах предприятия следующие этапы:

* обязательная проверка используемого ПО на совместимость с новой операционной системой и сбор информации о выявленных в процессе этого тестирования проблемах
* В процессе этого анализа нужно выяснить, насколько необходимы предприятию данные приложения, выпущены ли для них новые версии, в которых решена проблема совместимости, и в случае отсутствия таких версий - поиск инструментов для решения проблем совместимости
* тестирование инструментария для решения проблем совместимости, желательно не слишком усложняющего обслуживание компьютеров.

Тестирование на совместимостьпозволяет проверить приложение в различных браузерах, на разных ОС, с различным программным обеспечением и аппаратными средствами.

Знаете ли вы, что 10% пользователей используют нестандартные браузеры?

Вы знаете, как выглядит ваш сайт в этих браузерах? И работает ли он во всех этих браузерах? Не очень то бы хотелось потерять 10% потенциальных клиентов.

Во времятестирования на совместимость  команда тестировщиков проверяет, насколько хорошо приложение может работать с различными типами аппаратных технологий, драйверами иоперационными системами.

Инсталляционное тестирование (installation testing)

В процессе инсталляционного  тестирования проверяется корректность установки  и  удаления   программного  продукта  в  среде,  максимально  приближенной  к  эксплутационной.  Об  этом  аспекте  корректной  работы  программного обеспечения очень часто просто забывают (и напрасно).Правильно выполненная установка программы — необходимое условие её корректной дальнейшей работы. Проверка правильности установки должна быть обязательным элементом проекта по тестированию любого продукта. Если программу невозможно корректно установить, и при этом что-то не будет работать или будет работать неправильно, работа по тестированию самого программного тестирования бессмысленна. Почему? Потому что  заказчику не нужен продукт, который даже невозможно установить. Если пользователь уже на этапе установки сталкивается  с проблемами в разработанном программном продукте, что он подумает о самом программном продукте? Будет ли он связываться с такой фирмой-разработчиком?

Регрессионное тестирование (regression testing)

Повторное выполнение тестов для проверки того, что изменения, внесённые  в  программу  в  результате  разработки новой или изменения  существующей  функциональности,  устранения  ошибок,  не  повлияли  на  функциональность, которая не изменялась (т.е. текущая версия ведёт себя идентично предыдущей, за исключением измененных областей).

Тестирование новой функциональности (new feature testing)

В  данном  виде  тестирования  акцент  делается  на  тестировании  новой функциональности, появившейся в конкретном выпуске (build) программного продукта.

Конфигурационное тестирование (configuration testing)

данные тесты.

Тестирование совместимости (compatibility testing)

Тестирование  совместимости  помогает  убедиться  в  функциональных возможностях и надёжности  работы продукта  в поддерживаемых  браузерах (если речь идет о Web-приложениях) и  операционных системах. Также может проверяться работоспособность продукта при использовании различных аппаратных платформ.

Тестирование удобства эксплуатации (usability testing)

Тестирование  интерфейса  человек/машина  производится  в  отношении таких  моментов  как  внешний  вид  пользовательского  интерфейса,  удобство навигации (преимущественно для Web-сайтов). Практичность и удобство использования –  очень  важные  характеристики  программного  продукта.  Например, программа может вполне соответствовать всем предъявляемым к ней требованиям  с  точки  зрения  функциональности.  Но  функции  реализованы неудобно: некоторые шаги приходится повторять много раз, тогда как по логике  достаточно  выполнить  однажды;  расположение  элементов  интерфейса нелогично, программа быстро вызывает утомление и т.д. Для выявления такого  рода  недочётов  и  применяют  тесты  на  удобство  использования. Часто эта  группа  тестов  относится  к  категории  некритичных,  но  когда  речь  идёт, например,  о  рыночном  готовом  продукте,  пренебрегать  удобством  эксплуатации весьма опасно.

Программы тестирования совместимости

QATestLab предлагает:

* тестирование на совместимость с аппаратным обеспечением
* тестирование совместимости баз данных
* тестирование совместимости с операционными системами

Тестирование совместимости аппаратного обеспечения используется для оценки производительности системы / приложения / веб-сайта на определенной аппаратной платформе.

Тестирование совместимости баз данных используется для оценки работы приложения / системы во взаимодействии с базой данных.

Tестирование совместимости с операционными системами используется, чтобы оценить производительность системы / приложения во взаимодействии с операционной системой.

1. Опишите процесс выбора методов выявления совместимости. Охарактеризуйте проблемы перехода на новые версии программ.

При переходе на новую версию операционной системы неизбежно возникает проблема совместимости приложений, которая должна быть решена в ходе обслуживания компьютеров.

Проблема совместимости чаще всего возникает, когда приложение некорректно обращается к функциям операционной системы.

Как правило, ведущие производители программного обеспечения стараются своевременно выпускать новые версии программных продуктов, совместимые с актуальными операционными системами, но нередко предприятия для решения некоторых специфических задач вынуждены использовать ПО, совместимость которого с новой версией операционной системы находится под вопросом.

С учетом возможных проблем совместимости используемых приложений и новой версии операционной системы рекомендуется в обязательном порядке включить в подготовку к обновлению ОС на компьютерах предприятия следующие этапы:

· обязательная проверка используемого ПО на совместимость с новой операционной системой и сбор информации о выявленных в процессе этого тестирования проблемах

· Анализ приложений, у которых выявлены проблемы совместимости.  
В процессе этого анализа нужно выяснить, насколько необходимы предприятию данные приложения, выпущены ли для них новые версии, в которых решена проблема совместимости, и в случае отсутствия таких версий - поиск инструментов для решения проблем совместимости

· тестирование инструментария для решения проблем совместимости, желательно не слишком усложняющего обслуживание компьютеров.

**Тестирование на совместимость** позволяет проверить приложение в различных браузерах, на разных ОС, с различным программным обеспечением и аппаратными средствами.

**Знаете ли вы, что 10% пользователей используют нестандартные браузеры?**

Вы знаете, как выглядит ваш сайт в этих браузерах? И работает ли он во всех этих браузерах? Не очень то бы хотелось потерять 10% потенциальных клиентов.

Во время **тестирования на совместимость** команда тестировщиков проверяет, насколько хорошо приложение может работать с различными типами аппаратных технологий, драйверами и операционными системами.

**Инсталляционное тестирование (installation testing)**

В процессе инсталляционного тестирования проверяется корректность установки и удаления программного продукта в среде, максимально приближенной к эксплутационной. Об этом аспекте корректной работы программного обеспечения очень часто просто забывают (и напрасно). Правильно выполненная установка программы — необходимое условие её корректной дальнейшей работы. Проверка правильности установки должна быть обязательным элементом проекта по тестированию любого продукта. Если программу невозможно корректно установить, и при этом что-то не будет работать или будет работать неправильно, работа по тестированию самого программного тестирования бессмысленна. Почему? Потому что заказчику не нужен продукт, который даже невозможно установить. Если пользователь уже на этапе установки сталкивается с проблемами в разработанном программном продукте, что он подумает о самом программном продукте? Будет ли он связываться с такой фирмой-разработчиком?

**Регрессионное тестирование (regression testing)**

Повторное выполнение тестов для проверки того, что изменения, внесённые в программу в результате разработки новой или изменения существующей функциональности, устранения ошибок, не повлияли на функциональность, которая не изменялась (т.е. текущая версия ведёт себя идентично предыдущей, за исключением измененных областей).

**Тестирование новой функциональности (new feature testing)**

В данном виде тестирования акцент делается на тестировании новой функциональности, появившейся в конкретном выпуске (build) программного продукта.

**Конфигурационное тестирование (configuration testing)**

С помощью конфигурационных тестов проверяется совместимость продукта с различным программным (software) и аппаратным (hardware) обеспечением. Как правило, программный продукт делается с тем расчётом, чтобы он сразу работал в максимально разнообразной внешней среде. Если же речь идёт о «коробочном продукте», то фактор совместимости приобретает ещё более важное значение. Для того, чтобы выяснить реакцию продукта на окружение и соседство с другим программным обеспечением, и проводят  
данные тесты.

**Тестирование совместимости (compatibility testing)**

Тестирование совместимости помогает убедиться в функциональных возможностях и надёжности работы продукта в поддерживаемых браузерах (если речь идет о Web-приложениях) и операционных системах. Также может проверяться работоспособность продукта при использовании различных аппаратных платформ.

**Тестирование удобства эксплуатации (usability testing)**

Тестирование интерфейса человек/машина производится в отношении таких моментов как внешний вид пользовательского интерфейса, удобство навигации (преимущественно для Web-сайтов). Практичность и удобство использования – очень важные характеристики программного продукта. Например, программа может вполне соответствовать всем предъявляемым к ней требованиям с точки зрения функциональности. Но функции реализованы неудобно: некоторые шаги приходится повторять много раз, тогда как по логике достаточно выполнить однажды; расположение элементов интерфейса нелогично, программа быстро вызывает утомление и т.д. Для выявления такого рода недочётов и применяют тесты на удобство использования. Часто эта группа тестов относится к категории некритичных, но когда речь идёт, например, о рыночном готовом продукте, пренебрегать удобством эксплуатации весьма опасно.

1. Охарактеризуйте инструментарий учета аппаратных компонентов.
2. Опишите процесс анализа приложений с проблемами совместимости.
3. Опишите механизм решения проблем совместимости на основе «системных заплаток».

## Решение проблем совместимости приложений.

### Введение

Одним из самых важных новшеств в Microsoft® Windows® XP стало добавление целого ряда технологий совместимости приложений, доступных даже конечным пользователям через оболочку Windows XP. Распространение исправлений совместимости приложений на большом количестве компьютеров может быть трудным или невыполнимым, если оно предоставлено каждому пользователю компьютера. К счастью, есть более простой способ собирать группы исправлений совместимости и распределять их путем автоматической установки на компьютеры, работающие под управлением Windows XP.

### Администратор совместимости

После установки группы необходимых исправлений совместимости, Вы можете воспользоваться Администратором совместимости, чтобы скомпоновать исправления совместимости для распространения на других компьютерах, работающих под управлением Windows XP.

### Создание собственных оболочек совместимости с помощью Администратора совместимости

В этом разделе обсуждается как можно создавать и подготавливать файлы собственной базы данных с помощью Администратора совместимости, для поддержания множества приложений на одном или нескольких компьютерах, работающих под управлением Windows XP.

Администратор совместимости может компоновать исправления и оболочки совместимости для множества приложений в один файл базы данных совместимости (\*.sdb), который потом может быть перенесен на другие компьютеры, работающие под управлением Windows XP. Это особенно полезно в большом сетевом окружении, где несколько человек должны обеспечивать поддержку программного обеспечения огромному числу пользователей.

### Установка Администратора совместимости

Администратор совместимости, поставляемый с операционной системой Windows XP, может быть найден в папке Support Tools на установочном компакт-диске. Администратор совместимости распространяется как часть Пакета средств обеспечения совместимости приложений (Application Compatibility Toolkit) версии 2.0 и выше.

Для установки Пакета средств обеспечения совместимости приложений (Application Compatibility Toolkit) в Вашей ОС Windows XP:

1. Вставьте установочный компакт-диск Windows XP в привод компакт-дисков
2. Используя Мой компьютер ( My Computer ) или Проводник ( Windows Explorer), перейдите на привод, в который Вы вставили диск с ОС Windows XP, и откройте папку Support Tools.
3. Щелкните дважды файл ACT. EXE для начала установки программы. Примите настройки, предложенные по умолчанию программой установки.

После установки Пакета средств обеспечения совместимости приложений (Application Compatibility Toolkit) его можно будет найти в меню Пуск. Администратор совместимости находится в группе Пакета средств обеспечения совместимости приложений (Application Compatibility Toolkit) в меню Пуск.

### Использование Администратора совместимости

Windows XP содержит информацию о распространенных проблемах совместимости, которые могут происходить с некоторыми приложениями. Исправления совместимости приложений, предоставляемые Microsoft в Windows XP, созданы, чтобы помочь Windows XP должным образом поддерживать нормальную работу этих приложений, не ставя при этом под угрозу стабильность системы.

Четыре библиотеки DLL содержат все исправления совместимости  
Четыре библиотеки DLL, расположенные в папке % WINDIR% AppPatch, содержат все исправления совместимости. Файлы APPHELP.SDB и SYSMAIN.SDB обеспечивают работу справочных сообщений приложений, а исправления приложений являются частью Windows XP.

Справочные сообщения приложений содержат информацию, которая отображается при запуске действительно несовместимого с Windows XP приложения. Примеры приложений, которые могут вызвать появление Справочных сообщений приложений, включают:

* Антивирусные программы
* Программы, которые требуют доступа на уровне ядра операционной системы
* Программы, которые устанавливают специфические драйверы файловой системы

Причины неправильной работы приложений  
Приложения, которые были созданы для работы с предыдущими версиями Windows, могут неправильно работать в ОС Windows XP Professional. Причины, по которым это может происходить:

* Приложение не запускается, когда Windows сообщает ему о новой версии операционной системы. Зачастую приложение работает нормально, если пользователь сможет обойти этот момент.
* Приложение обращается к старым версиям функций Win32 API, которые возвращают непредсказуемые значения на компьютерах с большим количеством ресурсов, таких как дисковое пространство
* Приложение ожидает старых форматов данных Windows.
* Приложение ожидает, что информация пользователя, такая как личные и временные папки, будет в определенном месте или в определенном формате.

Для устранения этих проблем с помощью Администратора совместимости Вам необходимо создать собственную базу данных, содержащую информацию об исправлениях совместимости, необходимую Вашим приложениям, а также информацию о соответствии файлов, которая позволяет Windows XP однозначно распознать приложение, требующее поддержки.

### Создание собственной базы данных совместимости

Администратор совместимости позволяет Вам просматривать исправления совместимости приложений, хранящиеся в защищенных системой базах данных, чтобы применять нужные исправления для сотен приложений. Основной интерфейс Администратора позволяет контролировать приложения с исправлениями совместимости путем просмотра их в базе данных ОС Windows XP Professional. Эта информация отображается в верхней левой части (части системной базы данных) окна Администратора совместимости.

Системная база данных совместимости является составляющей операционной системы Windows XP Professional, обеспечивающей идеальную совместимость для сотен Windows-приложений. Эта база данных и соответствующие компоненты защищены операционной системой.

Как только Вы определили и проверили исправления для определенного приложения, можно запустить Администратор совместимости для создания базы данных исправлений. Вы можете создать базу данных, которая содержит приложения, поддерживаемые оболочками совместимости, или приложения, поддерживаемые определенными исправлениями совместимости.

Чтобы создать новую собственную базу данных с помощью Администратора совместимости:

1. Откройте Администратор совместимости выбрав в меню Пуск ( Start), Программы( All Programs), Пакет средств обеспечения совместимости приложений (Application Compatibility Toolkit), Администратор совместимости
2. Если у Вас открыта собственная база данных, в меню Файл ( File) выберите Новый ( New).
3. Зайдите в меню База данных ( Database) и нажмите Изменить название базы данных ( Change Database Name ). Как только Вы измените название базы данных, оно будет отображаться в заголовке собственной базы данных. Если пункт менюИзменить имя базы    данных ( Change Database Name ) не активен, щелкните по области базы данных окна.
4. В меню Файл ( File) нажмите Сохранить ( Save) и дайте название своему .sdb файлу. Теперь можно добавить исправления в Вашу собственную базу данных.

Как только Вы создали собственную пустую базу данных, которая будет содержать Ваши исправления совместимости приложений, можно добавить оболочку совместимости.

Для добавления оболочки совместимости

1. Выберите Создать исправление приложения ( Create Application Fix) в меню База данных. Появится диалоговое окноСоздание исправления приложения ( Create an Application Fix ).
2. Выберите Применить режим совместимости ( Apply Compatibility Mode ) и нажмите кнопку Далее ( Next).
3. Введите название приложения, для которого Вы будете определять режим совместимости, и нажмите кнопку Далее( Next).
4. Введите название файла, к которому будет применен режим совместимости. Вы можете набрать название файла вручную или использовать кнопку Обзор ( Browse), чтобы указать его.
5. Выберите из выпадающего списка режим совместимости, который нужно применить, и нажмите Далее ( Next).
6. Нажмите кнопку Добавить файл ( Add File), чтобы выбрать файлы, которые помогут точно определить нужный файл на целевых компьютерах (Выберите файлы, связанные с приложением, которые будут установлены в то же место. Например, выберите файл .hlp, находящийся в одной папке с .exe файлом. Постарайтесь однозначно определить Ваш файл, не выбирая большое количество соответствующих файлов).
7. Когда выберете все необходимые файлы, нажмите Далее ( Next).
8. Если Вы хотите проверить приложение с примененным исправлением, нажмите Выполнить тестирование ( Test Run). В противном случае нажмите Готово ( Finish).

Тот же процесс может быть использован для добавления индивидуальных исправлений совместимости в собственную базу данных, за исключением того, что в окне Создать исправление приложения ( Create an Application Fix ) Вы должны выбрать вариантПрименить определенное исправление совместимости ( Apply Specific Compatibility Fix ). Как только все исправления и оболочки будут добавлены в базу данных, сохраните базу данных и проверьте приложение.  
Совпадение имен файлов

### Применение собственной базы данных к системе

Как только Вы создали Вашу собственную базу данных исправлений совместимости приложений, она должна быть применена к системе компьютера, на котором это приложение будет работать. Общий процесс развертывания исправлений совместимости на нескольких компьютерах под управлением Windows XP включает следующие действия:

* Определите и проверьте исправления для необходимых приложений.
* Создайте файл выборочной базы данных с нужными исправлениями.
* Перенесите .sdb файл на нужные компьютеры под управлением Windows XP.
* Используйте команду SDBINST.EXE, чтобы зарегистрировать базу данных. Она автоматически установит и добавит информацию об исправлениях в реестр на выбранных компьютерах.

Перенос файла собственной базы данных на другие компьютеры под управлением Windows XP  
Перенос файла собственной базы данных на другие компьютеры под управлением Windows XP может быть проведена разными способами:

* Можно поместить файл базы данных в программу установки и распространить его с помощью Групповой политики в сети с Active Directory, но это требует дополнительной работы.
* Файл может быть скопирован вручную на каждый удаленный компьютер, или это можно сделать с помощью сценария входа в систему.
* Еще одной возможностью является размещение файла .sdb на общем сетевом ресурсе, к которому имеют доступ все пользователи Windows XP.

Несмотря на то, что файл перенесен на удаленные компьютеры, содержащаяся в нем информация должна быть зарегистрирована на каждом компьютере. Это делается с помощью запуска команды SDBINST.EXE из командной строки, за которой следует полный путь и имя созданного .sdb файла. Например:

Sdbinst c:WindowsAppPatchmyapp.sdb

Как только база данных зарегистрирована на компьютере, информация о совместимости будет использоваться каждый раз при запуске приложения.

1. Охарактеризуйте разработку модулей обеспечения совместимости
2. Опишите процесс создания в системе виртуальной машины для исполнения приложений.

## Режим виртуальных машин для исполнения приложений реального режима

Разработчики рассматриваемого семейства микропроцессоров в своем стремле­нии обеспечить максимально возможную совместимость архитектуры пошли не только на то, чтобы обеспечить возможность программам, созданным для первых 16-разрядных ПК, без проблем выполняться на компьютерах с более поздними моделями микропроцессоров за счёт введения реального режима работы. Они также обеспечили возможность выполнения 16-разрядных приложений реального режима при условии, что сам процессор при этом функционирует в защищённом режиме работы и операционная система, используя соответствующие аппаратные средства микропроцессора, организует мультипрограммный (мультизадачный) режим. Другими словами, микропроцессоры i80x86 поддерживают возможность создания операционных сред реального режима при работе микропроцессора в защищённом режиме. Если условно назвать 16-разрядные приложения DOS-приложениями (поскольку в абсолютном большинстве случаев это именно так), то можно сказать, что введена поддержка для организации виртуальных DOS-машин, работающих вместе с обычными 32-битовыми приложениями защищён­ного режима. Это даже нашло отражение в названии режима работы микропро­цессоров i80x86 – режим виртуального процессора i8086, иногда (для краткости) его называют режимом V86 или просто виртуальным режимом, – при котором в защищённом режиме работы может исполняться код DOS-приложения. Мультизадачность при выполнении нескольких программ реального режима будет поддержана аппаратными средствами защищённого режима.

Переход в виртуальный режим осуществляется посредством изменения бита VM(virtualmode) в регистре EFLAGS. Когда процессор находится в виртуальном режиме, для адресации памяти используется схема реального режима работы – (сегмент: смещение) с размером сегментов до 64 Кбайт, которые могут распола­гаться в адресном пространстве размером в 1 Мбайт, однако полученные адреса считаются не физическими, а линейными. В результате применения страничной трансляции осуществляется отображение виртуального адресного пространства 16-битового приложения на физическое адресное пространство. Это позволяет организовать параллельное выполнение нескольких задач, разработанных для реального режима, да ещё и совместно с обычными 32-битовыми приложениями, требующих защищённого режима работы. Естественно, что для обработки прерываний, возникающих при выполнении 16-битовых приложений в виртуальном режиме, процессор возвращается из этого режима в обычный защищённый режим. В противном случае невозможно было бы организовать полноценную виртуальную машину. Очевидно, что обработчи­ки прерываний для виртуальной машины должны эмулировать работу подсисте­мы прерываний процессораi8086. Другими словами, прерывания отображаются в операционную систему, работающую в защищённом режиме, и уже основная ОС моделирует работу операционной среды выполняемого приложения.

Вопрос, связанный с операциями ввода/вывода, которые недоступны для обыч­ных приложений, решается аналогично. При попытке выполнить недопустимые команды ввода/вывода возникают прерывания, и не­обходимые операции выполняются операционной системой, хотя задача об этом и «не подозревает». При выполнении команд IN,OUT,INS,OUTS,CLI,STIпроцессор, находящийся в виртуальном режиме и исполняющий код на уровне привилегий третьего (самого нижнего) кольца защиты, за счёт возникающих вследствие этого прерываний переводится на выполнение высоко привилегированного кода опе­рационной системы.

Таким образом, ОС может полностью виртуализировать ресурсы компьютера: и аппаратные, и программные, создавая иную полноценную операционную сре­ду; при существовании так называемых нативных приложений, создаваемых по собственным спецификациям данной ОС. Очень важным моментом для органи­зации полноценной виртуальной машины является реализация виртуализации не только программных, но и аппаратных ресурсов. Так, например, в ОСWin­dowsNTэта задача выполнена явно неудачно, тогда как в OS/2 имеется полно­ценная виртуальная машина как для DOS-приложений, так и для приложений, работающих в среде спецификацийWin16. Правда, в последнее время это уже перестало быть актуальным, поскольку появилось большое количество приложе­ний, работающих по спецификациям Win32API.

## Создание виртуальной машины с помощью Hyper-V в Windows10

Создание виртуальной машины и установка на нее операционной системы. Вам потребуется ISO-файл для операционной системы, которую требуется запустить.

### Создание виртуальной машины с помощью диспетчера Hyper-V

1. Откройте диспетчер Hyper-V, нажав клавишу Windows и введя в строке поиска "диспетчер Hyper-V", выполните поиск **диспетчера Hyper-V** в своих приложениях.
2. В диспетчере Hyper-V щелкните **Действие** > **Создать** > **Виртуальная машина**, чтобы открыть мастер создания виртуальной машины.
3. Просмотрите содержимое страницы "Приступая к работе" и нажмите кнопку **Далее**.
4. Присвойте имя виртуальной машине.

**Примечание.** Это имя, используемое Hyper-V для виртуальной машины, а не имя компьютера, присвоенное гостевой операционной системе, которая будет развернута внутри виртуальной машины.

1. Выберите расположение, где будут храниться файлы виртуальной машины (например, **c:\virtualmachine**). Можно также оставить расположение по умолчанию. По завершении нажмите кнопку **Далее**.
2. Выберите поколение виртуальной машины и нажмите кнопку **Далее**.

В виртуальных машинах поколения 2, которые появились в Windows Server 2012 R2, доступна упрощенная модель виртуального оборудования и некоторые дополнительные функциональные возможности. На виртуальную машину поколения 2 можно установить только 64-разрядную версию операционной системы.

Если новая виртуальная машина относится к поколению 2 и будет работать под управлением дистрибутива Linux, потребуется отключить безопасную загрузку.

1. Выберите **2048**МБ в качестве значения параметра **Память, выделяемая при запуске** и не снимайте флажок **Использовать динамическую память**. Нажмите кнопку **Далее**.

Память распределяется между узлом Hyper-V и виртуальной машиной, работающей на этом узле. Число виртуальных машин, которые могут работать на одном узле, зависит, в частности, от объема доступной памяти. Виртуальную машину можно также настроить для использования динамической памяти. При включении динамическая память высвобождает неиспользуемую память работающей виртуальной машины. Это позволяет большему количеству виртуальных машин работать на узле.

1. В мастере настройки сети выберите для данной виртуальной машины виртуальный коммутатор и нажмите кнопку **Далее**.
2. Присвойте имя виртуальному жесткому диску, выберите расположение или оставьте значение по умолчанию, а затем укажите размер. По окончании нажмите кнопку **Далее**.

Виртуальный жесткий диск представляет собой память для виртуальной машины, чем напоминает физический жесткий диск. Виртуальный жесткий диск требуется для установки операционной системы на виртуальной машине.

1. В мастере параметров установки выберите элемент **Установить операционную систему из файла загрузочного образа**, а затем выберите ISO-файл нужной операционной системы. Нажмите кнопку **Далее** после завершения.

При создании виртуальной машины можно настроить некоторые параметры установки операционной системы. Доступны три варианта.

* + **Установить операционную систему позднее**. Этот параметр не вносит какие-либо дополнительные изменения в виртуальную машину.
  + **Установить операционную систему из файла загрузочного образа**. Действие этого параметра аналогично вставке компакт-диска в физический дисковод компьютера. Чтобы настроить этот параметр, выберите ISO-образ. Этот образ будет подключен к дисководу виртуальной машины. Меняется порядок загрузки виртуальной машины: загрузка начинается с дисковода.
  + **Установить операционную систему с сетевого сервера установки**. Этот параметр доступен только тогда, когда виртуальная машина подключена к сетевому коммутатору. В этой конфигурации виртуальная машина предпринимает попытки загрузиться из сети.

1. Просмотрите информацию о данной виртуальной машине и нажмите кнопку **Готово** для завершения создания виртуальной машины.

### Создание виртуальной машины с помощью PowerShell

1. Откройте интегрированную среду сценариев PowerShell от имени администратора.
2. Выполните следующий скрипт.
3. # Set VM Name, Switch Name, and Installation Media Path.
4. $VMName = 'TESTVM'
5. $Switch = 'External VM Switch'
6. $InstallMedia = 'C:\Users\Administrator\Desktop\en\_windows\_10\_enterprise\_x64\_dvd\_6851151.iso'
7. # Create New Virtual Machine
8. New-VM -Name $VMName -MemoryStartupBytes 2147483648 -Generation 2 -NewVHDPath "D:\Virtual Machines\$VMName\$VMName.vhdx" -NewVHDSizeBytes 53687091200 -Path "D:\Virtual Machines\$VMName" -SwitchName $Switch
9. # Add DVD Drive to Virtual Machine
10. Add-VMScsiController -VMName $VMName
11. Add-VMDvdDrive -VMName $VMName -ControllerNumber 1 -ControllerLocation 0 -Path $InstallMedia
12. # Mount Installation Media
13. $DVDDrive = Get-VMDvdDrive -VMName $VMName
14. # Configure Virtual Machine to Boot from DVD
15. Set-VMFirmware -VMName $VMName -FirstBootDevice $DVDDrive

### Завершение развертывания операционной системы

Чтобы завершить создание виртуальной машины, необходимо ее запустить и выполнить пошаговые инструкции по установке операционной системы.

1. В диспетчере Hyper-V дважды щелкните виртуальную машину. Запустится средство VMConnect.
2. В VMConnect нажмите зеленую кнопку "Пуск". Это аналогично нажатию кнопки питания на физическом компьютере. Нажмите любую клавишу для загрузки с компакт- или DVD-диска при появлении соответствующего запроса.

**Примечание.** Чтобы убедиться, что нажатия клавиш отправляются в виртуальную машину, щелкните внутри окна VMConnect.

1. Виртуальная машина загрузится для установки, и можно будет выполнить пошаговую установку аналогично процедуре на физическом компьютере.

**Примечание.** Если вы не используете корпоративную лицензию на Windows, требуется отдельная лицензия для Windows, установленная в виртуальной машине. Операционная система виртуальной машины не зависит от операционной системы узла.

1. Опишите процесс тестирования на совместимость в безопасном режиме.

## Безопасный режим в Windows

Безопасный режим — это режим запуска системы Windows, предназначенный для устранения неполадок, в котором используется ограниченный набор служб и компонентов. Загружаются только базовые файлы и драйверы, необходимые для запуска Windows. В углах экрана отображаются слова Безопасный режим, соответствующие используемому режиму Windows.

Если при загрузке в безопасном режиме существующие проблемы не возникают, то из списка возможных причин можно исключить параметры, используемые по умолчанию, и базовый набор драйверов устройств. Если причина проблемы неизвестна, можно воспользоваться методом исключения, чтобы ее обнаружить. Попробуйте поочередно осуществить запуск всех обычно используемых программ, включая программы из папки «Автозагрузка», чтобы увидеть, какая из программ может приводить к возникновению проблемы.

Если компьютер автоматически, без запроса, запускается в безопасном режиме, возможно, проблема препятствует обычной загрузке Windows.

**Запуск компьютера в безопасном режиме**

В безопасном режиме запуск Windows выполняется с использованием ограниченного набора файлов и драйверов. В безопасном режиме не производится запуск программ, загружаемых при запуске, и устанавливается только базовый набор драйверов, необходимый для запуска Windows.

Безопасный режим полезен для устранения неполадок, вызванных программами и драйверами, которые могут неправильно загружаться или препятствовать правильному запуску Windows. Если при загрузке в безопасном режиме эти проблемы не возникают, то из списка возможных причин можно исключить параметры, используемые по умолчанию, и базовый набор драйверов устройств. Если недавно установленная программа, устройство или драйвер препятствуют правильному запуску Windows, можно запустить компьютер в безопасном режиме и удалить программу, вызывающую проблему. Дополнительные сведения об устранении неполадок в безопасном режиме см. в разделеСредства диагностики, которые следует использовать в безопасном режиме.

1. Извлеките из компьютера все гибкие диски, компакт-диски и DVD-диски, а затем перезагрузите компьютер.

Нажмите кнопку Пуск, щелкните стрелку рядом с кнопкой Завершение работы, затем нажмите кнопку Перезагрузка.

1. Выполните одно из следующих действий.
   * Если установлена одна операционная система, во время перезагрузки компьютера нажмите и удерживайте клавишу F8. Клавишу F8 необходимо нажать до того, как появится логотип Windows. Если появилась эмблема Windows, дождитесь появления приглашения Windows для входа в систему, затем завершите работу и перезагрузите компьютер.
   * Если на компьютере установлено несколько операционных систем, при помощи клавиш со стрелками выберите операционную систему, которую необходимо запустить в безопасном режиме, и затем нажмите клавишу F8.
2. С помощью клавиш со стрелками выберите на экране Дополнительные варианты загрузки необходимый вариант безопасного режима и нажмите клавишу ВВОД.
3. Войдите в систему под учетной записью пользователя с правами администратора.

Когда компьютер находится в безопасном режиме, по углам экрана отображается надпись Безопасный режим. Чтобы выйти из безопасного режима, перезагрузите компьютер и позвольте Windows запуститься в обычном режиме.

**Средства диагностики, которые следует использовать в безопасном режиме**

При использовании безопасного режима для устранения неполадок компьютера значительную помощь могут оказать описанные в данном разделе средства и функции.

* Откройте компонент **«Восстановление»**. Для этого нажмите кнопку Пуск  и выберите пункт Панель управления. В поле поиска введите восстановление и затем щелкните пункт Восстановление.

С помощью компонента «Восстановление» можно восстановить состояние системы до состояния на определенный момент времени, а также восстановить содержимое жесткого диска из архива. Также можно переустановить Windows из образа для восстановления, расположенного на жестком диске компьютера.

* Откройте панель управления. Для этого нажмите кнопку Пуск и выберите пункт Панель управления.

Панель управления служит для доступа к различным средствам изменения параметров Windows.

* Откройте диспетчер устройств. Для этого нажмите кнопку Пуск, выберите последовательно компоненты Панель управления, Система и безопасность, а затем в разделе Система выберите Диспетчер устройств.‌ Если отображается запрос на ввод пароля администратора или его подтверждения, укажите пароль или предоставьте подтверждение.

Диспетчер устройств позволяет обновлять драйверы устройства и для настраивать установленное оборудование. Для использования этой программы необходимо войти в систему в качестве администратора. В противном случае изменять можно только параметры текущей учетной записи пользователя.

* Откройте раздел «Просмотр событий». Для этого нажмите кнопку Пуск, выберите Панель управления, Система и безопасность, Администрирование, затем дважды щелкните Просмотр событий. Если отображается запрос на ввод пароля администратора или его подтверждения, укажите пароль или предоставьте подтверждение.

Просмотр событий служит для просмотра подробных записей о событиях системы и программ на вашем компьютере. Для использования этой программы необходимо войти в систему в качестве администратора. В противном случае изменять можно только параметры текущей учетной записи пользователя.

* Откройте раздел «Сведения о системе». Для этого нажмите кнопку Пуск. В поле поиска введите Сведения о системе, а затем в списке результатов выберите пункт Сведения о системе.

Компонент «Сведения о системе» служит для просмотра подробных сведений о конфигурации оборудования, компонентах и драйверах компьютера.

* Откройте окно командной строки. Для этого нажмите кнопку Пуск. В поле поиска введите Командная строка, а затем в списке результатов выберите пункт Командная строка.

Опытные пользователи могут использовать окно командной строки для вызова программ командной строки. Для использования этой программы необходимо войти в систему в качестве администратора. В противном случае изменять можно только параметры текущей учетной записи пользователя.

* Откройте редактор реестра. Для этого нажмите кнопку Пуск, введите regedit в поле поиска и затем нажмите клавишу ВВОД.‌ >Если отображается запрос на ввод пароля администратора или его подтверждения, укажите пароль или предоставьте подтверждение.

Опытные пользователи могут использовать редактор реестра, чтобы вносить изменения в файлы реестра Windows. Для использования этой программы необходимо войти в систему в качестве администратора. В противном случае изменение параметров будет доступно только для вашей учетной записи пользователя.

В безопасном режиме Windows запускается только с базовым набором драйверов и служб, перечисленных в данном разделе.

## Восстановление системы.

Иногда из-за установки программы или драйвера Windows может начать работать медленно или непредсказуемо. Функция восстановления системы позволяет восстановить системные файлы и программы компьютера.

Восстановление системы использует точки восстановления для возврата системных файлов и параметров к состоянию на определенный момент времени, не влияя на личные файлы. Точки восстановления автоматически создаются еженедельно, а также перед значительными системными событиями, такими как установка программ или драйверов устройств. Точку восстановления также можно создать вручную.

В Windows 7 можно не только создавать дополнительные точки восстановления системы, но и просматривать, какие именно файлы будут удалены или добавлены при восстановлении системы.

Прежде чем запустить восстановление системы, сохраните все открытые файлы и закройте все программы. В процессе восстановления системы будет перезагружен компьютер.

■ Откройте восстановление системы, нажав кнопку Пуск. В поле поиска введите восстановление системы, а затем в списке результатов выберите Восстановление системы. Введите пароль администратора или подтверждение пароля, если отобразится соответствующий запрос.

■ Выполните указания мастера, чтобы выбрать точку восстановления и восстановить компьютер.

1. Опишите процесс восстановления системы.

## Восстановление системы Windows 7

Для возобновления работы операционной системы можно прибегнуть к одному из нескольких методов. Часть вариантов основана на откате системы к сохраненной версии. Другая часть способов – стирает данные.

Реанимировать операционную системы можно через:

* выбор точек возобновления;
* командную строку;
* безопасный режим;
* использование среды возобновления.

Восстановление Windows 7 подразумевает задействование контрольных точек, которые реанимируют систему. Это вариант популярен и эффективен.

Для его использования потребуется:

* Воспользоваться меню «Пуск»
* Найти раздел «Восстановление системы»:
* Нажать кнопку «Далее»
* Выбрать точку возобновления:
* Подтвердить свое решение, нажав «Готово»:

Выполнение этих шагов позволяет устранять неполадки в компьютере, отменять изменения и возвращать рабочее состояние операционной системы. При использовании этого метода документы, файлы и прочая информация не стираются.

Операцию можно отменить и вернуть все в первоначальное состояние. А затем, снова пройдя эти шаги, выбрать другую точку для того, чтобы восстановить Windows 7.

Для самостоятельного создания точки возобновления необходимо нажать «Пуск», выбрать «Восстановление системы». Нужную версию можно сохранить в любое время. Файл помечается определенной датой, которую стоит запомнить.

**С точки восстановления**

Точкой восстановления называются сохраненные параметры электронного устройства. Обычно с каждой удачной загрузкой операционной системы, сохранение производится в автоматическом режиме. Использование этих параметров позволяет оперативно восстанавливать систему 7-ой версии.

* Перезагрузить компьютер, в процессе нажать клавишу F8
* Появятся варианты запуска операционной системы
* Выбрать максимально удачный вариант:

**Без точек восстановления**

Если точки восстановления отсутствуют, можно воспользоваться программой LiveCD. Утилиту скачивают с официального сайта и сохраняют на внешнем носителе, который имеет расширение .iso. В БИОС потребуется произвести настройку загрузки со съемного накопителя. Для этого в строчке «First boot device» раздела Boot необходимо выбрать USB-HDD.

* Приступаем к восстановительной работе
* Копируем на съемную флешку всю нужную информацию
* Можно воспользоваться специальным меню в программе LiveCD
* Архивная копия используется для исправления системных ошибок
* Теперь вставляем в адаптер накопитель с утилитой, открываем Windows\System32\config\
* Находим файлы, которые именуются «default», «sam», «security», «software», «system»:
* Переносим в любую папку на компьютере
* Открываем RegBack и из нее собираем файлы с идентичными названиями
* Осуществляем перезагрузку электронного устройства.

Воспользоваться этим методом стоит при проблемах с реестром.

**Командная строка**

Если операционная система работает медленно, но при этом загружается можно исправить системную ошибку с помощью командной строки.

* Нажимаем «Пуск», печатаем «cmd»
* Щелкаем правой кнопкой по программе
* Находим «Запуск от имени администратора»:
* Прописываем команду rstrui.exe, она открывает утилиту, что восстановит ОС
* Жмем «Далее», выбираем точку, на которую желаем перейти, опять «Далее»:
* Как только процесс завершится, электронное устройство будет работать исправно.

Второй способ перейти в программу заключается в вызове пункта меню «Выполнить», он размещается в «Пуске». Здесь потребуется прописать «CMD». Нажать мышью на файле с названием CMD.exe, запустить его. Ввести «rstrui.exe», подтвердить свое решение, нажав Enter.

Есть еще один легкий способ, как восстановить Windows. Несколько простых шагов позволяют исправить системную ошибку:

* Открываем «Мой компьютер» – «Свойства» – «Защита системы»:
* Найти и нажать кнопку восстановления и «Далее»
* Находим точку восстановления с определенной датой
* Обозначаем диски, которые требуется проверить
* Дать согласие на выполнение операции и перезагрузить электронное устройство.

**Восстановление Windows 7 посредством безопасного режима**

Этот способ применим при невозможности загрузить систему в привычном режиме. Требуется выполнить следующие шаги:

* Включить электронное устройство
* Удерживать F8, с помощью этой клавиши запустится «Меню запуска»
* Потребуется найти и выбрать «Безопасный режим»:
* После загрузки операционной системы стоит повторить действия, которые проделывались выше.

1. Раскрыть понятие производительности ПК. Опишите проблемы производительности.

Казалось бы, вполне шустрый компьютер, который вначале полностью удовлетворял своими возможностями, со временем начинает тормозить. Что могло послужить причиной такого снижения производительности? Или бывает еще другая ситуация, когда нужно работать в очень ресурсоемких программах (играх), а компьютер их просто не тянет. В любом случае, перед пользователем встает проблема производительности ПК, которую нужно решать.

### Основными признаками медленной работы компьютера являются:

1. Долгое включение и такое же долгое выключение компьютера.
2. Многие программы очень долго запускают, загрузка игр занимает очень много времени.
3. Реакция курсора может не успевать за движениями мышки, притормаживать, или наблюдается поздняя реакция на клик.
4. И другие признаки замедленной работы ПК, из-за которых пользователь начинает нервничать.

### Что может быть причиной торможения компьютера?

Если компьютер тормозит, то причина (или сразу несколько причин) могут быть следующие:

1. Плохая совместимость программного обеспечения, операционной системы, игр с конфигурацией «железа» самого компьютера.
2. Большая фрагментация файловой системы, на жестком диске скопилось много программного хлама.
3. Проблемы с операционной системой, сбои, неправильная настройка.
4. Большое количество запущенных служб, в которых нет необходимости.
5. Заражение компьютера вредоносным программным обеспечением.
6. Программные конфликты (между драйверами, антивирусами и др.).
7. Плохая терморегуляция, выход из строя кулера, засорение радиаторов.
8. Нарушение основных параметров в BIOS.

### Как решить эту проблему и повысить производительность компьютера?

Существует множество способов, позволяющих ускорить работу компьютера. Мы предлагаем десять методов улучшения работы ПК, показавших свою эффективность на практике.

1. Апгрейдим «железо».
2. Чистка внутренностей, ремонт неисправных элементов системы охлаждения.
3. Очищаем жесткий диск от всего ненужного, выполняем дефрагментацию.
4. Переустанавливаем операционную систему.
5. Оптимизируем работу ОС.
6. Настраиваем BIOS.
7. Отключаем все лишнее, контролируем автозагрузку.
8. Выполняем обновление драйверов.
9. Устанавливаем операционную систему соответствующую нашему ПК.
10. Чистим систему от вирусов.

Далее, для лучшего понимания, мы дадим более подробное описание каждого из этих пунктов. Но для начала, нам необходимо узнать производительность своего компьютера.

### Определяем производительность

В операционной системе Windows 7 уже встроена специальная функция для проверки производительности системы и быстродействия компьютера.

Откройте меню «Пуск» и в специальном поисковом поле (оно находится внизу меню) введите «произ», как это показано на рисунке.

Среди результатов поиска будет программа «Повышение производительности компьютера». Запустите ее и откроется следующие окно.

Снизу сделайте клик по кнопке «Повторить оценку». Программа начнет анализировать продуктивность вашей системы и по завершению выдаст результаты, по которым можно будет сделать вывод о быстродействии компьютера.

После этого можно приступать к повышению производительности компьютера выбранными способами.

### Апгрейдим комплектующие компьютера

У многих, кто является «счастливым» обладателем тормозящего компьютера, возникает стойкое желание заменить его чем-нибудь по мощнее. Но мы не станем двигаться в этом направлении, оно подходит только для серьезно устаревающих компьютеров. В большинстве случаев достаточно заменить только один или несколько слабых элементов, чтобы вся система снова начала летать — этот способ будет намного дешевле, чем покупка нового ПК.

* **Центральный процессор**. Смысл замены «камня» будет действительно оправдан, только если новый намного превзойдет старого по мощности, не менее чем на 30%. Иначе вы не почувствуете серьезной разницы, но зато потратите много денег.

Любители риска могут попытаться разогнать центральный процессор. Этот способ подойдет не многим, ведь далеко не каждый ЦП может разгоняться. Зато в случае успеха, покупку нового «камня» можно отложить еще на год другой. Суть этой операции заключается в повышении напряжения на центральном процессоре, и тем самым увеличение его тактовой частоты. Сама операция достаточно рискованная и далеко не для всех процессоров, в некоторых случаях можно просто вывести из строя ЦП, или значительно сократить срок его службы.

* **Оперативная память**. Ее никогда не бывает много, можете смело увеличивать оперативную память. Тем более она стоит достаточно дешево. Когда компьютер сильно загружен, откройте диспетчер задач на вкладке быстродействия и посмотрите уровень используемой памяти. Если он превышает 80%, то можете увеличивать память в полтора два раза.
* **Винчестер**. И суть тут вовсе не в его объеме и количестве свободного места. Основная фишка быстродействия жесткого диска заключается в скорости вращения его мотора. К примеру, мы имеем диск со скоростью 5400 оборотов и заменяем него на диск со скоростью 7200 оборотов. Это намного более ощутимо добавит скорости для систем, особенно скорости записи/чтения.
* **Видеокарта**. Большинству понятно, что для более реалистичной графики в играх необходимо иметь мощный видеоадаптер. Так что, если ваша видеокарта не может похвастаться хорошей производительностью, меняйте на новую, которая гораздо превосходит ее по мощности. Главное – нужно учитывать, что для мощной видеокарты нужен и мощный процессор.

Чтобы приблизительно определить слабые места системы, воспользуйтесь встроенной функцией оценки производительности компьютера, о которой мы уже рассказывали выше. Определить «слабое звено» можно по оценкам, которые выводятся после проверки производительности. Там, где самая низкая оценка, в том направлении и необходимо увеличивать мощность ПК. К примеру, если возле скорости обмена данными на диске стоит самая низкая оценка, то стоит задуматься о покупке более шустрого жесткого диска.

### Чистка внутренностей, ремонт неисправных элементов системы охлаждения

Различные неисправности в системе охлаждения также способны существенно замедлить работу всей системы. Если ломается кулер на центральном процессоре, то это привод к его перегреву и снижению тактовой частоты.

Перегрев вообще опасная штука и возникать может даже при исправной системе охлаждения. Возьмите, снимите крышку со своего системного блока, посмотрите сколько там пыли. Она не просто покрывает все поверхности, но и плотно забивается во все радиаторы и оседает на лопастях кулеров. Пыль выступает таким себе теплоизолятором, который приводит к перегреву. Очень аккуратно почистите системный блок и вам удастся увеличить скорость работы компьютера, а также продлить срок службы его компонентов и избавиться от главного рассадника различных аллергенов и бактерий — пыли.

### Очищаем жесткий диск от всего ненужного, выполняем дефрагментацию

Те, кто впервые слышит термин «дефрагментация», должны понять, что это самое первое что нужно делать для повышения производительности системы. Дефрагментация позволяет собирать различные фрагменты программ, которые находятся в разных частях жесткого диска, в одну кучу. Благодаря этому, считывающему устройству винчестера не нужно выполнять много лишних перемещений по дискам, ведь все находится в одном месте. Таким образом и повышается производительность.

Кроме того, нужно избавиться от лишней информации и всякого программного хлама, который накапливается на диске со временем работы. Особенно это важно, когда в разделе операционной системы практически нет свободного места. Если места менее 2 Гб, система теряет свою производительность. Так что не перегружайте диск и старайтесь, чтобы на нем было процентов тридцать свободного места, если это конечно возможно.

#### Переустанавливаем**операционную систему**

Этот шаг почти всегда помогает увеличить скорость работы компьютера. В некоторых случаях, производительность может увеличиться в три раза. Просто такова суть операционной системы, со временем в ней накапливаются различные ошибки, она забивается ненужными службами, которые даже выполняют серьезные изменения в самой системе. Это и многое другое приводит к ухудшению скорости работы компьютера, на многие операции теперь требуется гораздо больше времени.

Если тщательно следить за чистотой системы и ничего туда не устанавливать, то можно годами пользоваться одной и той же Windows. Но чаще всего, на компьютере постоянно происходит движение: устанавливаются и удаляются программы, обновляются драйвера, загружаются большие объемы различной информации — в таких условиях система постепенно начинает «тупить». Лучше всего, для профилактики, где-то раз в год форматировать диск и с чистого листа устанавливать новую операционную систему.

#### Оптимизируем работу ОС

Хорошо себя показала программа PCMedic. Главная фишка этой утилиты, это полная автоматизация всех операций. Нужно только выбрать подходящие параметры и запустить процесс настройки.

Программа состоит только из одного главного окна. Тут мы выбираем свою операционную систему, тип центрального процессора (например, Intel или AMD), дальше нужно выбрать один из двух способов оптимизации — Heal (очистка системы), либо Heal & Boost (кроме очистки выполняется еще и ускорение). После того, как выберите все параметры нажмите на кнопочку «Go» – программа выполнит все необходимые изменения.

Есть еще одна программа, которая может похвастаться воистину мощным функционалом — Ausloqics BoostSpeed, к сожалению, платная. Она состоит из нескольких утилит, позволяющих выполнять оптимизацию системы практически во всех направлениях. Используя это приложение можно провести дефрагментацию, почистить файловую систему, очистить реестр, увеличить скорость работы интернета и еще много другого. Программа обладает встроенным советчиком, который помогает определить приоритетные направления в оптимизации системы. Хотя рекомендую слепо не доверяться советчику и смотреть, действительно ли эти все действия вам необходимы.

Для того, чтобы чистить систему от всякого хлама, существует множество специальных приложений для очистки. Например, хорошим помощником может оказаться Ccleaner. Он способен почистить диск от ненужных, временных фалов и выполнить очистку реестра. Благодаря удалению ненужных файлов, можно увеличить количество свободного места на жестком диске. А вот при чистке реестра, особого повышения производительности не наблюдается. Зато если будет случайно удален какой-нибудь важный параметр, система начнет выдавать ошибки и это может привести к серьезным сбоям.

ВНИМАНИЕ! Перед тем, как выполнять все эти действия, настоятельно рекомендуется создать точку восстановления!

ВСЕГДА смотрите файлы, которые удаляют утилиты для очистки системы. Бывают случаи безвозвратного удаления нужных и даже важных файлов, которые программы ложно приняли за ненужные, или временные файлы.

Пользователи Windows 7 могут немного улучшить скорость работы компьютера при помощи упрощения графического интерфейса. Что бы это сделать откройте Панель управления и зайдите в раздел «Система» откройте пункт «Дополнительно» и выберите «Параметры». Здесь снимаем часть ненужных флажков, либо устанавливаем переключатель в положение для обеспечения наилучшего быстродействия.

#### Настраиваем BIOS

В БИОСе хранятся параметры компьютера, отвечающие за оборудование, загрузку ОС, время и других ключевые элементы. Чтобы попасть в параметры BIOS, в процессе включения ПК, перед загрузкой самой операционной системы, нажмите клавишу Del, F2 или другую (это зависит от производителя материнской платы, обычно название клавиши отображается на экране). Чаще всего настройки БИОСа не вызывают никакого снижения производительности и туда даже не стоит лазить. Но в редких случаях, при неправильных критических параметрах, ваш компьютер может начать тормозить.

Если сомневаетесь в правильности настроек, но не знаете какие нужно указать, то можете воспользоваться опцией автоматической настройки оптимальных параметров «Load Optimal Settings» (название функции может быть иным, в зависимости от производителя). После этого сохраните все настройки и выполните перезагрузку компьютера.

#### Отключаем все лишнее, контролируем автозагрузку

В наше время практически каждая программа пытается прописать себя в автозагрузку. И это не слишком весело для вашей системы. Постепенно, программы в автозагрузке накапливаются и их всех надо запускать, при каждом старте системы. Из-за этого компьютер очень долго включается и выключается. Дополнительно, после автозагрузки, все эти приложения остаются в рабочем состоянии, пожирая ресурсы. Вы только взгляните сколько на панели задач (возле часов) ненужных значков, разве вы их используете? Поэтому, лучше всего удалять ненужные приложения, или хотя бы отключать для них автозагрузку.

Что бы посмотреть все приложения, которые стартуют вместе с Windows, и отключить лишние, одновременно зажмите две клавиши Win+R и в следующем окошке напишите msconfig, затем нажмите Enter. Появится окошко конфигурацией системы, теперь перейдите в раздел автозагрузки. Здесь будет находиться список приложений, стоящих в автозагрузке. Просто снимаете галочки со всех, которыми не пользуетесь. В случае чего, программу всегда можно вернуть в автозагрузку, установив нужную галочку. Главное иметь представление, что это за программы и какое их назначение.

В самых крайних случаях, можно отключить антивирус, и система немного разгрузится. За счет этого повысится производительность. Но этого не стоит делать, когда вы находитесь в интернете, или устанавливаете новые программы, потому что есть риск заражение вирусами.

#### Выполняем обновление драйверов

Этот шаг может дать нужных эффект, если установлены устаревшие драйвера или те что установились вместе с системой. Больше всего может влиять драйвер для материнской платы, хотя и другие неплохо воздействуют на скорость работы.

Надежнее всего вручную выполнять обновление драйверов. Но самые ленивые могут воспользоваться специальными утилитами, которые способны сами определять необходимые драйвера, находить их в интернете и устанавливать. Например, можно воспользоваться Drover Checker.

#### Устанавливаем операционную систему соответствующую нашему ПК

Если у вас компьютер, с 2 Гб оперативной памяти и вы используете Windows XP, то стоит задуматься о переходе на «семерку», после этого сразу почувствуете положительную разницу. А если на компьютере двухъядерный процессор и 4 Гб памяти (а может даже больше), то необходимо использовать 64х разрядную Windows 7.

И да, избегайте кустарных сборок Windows, таких как Zver и тому подобных. Используйте исключительно чистые, оригинальные дистрибутивы.

#### Чистим систему от вирусов

Не обращайте внимание на то, что этот пункт стоит под номером десять. На самом деле это одна из самых распространенных причин медленной работы компьютера. Если заметили, что компьютер внезапно начал долго «думать», срочно выполните сканирование на наличие вредоносного ПО. Лучше всего выполнить сканирование системы штатным антивирусом (например, Антивирусом Касперского), а затем просканировать дополнительными антивирусными утилитами, типа Dr. Web CureIt, AVZ и др.

1. Оптимизация использования процессора. Оптимизация использования памяти. Оптимизация использования жесткого диска. Оптимизация использования сети.

## Оптимизация использования оперативной памяти

Оптимизация использования памяти может преследовать две цели – увеличение объема памяти, доступной приложениям, и повышение быстродействия обращений к памяти. На старых машинах, когда объем установленной физической памяти не превышал единиц мегабайт, зачастую приходилось чем-то жертвовать; в современных системах достижение обеих целей уже не противоречит друг другу. Нет сомнений в том, что, чем больше установленный объем ОЗУ, тем лучше – компьютер будет мощнее: он позволит загружать приложения, особо «жадные» до памяти; увеличить число одновременно работающих приложений (в многозадачной ОС); ускорить работу. Однако для обеспечения возможности использования памяти в ряде случаев требуются некоторые дополнительные действия по конфигурированию компьютера.   
В системах с размером установленной памяти более 640 Кбайт возможны различные варианты использования последних 384 Кбайт из первого мегабайта физической памяти:

* память не используется;
* область (или часть ее) перемещается в конец дополнительной памяти;
* область (или часть ее) используется в качестве теневой (Shadow) памяти адаптеров и ROM BIOS.  
  Перемещение неиспользуемого остатка первого мегабайта в конец дополнительной памяти (разрешается установкой параметра Memory Relocation в CMOS Setup) возможно не всегда. Обычно такое перемещение становится невозможным, если хоть часть из этого кусочка используется в качестве теневой памяти.   
  Также перемещение может предлагаться лишь при небольших объемах установленной памяти, и на современных системных платах эта возможность почти не встречается. Поэтому не стоит удивляться сообщению об объеме памяти, обнаруженном тестом POST, в котором относительно установленного «не хватает» 384 Кбайт.   
  Иногда BIOS предлагает такое распределение памяти, при котором под стандартную память выделяются 512 Кбайт, а остальная память идет как расширенная. Пользу такого распределения оценить трудно. При этом вышесказанное про верхние 384 Кбайт становится справедливым для оставшихся 512 Кбайт, правда, на возможность их перемещения в конец дополнительной памяти ограничения будут мягче.   
  Острее всего проблема обеспечения доступности памяти стоит для приложений MS-DOS, исполняемых в среде этой системы. Обычно «битва» идет за килобайты стандартной памяти (conventional memory), доступной приложениям. Из 640 Кбайт после загрузки ОС и необходимых резидентных драйверов на долю приложений может остаться около 500 Кбайт, а то и меньше, что для ряда приложений неприемлемо. Несмотря на повсеместное внедрения ОС типа Windows 9х, интерес к запуску больших приложений MS-DOS (например, бухгалтерских программ, в том числе и принудительно распространяемых налоговыми органами, а также игр) сохраняется и поныне. Сообщение о нехватке памяти на компьютере с ОЗУ размером, например, 32 Мбайт для приложения, скромно просящего всего 590 Кбайт, неприятно удивляет неискушенных пользователей. Однако эта нехватка не фатальна, если правильно выбрать версию и способ загрузки ОС. Все программные настройки, влияющие на объем доступной памяти, кроются в файлах CONFIG.SYS и AUTOEXEC.BAT, размещенных в корневом каталоге загрузочного диска.   
  Понимая проблемы, возникающие в связи с использованием памяти различными процессорами, для DOS-приложений можно рекомендовать следующие версии MS-DOS, оптимальные по размеру свободной стандартной памяти:
* для компьютеров класса XT на процессорах 8086/88 – MS-DOS 3.30;
* для компьютеров класса AT на процессорах 80286 – MS-DOS 5.x;
* для компьютеров класса AT на процессорах 80386 и выше – MS-DOS 6.2x (правда, одно время встречались некачественные системные платы для 80386, на которых MS-DOS 6.2x отказывалась загружаться в НМА, хотя MS-DOS 5.x работала нормально).  
  Кроме MS-DOS, существуют и другие операционные системы реального режима, совместимые с MS-DOS (например, PC DOS, DR DOS, COMPAQ DOS). Каждая из них имеет свои преимущества и недостатки по сравнению с соответствующим поколением MS-DOS, но их обсуждение выходит за рамки данного пособия. Операционные системы (и оболочки) защищенного режима на компьютеры с процессором класса ниже 386 пытаться устанавливать почти бессмысленно. Система MS-DOS 3.3 (более ранние рассматривать не будем) загружалась целиком в стандартную память, но была довольно компактной и оставляла приемлемое место для приложений своего времени. Когда вышла MS-DOS версии 4 с более развитыми возможностями, появившимися ценой ее разрастания, многие приложения отказывались работать в ее среде именно из-за нехватки памяти, и эта версия ОС широкого распространения не получила. Более удачной стала версия 5, которая «научилась» использовать «высокую память» (НМЛ) на компьютерах с процессором 286 и выше, если таковая присутствовала. Для этого был введен специальный драйвер высокой памяти, и в файле CONFIG.SYS должны присутствовать строки:   
  DEVICE=[<path>]HIMEM.SYS (загрузка драйвера высокой памяти)   
  DOS=HIGH (указание на загрузку ОС в высокую память)   
  Конечно, в верхнюю память загружается не вся ОС – часть все-таки попадает и в стандартную память. Кроме того, в стандартную память загружаются и резидентные драйверы – например, русификаторы клавиатуры и экрана, драйвер мыши и т. п. Все они отрывают свой кусок от памяти, которую могли бы использовать приложения. На компьютерах с 32-разрядными процессорами (386 и выше), имеющих механизм страничной переадресации, появилась возможность использования «верхней памяти» (UMA) с помощью драйвера EMM386.EXE. Этот драйвер отыскивает в области UMA (A0000-FFFFFh) регионы, не занятые памятью устройств, и отображает их на области доступной дополнительной памяти. В эти регионы, нормально адресуемые процессором в реальном режиме, можно помещать модули операционной системы и загружаемые драйверы; их же могут использовать и приложения.   
  Для наиболее компактной загрузки MS-DOS версий 5 и выше (а также Windows 9х, которая может представляться как MS-DOS 7) в файле CONFIG.SYS должны быть следующие директивы:   
  DEVICE=[<path>]HIMEM.SYS   (загрузка драйвера  высокой  памяти);   
  DEVICE=[<path>]EMM386.EXE   (загрузка диспетчера  расширенной  памяти);   
  DOS=HIGH.   UMB   (указание  на  загрузку ОС в высокую и  верхнюю память).   
  Резидентные драйверы (русификаторы клавиатуры, экрана и принтера, драйвер мыши) по умолчанию обычно загружаются в стандартную память. Если приложениям MS-DOS не хватает свободной памяти, то, по крайней мере, часть резидентных драйверов можно загрузить в верхнюю память UMA. Для драйверов, загружаемых из файла CONFIG.SYS, вместо команды DEVICE» следует использовать команду DEVICEHIGH [/L:n[,m]]=, которая попытается загрузить драйвер в n-регион UMB. Необязательный параметр m задает требуемый объем памяти (он может отличаться от размера файла с драйвером). Ключ загрузки /L вместе с параметрами n, m используется для ручной оптимизации памяти. Если драйверу требуется выделить несколько областей памяти, то они перечисляются в списке вида /L:nl[,ml]:n2[,m2][;...]. Для резидентных программ, загружаемых из файла AUTOEXEC.BAT, тем же целям служит команда LOADHIGH (LH) с аналогичными необязательными параметрами, задающими размер одной или нескольких областей UMA. Строка запуска будет иметь вид:   
  LH[/L:n[,m]]   <путь\><файл>[<параметры>]   
  Посмотреть текущее положение свободных блоков памяти можно командой MEM/F из командной строки DOS/Windows. Чтобы не заниматься ручной оптимизацией памяти, в состав DOS/Windows введена диалоговая утилита MEMMAKER.EXE, которая за несколько этапов расставит требуемые команды в файлах CONFIG.SYS и AUTOEXEC.BAT. До ее запуска в этих файлах должны присутствовать ссылки на все необходимые драйверы и программы. Файл CONFIG.SYS должен начинаться с вышеприведенных трех строк (если не указать DOS=HIGH, утилита может «постесняться» использовать высокую память для загрузки DOS). Запуск нерезидентных приложений, а также оболочки типа Norton Commander из файла AUTOEXEC.BAT на время работы MEMMAKER следует отменить (MEMMAKER несколько раз перезагружает компьютер и автоматически запускается после отработки файла AUTOEXEC.BAT). При начальном запуске MEMMAKER спрашивает, требуется ли память EMS для приложений, и, если нет, то установит ключ NOEMS в строке запуска EMM386.   
  Для того чтобы в верхнюю память поместилось как можно больше модулей, ее следует оптимизировать. Оптимизация UMA подразумевает такое конфигурирование базовых адресов буферной и постоянной памяти адаптеров, при котором свободные области UMA получались бы максимально возможного размера. Нужно стремиться к плотному соединению областей UMA, занимаемых адаптерами. Тогда максимальный размер блоков UMB увеличится, и драйвер типа EMM386 сможет разместить в нем более крупные модули, что, в свою очередь, позволит освободить дополнительное место в остродефицитной стандартной памяти. Если на конфигурирование адаптеров не обращать внимания, то может оказаться, что два компьютера с одинаковым составом аппаратных средств и программным обеспечением будут после загрузки ОС иметь значительно различающиеся размеры стандартной памяти.   
  Многие адаптеры (контроллеры SCSI, адаптеры локальных сетей и др.) позволяют задавать адреса областей встроенных RAM и ROM, отображаемых в пространство памяти компьютера. Это позволяет разрешать (или, наоборот, создавать) конфликты использования UMA, а также оптимизировать использование ее блоков. При конфигурировании устанавливаемых адаптеров необходимо исключить перекрытия занимаемых адресов, поскольку из-за этого, скорее всего, ни одно из конфликтующих устройств работать не будет. Самый неприятный (трудно устранимый) конфликт возникает, если при конфигурировании какого-либо адаптера его память перекрывает буфер или BIOS графического адаптера. Если конфигурирование осуществляется джамперами, то вернуть нормальную конфигурацию труда не составит. А если адаптер конфигурируется только с помощью специальной утилиты, изменяющей содержимое его энергонезависимой памяти, то изменить его конфигурацию можно, лишь загрузив и выполнив соответствующую утилиту. Но при конфликте с графическим адаптером это сделать не так-то просто – в лучшем случае удастся загрузить компьютер со «слепым» экраном, а в худшем – POST откажется продолжать тестирование и загрузку, обнаружив ошибку графического адаптера и сообщив об этом попискиванием динамика. Но эта ситуация не так безнадежна: есть еще в природе графический адаптер MDA, у которого видеобуфер не совпадает по адресам сраспространенными адаптерами EGA и VGA, а расширение BIOS у него отсутствует. Вставив неудачно сконфигурированную плату в компьютер с адаптером MDA (и, конечно, соответствующим монитором), можно утилитой задать правильную конфигурацию. Однако этот способ для новых компьютеров, не имеющих слотов шины ISA, уже непригоден.   
  Кроме занимаемого пространства, оптимизация использования UMA касается, как ни странно, и быстродействия. Для областей памяти адаптеров часто бывает полезным применение теневой памяти. Управление теневой памятью осуществляется через CMOS Setup для определенных областей. Поэтому, размещая конфигурируемые области памяти, иногда следует учитывать возможности задания границ теневой памяти в CMOS Setup. Необходимо помнить, что Shadow ROM блокирует запись, a Shadow RAM игнорирует возможность изменения со стороны адаптера затеняемой области памяти, что приведет к ошибкам при некорректном использовании теневой памяти. Для процессоров 386 и выше теневую память может организовать и драйвер EMM386, но этой его функцией пользуются редко.   
  Работу приложений в среде MS-DOS, в которой активно используется программный код BIOS, значительно ускоряет затенение ROM BIOS, как системной, так и BIOS графического адаптера и дискового контроллера. Для многозадачных ОС защищенного режима (Windows и др.) затенение ROM BIOS ускоряет только начальный процесс загрузки ОС, поскольку в рабочем режиме здесь в основном используются драйверы, загружаемые в ОЗУ.   
  Приложения ОС Windows пользуются виртуальной памятью,и никакие старые спецификации EMS и XMS им не нужны. Суммарный объем виртуальной памяти, доступной всем приложениям, определяется размером ОЗУ и файлов подкачки (их может быть и несколько). В Windows 9x размер файла подкачки изменяется динамически, по мере потребностей системы. Для того чтобы приложениям хватало памяти, на диске, несущем файл подкачки, должно быть достаточно свободного пространства (десятки и сотни мегабайт). Конечно же, важен и объем установленной физической памяти – ее малый объем может быть принципиальным ограничением на запуск ряда приложений или установку операционных систем. При малом объеме ОЗУ свопинг (подкачка страниц) будет слишком интенсивным, в результате чего скорость работы приложений существенно снижается (обращения к диску выполняются на несколько порядков медленнее, чем к ОЗУ). Приложения реального времени (например, аудио- и видеопроигрыватели и тем более кодеры) могут стать неработоспособными именно из-за малого объема ОЗУ. Поскольку файл подкачки изменяет свой размер в процессе работы, важно следить за фрагментацией диска, несущего этот файл – обращение к фрагментированному файлу выполняется медленнее, чем к нефрагментированному. При выборе диска для размещения файла подкачки следует учитывать его быстродействие – время доступа и скорость передачи данных. При использовании приложений реального времени, интенсивно обменивающихся с дисками (те же проигрыватели и кодеры, а также программы, записывающие компакт-диски), по возможности файл подкачки следует размещать на других дисках.   
  Если на компьютере под управлением ОС защищенного режима (Windows, Unix, OS/2...) перестают запускаться приложения с сообщениями о недостаточном объеме оперативной памяти – проверьте наличие свободного места на жестких дисках, используемых для подкачки. Если памяти не хватает приложениям MS-DOS – проверьте файлы AUTOEXEC.BAT и CONFIG.SYS и запустите утилиту MEMMAKER или выполните ручную оптимизацию памяти.   
  Увеличение физического объема оперативной памяти в ряде случаев может привести к неожиданному снижению производительности компьютера. Это возможно, когда системная плата (или процессор со вторичным кэшем) не способна кэшировать весь объем ОЗУ. У многих системных плат для процессоров Pentium кэшируются только первые 64 Мбайт ОЗУ; у первых процессоров Pentium II кэшировались только 512 Мбайт. Память, выходящая за размеры кэшируемой области, конечно же, доступна, но ее производительность гораздо ниже кэшируемой. ОС Windows 9x распределяет память, начиная с верхней границы доступной памяти, причем наверх попадает ее ядро, скорость работы которого существенна для работы многих приложений. Если после увеличения ОЗУ ядро попадает в некэшируемую область, можно наблюдать снижение производительности. Для лечения этого недуга можно воспользоваться условно-бесплатной программой W2CACHE.COM, которая запускается в начале загрузки Windows и, оставаясь резидентной, «съедает» верхнюю часть памяти, заставляя ядро Windows загружаться в нижнюю, кэшируемую область. После окончания загрузки Windows программа освобождает занимаемую память, и ОС отдает ее в распоряжение приложений.

## Оптимизация и ускорение работы жесткого диска

Наверно все пользователи хотят, чтобы их компьютеры могли выполнять требуемые действия намного быстрее, чем в текущий момент. Процесс обработки информации можно ускорить разными способами. К подобным вариантам относится оптимизация работы винчестера. Для подобного процесса в Виндовс имеется огромное количество настроек и инструментов. Давайте рассмотрим, как ускорить работу жесткого диска и в целом работу компьютера.

### Отключение индексирования

Данный способ подойдет для тех, кто не использует функцию поиска в операционной системе.

Основой метода является закрытие раздела винчестера от индексирования, что обеспечит ускорение его работы. Для этого:

1. Переходим в «Мой компьютер». Там нужно вызвать контекстное меню на главном диске, зачастую это диск С:\. Затем нажимаем «Свойства».
2. В меню «Общие» выбираем пункт «Разрешить индексировать содержимое…». Нужно снять галочку и подтвердить действие.
3. После данных действий появится оповещение, где будет сообщаться про возможность использования новой политики только к определенным файлам. Выбираем «Продолжить», а затем «Пропустить все».

### Отказываемся от ведения журнала

В стандартных настройках в системе имеется функция ведения журнала. Там отображаются все сбои и произведенные изменения. Если данную систему Вы не используете или даже не понимаете, что это значит, то можете смело ее отключить.

Инструкция по отключению:

1. Запуск командной строки. Наиболее простейший способ: нажатие «Win+R», а в появившемся окне «Выполнить» ввести «cmd».
2. Откроется приложение, где требуется ввести: fsutil usn deletejournal /D С:, где С – это буква необходимого жесткого диска.
3. Для подтверждения своих намерений нужно нажать на «Enter». Затем командную строку необходимо закрыть.
4. Для вступления в силу изменений, компьютер необходимо перезагрузить.

### Очистка от ненужных файлов

Довольно часто показатели скорости работы жестких дисков заметно улучшаются после удаления ненужных файлов, которые за длительное время могли собраться на накопителе. Чем больше свободного пространства на винчестере, тем быстрее будет происходить поиск файлов, а это ускорит работу.

Если этот процесс рассмотреть более подробно, то получается, что система не будет производить долгий поиск в своих таблицах адреса определенных единиц информации. При удалении не нужных данных количество таких таблиц уменьшится, чем при заполненном пространстве.

Поэтому нужно чаще удалять файлы с компьютера, которые не используются. Если они важные, то можно просто их собрать в одну папку и заархивировать. Подобная процедура освободит место, а в параметрах основной таблицы будет только один адрес.

Чтобы очистить диски в Виндовс 7 и иных версиях, используются специальные инструменты. Они располагаются в меню «Пуск» в категории служебных программ. Точное название – «Очистка диска». Программа производит поиск неиспользуемых файлов в разделе, а затем предлагает пользователю их удалить.

Кроме этого существует большое количество программ, обеспечивающих проведение данной процедуры.

### Дефрагментация

Когда производится запись новой информации на устройство, она не всегда размещается рядом. Размещение отдельных пакетов производится по свободному месту тома. После проведения многочисленных записей, одни файл может быть расположен по всему носителю. Это приводит к тому, что система постоянно обращается к разным секторам. Используя дефрагментацию, можно искать подобные данные и собирать их вместе. Затем система будет обращаться к соседним местам, что увеличит скорость работы.

Подобный инструмент способен помочь только при использовании HDD дисков.

Функция имеется в различных программах, которые предлагают сервисы по ускорению работы компьютера.

Операционные системы от Microsoft имеют возможность проводить дефрагментацию с использованием встроенных возможностей. Для этого в «Компьютере» необходимо выбрать необходимый диск и произвести запуск контекстного меню. Затем выбирается «Свойства», там пункт «Сервис» и указать «Оптимизировать». Затем ОС предложит выполнить дефрагментацию.

### Переразметка

В случаях, когда компьютер используется длительное время и работает винчестер на полную мощностью, можно произвести переразметку. Это поможет при наличии двух и более разделов.

Это форматирование диска, а некоторые разработчики указывают, что заново «режется» жесткий диск на сектора и дорожки.

Идеальным является вариант, когда в системе применяется один жесткий диск с единственным томом. Нормальной считается система с двумя разделами. При большем количестве – система будет более медленно работать, тратя время на поиск необходимых файлов. В данном процессе поиск осуществляется по всем жестким дискам поочередно. Можно один жесткий диск использовать для установки ОС и софта, на другой записывать фильмы и прочие мультимедийные файлы.

### Перемещение файлов на съемный носитель

Неплохим вариантом считается перемещение неиспользуемых данных на отдельные носители. Зачастую для этого применяются внешние накопители. Важно выбирать надежные внешние HDD, которые будут отличаться высокой скоростью обработки информации.

Изделия можно использовать не только для хранения устаревших файлов, но и в виде флешки. Для подобных целей 500 Гб будет достаточно. С компьютером он соединяется по USB интерфейсу и отличается бесшумной работой.

### Отключение автозагрузки программ

Наличие лишних программ в автозагрузке существенно замедляет работу компьютера и использует его процессор и память. Если их отключить, то можно увеличить скорость работы диска и в целом производительность системы.

В разных версиях Виндовс управление автозагрузкой располагается в разных местах. Наиболее удобно для подобной цели использовать утилиту «CCleaner». Требуется осуществить запуск утилиты и перейти в меню «Сервис/Автозагрузка».

В автозагрузке нужно оставить только те программы, которые нужны для работы компьютера, например, антивирус. Остальные компоненты можно отключить кнопкой «Выключить» в правой панели.

Любую из отключенных программ можно всегда снова запустить вручную. Если пропадет значок в системном трее, то ее запуск можно вернуть с использованием кнопки «Включить».

### Обновление драйвера контроллера диска

От актуальности драйверов зависит стабильность всей системы. Рекомендуется обновлять драйверы до новейшей версии, особенно это касается SSD дисков.

Для проверки версии установленных драйверов нужно зайти в «Диспетчер устройств». Необходимо вызвать командную строку, а в ней ввести «devmgmt.msc» и нажать «Enter». Затем нажать правой кнопкой мыши по SATA контроллеру и выбрать пункт «Свойства».

Если используется старый драйвер, то новый можно найти на официальном сайте производителя системной платы или ноутбука, скачать и обновить его.

Существуют специальные утилиты, которые позволяют обновить драйверы. Можно выбрать софт «Driver Booster», который при запуске покажет список драйверов, для которых требуется обновление.

Данная информация поможет Вам оптимизировать жесткий диск и повысить его работоспособность.

## Способы увеличения быстродействия программ

Современные ЭВМ обладают очень большой мощностью. Скорость работы процессора (ЦП) современных ЭВМ измеряется гигагерцами, объём оперативной памяти гигабайтами, а современные интерфейсы устройств обеспечивают скорость обмена данными порядка, как минимум, нескольких сотен мегабайт в секунду. Производительность, которая ещё несколько лет назад казалась «сказочной» в настоящее время стала нормой жизни.

Однако параллельно росту мощности ЭВМ увеличивается и ресурсоёмкость приложений. У приложений совершенствуется функционал, интерфейс, возрастает объём обрабатываемых данных и как следствие системные требования. Поэтому вопрос об увеличении быстродействия приложений не теряет своей актуальности.

**Общие вопросы быстродействия программ.**

Быстродействие программ (ПО) зависит от многих факторов, но основными из них являются два:

* Соотношение между реальными системными требованиями ПО и существующей аппаратной конфигурацией ЭВМ;
* Алгоритмы работы ПО.

Если низкое быстродействие обусловлено первым фактором, то решением является модернизация аппаратной части (hardware). В некоторых случаях проблему можно решить также с помощью тонкой настройки hardware и операционной системы. Однако этот путь имеет ряд недостатков:

1. Увеличивается производительность hardware, а вовсе не быстродействие ПО;
2. Производительность hardware ограничена возможностями существующих в данный момент элементной базы и инженерных решений в данной области;
3. Большие финансовые затраты на модернизацию и настройку по причине высокой стоимости комплектующих ЭВМ и услуг специалистов требуемой квалификации.

По этим причинам при разработке ПО прибегают к увеличению его быстродействия с помощью различных средств программной инженерии. Это позволяет:

1. Обеспечить работу нового ПО на уже существующем оборудовании;
2. Разработать масштабируемое ПО;
3. Значительно уменьшить финансовые и трудовые затраты при внедрении.

Вместе с тем и у этого пути имеется ряд недостатков:

1. Значительно усложняется процесс разработки ПО, так как более «быстрые» алгоритмы сложнее более «медленных» (на пример алгоритм бинарного поиска сложнее, чем алгоритм линейного поиска);
2. Реализация более сложных алгоритмов, как правило, требует привлечения специалистов более высокой квалификации;
3. В случае работы с большими объёмами данных или выполнении задач требующих больших и сложных вычислений, ресурсоёмкость ПО всё равно остаются достаточно высокой. Несмотря, на какие либо способы увеличения быстродействия.

Таким образом, в общем случае обеспечение быстродействия ПО является комплексной задачей.

Однако следует заметить, что среди существующих задач, очень немногие обладают высокой ресурсоёмкостью. Вследствие этого в большинстве случаев не требуется никаких действий относительно hardware и требуемого результата можно достичь, прибегая только к программной инженерии.

Программная инженерия предоставляет несколько способов увеличения быстродействия программ. Рассмотрим их на примере языков программирования Delphi и Assembler.

**Увеличение быстродействия программ.**

Как было показано в предыдущем параграфе, можно увеличить быстродействие ПО соответствующим образом реализовав его алгоритмы. Количественным показателем быстродействия алгоритма (а, следовательно, и ПО) является время его выполнения, измеренное по специальной методике, так называемого профилирования. Таким образом, в общем случае выбор наиболее «быстрых» алгоритмов сводится к измерению времени их выполнения и сравнении полученных результатов между собой. Такой способ анализа быстродействия является наиболее объективным. На протяжении многих лет программистами был накоплен большой опыт профилирования, который позволяет сделать определённые выводы относительно возможности оптимизации быстродействия ПО ещё на стадии написания.

Эти выводы были обобщены и представлены в виде определённых рекомендаций. Если программист будет следовать данным рекомендациям, то написанная программа вероятнее всего будет обладать большим быстродействием, чем в случае их игнорирования. Однако следует ещё раз подчеркнуть, что достоверные сведения о быстродействии может дать только профилирование. Это обусловлено тем, что быстродействие алгоритма определяет в первую очередь его конкретная реализация. Кроме того необходимо ещё раз отметить, что в отношении увеличения быстродействия ПО программная инженерия не всесильна.

В чём же состоят выше упомянутые рекомендации? Их краткое содержание применительно к языку программирования Delphi приведено ниже.

1. При написании кода программ рекомендуется избегать процедур, состоящих из сотен строк. Практически всегда в них можно выделить блоки, которые лучше оформить в виде отдельной процедуры. Возможно, позже вы ей даже воспользуетесь где-то в другом месте. Не говоря уже о том, что это повышает понимание программы и вами, и другими программистами. К тому же так проще искать «узкие» места в программе.
2. Использование оператора case (switch) вместо многократных if… then… else (if… else). Во втором варианте компилятор будет выполнять проверку условия столько раз, сколько у вас вариантов. В первом проверка выполняется лишь однажды.
3. Некоторые действия могут быть довольно продолжительными, поэтому рекомендуется выносить за рамки цикла всё, что можно выполнить вне его, чтобы избежать большого числа повторений внутри цикла.
4. В циклах типа for нужно стараться, чтобы значение счетчика уменьшалось до нуля, а не наоборот — начиналось с нуля. Это связано с особенностями процессора. Сравнение с нулём выполняется гораздо быстрее, чем с другим числом.
5. Пользоваться типом Variant только при необходимости. Операции над этим типом сложнее, чем, например, над Integer или String.
6. Не злоупотреблять «программированием на компонентах». В частности не использовать компонент TTreeView для хранения древовидных структур данных — он работает очень медленно и предназначен только для визуального отображения. В случае работы со структурами данных лучше использовать алгоритмы, созданные самостоятельно на основе фундаментальных.
7. Сохранение и загрузка свойств компонентов с помощью методов ReadComponent и WriteComponent работает довольно медленно, поэтому по возможности рекомендуется сохранять и восстанавливать состояние программы между сеансами при помощи других способов.
8. Заменить простой в реализации алгоритм на более сложный, но с большим быстродействием. Например, если заранее известно, что в списке для поиска будет много элементов, лучше его отсортировать и применять бинарный поиск вместо линейного.
9. В критических с точки зрения быстродействия местах программы делать вставки на ассемблере. Команды ассемблера напрямую транслируются в машинный код. Таким образом, в отличие от высокоуровневых языков при компиляции отсутствует проблема синхронизации и ряд других негативных обстоятельств.

Для других языков программирования вышеприведённый список может несколько отличаться, в частности отсутствием поддержки ассемблера и как следствие возможности оптимизации с его помощью (Java, Visual C# и др.).

Особо следует отметить, что рекомендации 3 и 4 применяются не только для языков высокого уровня, но и для ассемблера. Помимо вышеуказанных для увеличения быстродействия программ написанных на ассемблере, в том числе и вставок, существуют следующие рекомендации:

1. Замещение универсальных инструкций на учитывающие конкретную ситуацию, например, замена умножения на степень двойки на команды сдвига (отказ от универсальности).
2. Уменьшение количества передач управления в программе: за счет преобразования подпрограмм в макрокоманды для непосредственного включения в машинный код; за счет преобразования условных переходов так, чтобы условие перехода оказывалось истинным значительно реже, чем условие для его отсутствия; перемещение условий общего характера к началу разветвленной последовательности переходов; преобразование вызовов, сразу за которыми следует возврат в программу, в переходы («сращивание хвостов» и «устранение рекурсивных хвостов») и т.д.
3. Максимальное использование всех доступных регистров за счет хранения в них рабочих значений всякий раз, когда это возможно, чтобы минимизировать число обращений к памяти, упаковка множественных значений или флагов в регистры и устранение излишних продвижений стека (особенно на входах и выходах подпрограмм).
4. Использование специфических для данного процессора инструкций, например, инструкции засылки в стек непосредственного значения, которая имеется в процессоре 80286 и более поздних. Другие примеры – двухсловные строковые инструкции, команды перемножения 32-разрядных чисел, деление 64-разрядного на 32-разрядное число и умножение на непосредственное значение, которые реализованы в процессорах 80386 и 80486. Программа должна, разумеется, вначале определить, с каким типом процессора она работает!

**Заключение.**

Методы оптимизации быстродействия, рассмотренные в этой статье, были выработаны и проверены не одним поколением программистов и уже стали классическими. В то же время информационные технологии, в частности технологии программирования, постоянно развиваются. Появляются новые технологии, старые – модернизируются или уходят в прошлое. Растёт производительность аппаратной части ЭВМ и параллельно с этим растёт сложность и ресурсоёмкость выполняемых ими задач.

1. Опишите инструменты повышения производительности программного обеспечения.

## 25 инструментов, повышающих производительность (некоторые из них не являются приложениями) 1. [Unroll.me](https://unroll.me/) — Обработка электронной почты Поместите все свои подписки и рекламные предложения в один удобоваримый email 2. [Evernote](https://evernote.com/intl/ru/) — Все ваши мысли в одном месте и с удобным поиском Заплатите за премиум-версию и получите возможность находить нужный текст среди рукописных заметок. Отличное решение для собраний с электронной «доской для записей». 3. [Scannable](https://evernote.com/intl/ru/products/scannable/) — Сканируйте документы с помощью своего iPhone и загружайте их в Evernote или на облако, в ваш аккаунт iCloud, GoogleDrive или на электронный адрес. Обычно все происходит быстро, но обязательно располагайте документы на контрастном фоне. 4. [GoogleDrive](https://www.google.com/drive/) — Все ваши данные доступны в любое время … на облаке. Отличное решение для бизнеса. 5. [Calm.com](http://www.calm.com/) — Медитация для новичков Очень полезна опциональная звуковая навигация. 6. [SleepTime](http://www.azumio.com/s/sleeptime/index.html) — Будильник, который синхронизируется с вашими циклами сна. Как бы круто. Реально ценны здесь тэги. Отметьте каждую ночь с помощью тегов «поздно поел», «чужая постель» или «принимал кофеин сегодня». Таким образом вы сможете соотнести хороший сон с собственным поведением и сделать соответствующие выводы. И не так уж важно, [сова вы или жаворонок](http://testutor.ru/blog/15-lyudej-kotorye-dokazyvayut-chto-vam-ne-pridetsya-prosypatsya-rano-chtoby-byt-uspeshnym/)! 7. [TrunkClub](https://www.trunkclub.com/) — Персональный стилист и управление гардеробом «Мужчины ненавидят ходить по магазинам, но им нравится хорошо выглядеть». Определите свои размеры и объединитесь с тем, кто способен подобрать вещи, которые вам подойдут. Если вам что-то не понравится, верните. Вы платите только за то, что используете. 8. [IfTTT](https://ifttt.com/) — Веб-триггеры, автоматизирующие скучные задания Работать и [учиться можно совершенно без скуки](http://testutor.ru/blog/kak-prevratit-izuchenie-anglijskogo-yazyka-v-veselyj-i-uvlekatelnyj-process-2/)! 9. [Zapier](https://zapier.com/) — Как и IfTTT автоматизирует, обладает различным интегрированным ПО Это приложение чуть больше сфокусировано на бизнесе. В нем привлекает возможность перемещать файлы между разными аккаунтами GoogleDrive. 10. [Mint](https://www.mint.com/) — Лучший и простейший из доступных сегодня бесплатных инструментов для управления личными финансами 11. [Expensify](https://www.expensify.com/) – Сделайте фото документа, подтверждающего производственные расходы, … с помощью телефона. Возмещение затрат, удержание налогов, добавление заметок и т. д. 12. [Siri](https://www.apple.com/ios/siri/) — Личный помощник от Apple Один из наиболее незаметных инструментов, к которому у вас уже есть доступ. Он предназначен для фиксации идей, мыслей или списков дел. Поверьте, вы получаете от Siri гораздо меньше, чем следовало бы. Практическое применение:

* Какая сегодня погода?
* Перенеси собрание с 14:00 на 16:00
* Позвони на работу моей маме
* Поставь таймер на 25 минут  
  **13.**[**Sunrise**](https://calendar.sunrise.am/)**— Самый синхронизированный и красивый из доступных календарей**  
  Его пульт управления/приложение Meet серьезно меняет индивидуальное расписание. Есть приложения для настольного компьютера и iOS. Поразительно удобно.  
  **14.**[**HealthTap**](https://www.healthtap.com/)**— Быстрые и бесплатные ответы на вопросы о здоровье от настоящих враче**  
  Обычно действует очень шустро. Вы можете оплатить подписку, если хотите получать еще более быструю реакцию или персональную помощь.  
  **15.**[**The Workshop School of Action**](https://thisisworkshop.com/)**— Самый краткий путь к успеху в бизнесе**  
  Вы удивитесь, зачем вы (или кто-то еще) платили за MBA, если можно пройти это обучение.  
  **16.**[**Hipmunk**](https://www.hipmunk.com/)**— Расписание полетов, в котором действительно можно разобраться**  
  Нигде больше нет такой графики. Изумительно. Просто.  
  Сортировка по таким критериям, как «Цена», «Продолжительность» и «Мучение».  
  **17.**[**Airfare Watchdog**](http://www.airfarewatchdog.com/)**— Поиск дешевых перелетов, которым занимаетесь не вы**  
  **18.**[**ResumeRuby**](https://resumeruby.com/)**— Быстрые и красивые шаблоны для создания резюме**  
  Это лучший и самый быстрый способ выделить свое резюме среди кипы заявок от других соискателей на рабочем столе любого рекрутера.  
  **19.**[**Christian Baum`s Biz Resources**](http://www.achristianbaum.com/resources/)  
  Одна из шикарнейших коллекций предпринимателя, состоящая из полезных для бизнеса инструментов.  
  **20.**[**LastPass**](https://lastpass.com/)**— Управление паролями для всех платформ в режиме реального времени**  
  Это конкурент OnePassword. Если вы пользуетесь Mac, вам следует попробовать и OnePassword тоже. Оно милее и в большей степени ориентировано на пользователя.  
  **21.**[**ToDoist**](https://todoist.com/)**— Вездесущий список дел**  
  Синхронизируется со всем и везде, дает прекрасную визуализацию скорости реализации проекта и выдает очень «человеческие» варианты даты исполнения, к примеру «к следующему понедельнику» или «каждый понедельник и вторник до начала сентября».

1. Опишите средства диагностики оборудования и процесс разрешения проблем аппаратного сбоя

## Диагностика аппаратных проблем

### Введение

Большинство пользователей беззаботно работают на компьютере и не задумываются о том, что в какой-то момент компьютер может выключиться и больше не включиться вовсе. Да и достаточно часто возникает проблема – только что собранный или обновленный компьютер не включается. А еще хуже, если компьютер внезапно перестает работать. В таком случае главное – правильно идентифицировать поломку. Ведь может и ремонт не понадобится.

### Отчего и почему?

Для начала стоит разобраться с причинами, которые могут вызвать такое явление. Как известно и пыль и неблагоприятные климатические условия ухудшают состояние компонентов ПК. Соответственно, выход железа из строя может быть вызван окислением контактов, попаданием пыли (и следственно, статического электричества) на микросхемы и разъемы, их перегрев. Перегрев также может быть вызван и плохим охлаждением.

Также все эти ужасы также могут стать следствием скачка напряжения, нестабильностью блока питания, а также неправильного заземления. Первое, что здесь можно порекомендовать – использовать сетевые фильтры, UPS и заземление компьютера. Но помните – лучше вообще не заземлять компьютер, чем заземлять его неправильно. Во-первых, заземлять корпус ПК и модем с телефонной линией надо отдельно. Не стоит заземлять корпус на отопительную батарею, поскольку на тот же стояк ваши соседи могут заземлять, например, холодильник, стиральную машину или перфоратор. В таком случае, эта «земля» уже станет фазой с разностью потенциалов. Нежелательно заземлять несколько устройств в одну «землю» одновременно. Кстати говоря, поэтому не рекомендуется бытовую технику подключать в один сетевой фильтр с компьютером, а вот монитор, принтер и системный блок лучше запитать от одного сетевого фильтра.

К неплохому фейерверку из микросхем может привести и закорачивание какого-либо провода или попаданием питания на земляной контакт. Поэтому всегда стоит следить за качеством подключения кабелей и их состоянием.

### Типичные проблемы

Ну а если уж беда случилась, то придется ее диагностировать. Итак, начнем. Для начала приведем полезную статистику, чтобы примерно знать, где может быть собака зарыта.

Если компьютер в состоянии клинической смерти, то, прежде всего, надо сделать вскрытие и постараться найти характерный запах гари и выяснить, откуда он идет. Если его нет, то стоит проверить надежность подключения питания. Если проверка не помогла, то стоит включить ПК и проверить, крутятся ли вентиляторы блока питания (БП), корпуса и кулера процессора (заодно проверьте крепление кулера). Если не крутятся, и винчестер не издает характерного звука раскручивания шпинделя, то вышел из строя блок питания. Наличие напряжения на его выходе можно проверить тестером померив величину напряжения на контактах системной платы в том месте, где жгут проводов питания соединен с БП. Стоит подключить новый БП и проверить целостность остальных компонентов. Для начала их необходимо визуально осмотреть на предмет наличия горелых элементов.

Несмотря на то, что рабочий монитор ломается достаточно редко, стоит проверить, подаются ли на него сигналы с видеоадаптера. Для этого осциллографом на контактах 10 и 13 (земля и синхронизация соответственно) 15-контактного разъема D-Sub видеоадаптера, вставленного в материнскую плату, нужно проверить наличие рабочих сигналов.

Чтобы облегчить задачу поиска неисправного компонента, приведу наиболее часто встречающиеся симптомы поломок различного оборудования. Когда процессор выходит из строя, то чаще всего на его ножках видны следы гари.

В материнских платах наиболее часто встречающаяся поломка – выход из строя дискретных элементов, особенно конденсаторов в VRM (Voltage Regulation Module, представляет собой LC-фильтр). Да и сам этот блок может выгореть. Нередко электролитические конденсаторы попросту вздуваются, что требует их замены. Также часто встречающийся момент – «выбивание» транзисторов в районе северного моста, модулей памяти и VRM. Их можно определить по подгоревшим ножкам и потемнениям в этой области. Встречаются и выходы из строя тактовых генераторов и линий задержки, а также выгорание портов.

Также иногда встречающееся явление – нарушение контакта на плате. Это может быть вызвано помещением платы расширения в слот не до конца, прогибом платы, закорачиванием контактов на обратной стороне платы на корпус, нехваткой длины проводов, идущих от БП к материнской плате.

В винчестерах самое уязвимое место – перегревшийся контроллер и IDE-разъем. Сгоревший контроллер можно определить по потемнениям рядом с местами его крепления. Перегрев микросхемы приводит и к ухудшению контакта между контроллером HDD и гермоблоком. Механические проблемы двигателя винчестера можно определить по сильной вибрации корпуса HDD при вращении дисков. Массовые неполадки были замечены у дисков IBM серии DTLA и Ericsson (70GXP и 60GXP), Maxtor 541DX, Quantum Fireball 3, Fujitsu серии MPG.

В CD-приводах чаще всего выходит из строя оптико-механическая часть. В частности механизм позиционирования лазера и определения диска. Как правило, такая поломка вызывается неисправностью МСУ (микропроцессор системного управления), который вырабатывает управляющие сигналы, а также драйвера двигателя лазерного считывателя, который отвечает за сигнал возбуждения. Для их проверки необходимо промерить выходные сигналы на соответствующих контактах МСУ. Характерным симптомом неисправности МСУ является отсутствие перемещения лазерного считывателя при первоначальном включении питания. У флоппи-дисководов чаще всего встречаются механические поломки связанные с подъемником и прижимом дискеты.

### Программно-аппаратная диагностика

Если все вышеперечисленное не помогло определить поломку, то придется перейти к программно-аппаратной диагностике. А для того, чтобы она прошла успешно необходимо точно знать, каков порядок включения устройств ПК.

Итак, рассмотрим порядок загрузки компьютера.

1. После включения питания БП выполняет самотестирование. Если все выходные напряжения соответствуют требуемым, БП выдает на материнскую плату сигнал Power\_Good (P\_G) на контакт 8 20-контактного разъема питания ATX. Между включением ПК и подачей сигнала проходит около 0,1-0,5 с.
2. Микросхема таймера получает сигнал P\_G и прекращает генерировать подаваемый на микропроцессор сигнал начальной установки Reset. Если процессор не исправен, то система зависает.
3. Если CPU жив, то он начинает выполнять код, записанный в ROM BIOS по адресу FFFF0h (адрес программы перезагрузки системы). По этому адресу находится команда безусловного перехода JMP к адресу начала программы загрузки системы через конкретный ROM BIOS (обычно это адрес F0000h).
4. Начинается выполнение конкретного кода ROM BIOS. BIOS начинает проверку компонентов системы на работоспособность (POST – Power On Self Test). Обнаружив ошибку, система подаст звуковой сигнал, так как видеоадаптер пока еще не инициализирован. Проверяется и инициализируется чипсет, DMA и происходит тест определения объема памяти. Если модули памяти вставлены не до конца или некоторые банки памяти повреждены, то или система зависает или звучат длинные повторяющие сигналы из системного динамика.
5. Происходит разархивирование образа BIOS в оперативную память для более быстрого доступа к коду BIOS.
6. Инициализируется контроллер клавиатуры.
7. BIOS сканирует адреса памяти видеоадаптера, начиная с С0000h и заканчивая C7800h. Если BIOS видеоадаптера найден, то проверяется контрольная сумма (CRC) его кода. Если CRC совпадают, то управление передается Video BIOS, который инициализирует видеоадаптер и выводит на экран информацию о версии Video BIOS. Если контрольная сумма не совпадает, то выводится сообщение «C000 ROM Error». Если Video BIOS не найден, то используется драйвер, записанный в BIOS ROM, который инициализирует видеокарту.
8. ROM BIOS сканирует пространство памяти начиная с C8000h в поисках BIOS других устройств, таких как сетевые карты и SCSI-адаптеры, и проверяется их контрольная сумма.
9. BIOS проверяет значение слова по адресу 0472h, чтобы определить, какая загрузка должна быть выполнена – «горячая» или «холодная». Если по этому адресу записано слово 1234h, то процедура POST не выполняется, происходит «горячая» загрузка.
10. В случае холодной загрузки выполняется POST. Инициализируется процессор, выводится информация о его марке, модели и т.д. Выдается один короткий сигнал.
11. Тестируется RTC (Real Time Clock).
12. Определение частоты CPU, проверка типа видеоадаптера (в том числе встроенного).
13. Тестирование стандартной и расширенной памяти.
14. Присвоение ресурсов всем ISA-устройствам.
15. Инициализация IDE-контроллера. Если используется 40-контактный шлейф для подключения ATA/100 HDD, то появится соответствующее сообщение.
16. Инициализация FDC-контроллера.
17. ROM BIOS ищет системную дискету или MBR жесткого диска и читает сектор 1 на дорожке 0 стороны 0, копирует этот сектор по адресу 7С00h. Далее происходит проверка этого сектора: если он оканчивается сигнатурой 55AAh, то MBR просматривает таблицу разделов (Partition Table) и ищет активный раздел, а затем пытается загрузиться с него. Если первый сектор оканчивается любой другой сигнатурой, то вызывается прерывание Int 18h и на экран выводится сообщение «DISK BOOT FAILURE, INSERT SYSTEM DISK AND PRESS ENTER» или «Non-system disk or disk error».

В общем-то все. Что касается последнего пункта, то ошибки указанные в нем говорят о неисправности винчестера (программной или аппаратной). Теперь вам остается только выявить, в какой именно момент перестает работать ваш компьютер. Если это происходит до появления сообщений на мониторе, то неисправность можно определить по звуковым сигналам. Наиболее часто встречающиеся звуковые сигналы приведены в таблице.

Стоит заметить, что звуковые сигналы могут отличаться от приведенных выше из-за различия версий BIOS. Если же и звуковые сигналы не помогли определить неисправность, то остается лишь уповать на аппаратную диагностику. Она производится несколькими средствами.

### Аппаратная диагностика

Первое средство весьма банально, но вполне действенно. Работу отдельных блоков можно проверить, дотронувшись до них рукой, чтобы проверить их нагрев. После минутного включения должны греться чипсет, процессор, чипы памяти и блоки видеокарты. Если они кажутся теплыми, то этого достаточно, чтобы сделать вывод хотя бы о том, что на эти элементы подается питание. С большой долей вероятности они должны оказаться рабочими.

Второе средство более научно и требует некоторой инженерной подготовки. Заключается оно в измерении потенциалов на различных элементах. Для этого нужен тестер и осциллограф. Желательно иметь карту разводки материнской платы, поскольку она многослойная, и прохождение сигналов не так очевидно. Начать измерения стоит с силовых элементов входных цепей и стабилизирующих и шунтирующих конденсаторов, проверить наличие +3,3 и +5 В в соответствующих местах материнской платы, работу тактовых генераторов. После этого стоит проверить наличие штатных сигналов на выводах сокета процессора. Далее проверить наличие сигналов в слотах и портах. В последнюю очередь стоит заняться логическими элементами (хотя ремонт их часто оказывается делом неразумным). Для этого вам потребуется знание разводки портов и слотов. Эта информация приведена в таблицах

Третье и последнее средство диагностики – профессиональные аппаратные средства диагностики. К ним относится использование диагностических карт типа ДП-1 и комплекса PC-3000, созданных компанией «РОСК». Диагностическая плата устанавливается в свободный слот материнской платы, и после включения ПК на ее индикаторе отображается код ошибки в шестнадцатеричном виде. Применение такой платы существенно повышает вероятность локализации неисправности. Использование ДП-1 рассчитано на корректную работу процессора, а CPU выходит из строя крайне редко.

На данный момент в России диагностические карты, тестовые ROM BIOS и другие средства диагностики производятся компанией ACE Laboratory.

При аппаратной диагностике следует иметь ввиду, что в большинстве случаев выходит из строя только одно устройство, и проще всего его выявить, заменив на аналогичное, гарантированно работающее.

Что касается блоков питания и периферийных устройств, то диагностика неисправностей в них – тема отдельного разговора, но по поводу мониторов можно дать ряд советов. Достаточно часто из строя выходит промежуточный строчный трансформатор, включаемый между предоконечным и выходным транзистором строчной развертки. Основной его неисправностью, как правило, бывает короткое замыкание витков. Этот трансформатор – часть высоковольтного блока строчной развертки. Это высокое напряжение подается на ЭЛТ. Поэтому часто отсутствие свечения на экране и отсутствие растра указывают на отсутствие высокого напряжения. Как правило, вертикальная полоса на экране также указывает на отказ блока строчной развертки. Проверить наличие высокого напряжение на ЭЛТ можно проведя рукой по поверхности экрана. Если высокое напряжение подается, то вы должны почувствовать некоторую вибрацию или потрескивания статического электричества.

### Программная диагностика

Если же ваш компьютер все же включается, но работает нестабильно, зависает при загрузке, «выпадает» в синий экран, то это чаще всего является следствием переразгона, локального перегрева или «глючностью» памяти, а также ошибками работы HDD (к ним относится и «падение» Windows).

Стабильность их работы можно проверить под DOS, загрузившись c системной дискеты или диска. Для этого следует использовать утилиты CheckIT, PC Doctor, Memtest 86, Stress Linux, Norton Diagnostics, The Troubleshooter. Для профессионального тестирования и восстановления HDD следует использовать HDDUtility и MHDD, но они корректно работают только под MS-DOS 6.22. Первое, что требуется сделать с помощью них – проверить SMART-атрибуты состояния HDD. Также для диагностики, проверки и пометки bad-секторов можно использовать Norton Disk Doctor.

Следует помнить, что полноценную проверку железа можно произвести только под Windows, тестируя стабильность работы в burn-in тестах в течение не менее чем 24 часов. Среди таких тестов можно привести CPU Hi-t Professional Edition, CPU Stability Test, Bionic CPU Keeper, CPU Burn, Hot CPU Tester Pro, HD\_Speed, DiskSpeed 32, MemTest.

А вообще, как известно, гораздо легче предупредить событие, чем исправить его последствия, поэтому гораздо легче регулярного (хотя бы раз в несколько недель) следить за параметрами выдаваемых блоком питания напряжений, смотреть SMART-параметры HDD (программы Active SMART, SMARTVision, SMART Disk Monitor), изучать температуру процессора, проверять наличие хорошего охлаждения и отсутствие посторонних звуков. Нелишним было бы и смазывание вентиляторов машинным маслом, как минимум раз в полгода.

## Как выявлять и устранять проблемы с аппаратным сбоем

**Семь методов:**

* Проверка жесткого диска
* Проверка памяти
* Проверка питания
* Крепление материнских плат
* Остановить перегрев
* Случайные выключения
* Поиск профессиональной помощи

Вы когда-нибудь находили проблемы с компьютером? Удалено и переустановлено программное обеспечение и найти там что-то не так? В этой статье вы узнаете, как выявлять и устранять проблемы, связанные с сбоем компьютерного оборудования, если вы используете операционную систему Windows.

### Метод 1. Проверка жесткого диска

#### 1. Попытайтесь определить плохие сектора

Плохие сектора — это сектора на жестком диске, которые больше нельзя использовать. Это может быть связано с постоянным повреждением или невозможностью доступа ОС к ним. Если вы обнаружите, что система замерзает, получают ошибки остановки или другие ошибки, это может быть связано с плохими секторами. Используйте chkdsk и устраните эти проблемы.

#### 2. Запуск chkdsk

Для этого нажмите «Пуск» (в нижнем левом углу), а затем вы увидите несколько параметров и выберите компьютер.

* Щелкните правой кнопкой мыши том, который вы хотите проверить, и нажмите на свойства.
* В диалоговом окне «Свойства» перейдите на вкладку «Инструменты».
* В разделе «Проверка ошибок» есть кнопка «Проверить сейчас». Нажмите на это, чтобы запустить chkdsk.
* В диалоговом окне «Проверить диск» выберите параметры, которые вы хотите запустить. Чтобы попытаться исправить плохие сектора, вы должны проверить второй вариант: «Сканировать и пытаться восстановить поврежденные сектора».

Если вы проверяете системный том, вы увидите сообщение «Windows не может проверить диск во время использования. Вы хотите проверить ошибки жесткого диска при следующем запуске вашего компьютера? »Нажмите« Запуск проверки диска », чтобы запустить проверку в следующий раз при запуске компьютера.

#### 3. Запуск chkdsk из командной строки:

Нажмите «Пуск», введите cmd, затем щелкните правой кнопкой мыши cmd и выберите «Запуск от имени администратора».

* Введите chkdsk без параметров, чтобы увидеть состояние диска.
* chkdsk /? Отобразятся все возможные параметры команды.

Введите chkdsk c: \ f \ v, чтобы проверить и восстановить диск, а также отобразить любые сообщения очистки.

* Если вы хотите проверить том, отличный от C: измените «С» на соответствующую букву.

Если вы проверяете свой системный том, вы увидите сообщение:

* «Тип файловой системы — NTFS. Невозможно заблокировать текущий привод. Chkdsk не может работать, поскольку этот том используется другим процессом. Вы планировали бы, чтобы этот том был проверен при следующей перезагрузке системы? (Y / N)»
* Введите Y, а затем перезагрузите компьютер. Появится сообщение о том, что chkdsk запущен. Когда это закончится, Windows запустится автоматически.

### Метод 2. Проверка памяти

#### 1. Диагностика проблем памяти

Неисправная оперативная память может вызвать проблемы с системой. Некоторые из наиболее распространенных признаков проблемы с памятью — это ошибки остановки, которые система не запускает.

#### 2. Если система не запускается

первое, что вы должны попробовать, это «Восстановление при запуске». Это попытается устранить любые ошибки на жестком диске или проблемы с конфигурацией программного обеспечения, которые могут помешать запуску компьютера в обычном режиме. Если после этого компьютер не запускается, запустите диагностику памяти Windows в диспетчере загрузки Windows.

#### 3. Обратите внимание, что Windows Memory Diagnostic не может быть запущена во время работы Windows

Таким образом, вы можете запланировать его запуск при следующем запуске компьютера. Для этого перейдите в Панель управления, нажмите «Система и безопасность», а затем «Администрирование». Дважды щелкните значок «Диагностика памяти Windows», а затем выберите соответствующий параметр.

#### 4. Откройте Диагностический планировщик памяти Windows

Введите mdsched в командной строке или нажмите «Пуск» и введите mdssched.

#### 5. Запустите диагностику памяти Windows

через диспетчер загрузки Windows, если ваш компьютер не стартует. Чтобы получить доступ к этому, несколько раз нажмите клавишу пробела при запуске системы.

* Нажмите «Tab», чтобы выбрать «Диагностика памяти Windows», также доступны через «Параметры восстановления системы».

#### 6. Обратите внимание, что по умолчанию Windows Memory Diagnostic запускает стандартный тест с двумя проходами

Существует три уровня тестирования: **базовый**, **стандартный** и **расширенный**.

#### 7. Выберите количество проходов, выполняемых тестами

Больше проходов занимает больше времени, но с большей вероятностью можно найти проблемы с прерывистой памятью.

### Способ 3. Проверка питания

#### 1. Выключите питание и немедленно отключите компьютер, если из него выйдет дым

При необходимости используйте огнетушитель. Убедитесь, что огнетушитель одобрен для использования на электрических устройствах.

#### 2. Если компьютер не делает ничего, когда вы нажимаете кнопку питания

первое, что нужно сделать, это проверить, что он подключен, и настенная розетка включена.

* Убедитесь, что настенная розетка работает. Вы можете сделать это, подключив что-то, что вы знаете, и смотрите, будет ли оно включено.

#### 3. Убедитесь, что шнуры питания подключены к материнской плате

#### 4. Убедитесь, что силовой кабель исправен

Вы можете сделать это с помощью мультиметра или просто поменять кабель на тот, который, как вы знаете, работает.

#### 5. Убедитесь, что все внутренние или внешние выключатели питания включены

Убедитесь, что напряжение правильно установлено на источнике питания

#### 7. Проверьте источник питания, используя его на другом компьютере

Если он не работает, замените его.

#### 8. Если компьютер застывает до запуска операционной системы

возможно, что источник питания может быть недостаточно мощным. Убедитесь, что источник питания обеспечивает необходимую мощность для питания машины.

#### 9. Обратите внимание, что если компьютер выключается с произвольными интервалами

проблема может быть связана с вентилятором питания. Убедитесь, что вентилятор работает.

#### 10. Убедитесь, что вентилятор на материнской плате работает правильно

Система может быть отключена, поскольку она перегревается. Убедитесь, что ваша система чиста от пыли, при необходимости замените вентилятор.

### Способ 4. Фиксация материнской платы

#### 1. Запустите диагностическое программное обеспечение материнской платы

(если оно предоставлено производителем), чтобы убедиться, что материнская плата не повреждена.

#### 2. Устранение неполадок, когда вы не слышите звуковые коды, доступные для просмотра любого видео

* Убедитесь, что компьютер получает питание и что монитор включен и подключен.
* Удалите все внешние аксессуары, такие как беспроводные карты или внешние диски.
* Убедитесь, что вентилятор питания работает. Если это не так, проблема, скорее всего, будет связана с источником питания.
* Откройте компьютер и визуально осмотрите материнскую плату. Если он почернел или расплавлен, замените материнскую плату.
* Убедитесь, что все необходимые разъемы питания подключены к материнской плате и что любой внутренний выключатель питания включен. Также убедитесь, что источник питания настроен на правильное напряжение.
* Убедитесь, что материнская плата, ОЗУ и процессор установлены правильно.
* Если на материнской плате есть перемычки, проверьте руководство, чтобы убедиться, что они находятся в правильном положении.

#### 3. Если ни один из этих шагов не сработал

установите BIOS обратно по умолчанию, удалив батарею с материнской платы в течение 30 минут.

#### 4. Примите меры, если вы слышите звуковые сигналы, но компьютер не запускается

Удалите все внешние аксессуары, например, беспроводные или внешние диски. Просто оставьте монитор, клавиатуру и мышь подключены. Делая это, вы выделяете устройства, которые могут вызывать звуковые коды.

Обратитесь к руководству или веб-сайту производителя, чтобы проверить значение звукового сигнала, который вы слышите.

### Способ 5. Остановить перегрев

#### 1. Потеря мощности после нескольких минут работы является симптомом перегрева

#### 2. Убедитесь, что вентилятор процессора работает

#### 3. Измените положение компьютера, чтобы обеспечить циркуляцию воздуха вокруг него

Это особенно важно с ноутбуками, убедитесь, что ни один из портов охлаждения не заблокирован.

#### 4. Убедитесь, что ни один из вентиляторов не заблокирован

Это, очевидно, приведет к перегреву охлажденной зоны. Вентилятор также может выгорать.

#### 5. Если возможно, проверьте температуру компьютера в BIOS или диагностическую программу

#### 6. Удалите пыль в компьютере

### Mетод 6. Случайные выключения

#### 1. Запустите Windows Memory Diagnostic, чтобы проверить, что ОЗУ не является неисправной

#### 2. Используйте диагностическое программное обеспечение материнской платы

чтобы определить, является ли материнская плата источником проблемы. Диагностическое программное обеспечение может быть получено от производителя материнской платы

### Метод 7. Поиск профессиональной помощи

#### 1. Попросите помощи у профессионала

Нет ничего плохого в просьбе о помощи

* Профессионал может заметить то, что вы, возможно, пропустили

1. Аппаратно-программные платформы серверов и рабочих станций. Установка серверной части. Виды серверного программного обеспечения.

Правильный выбор технических средств оказывает определяющее влияние на эффективность функционирования информационной системы. Для сходных информационных систем построение технических средств может быть осуществлено  в самых различных, но равноценных по функциональному назначению вариантах.

В качестве критериев оптимальности при равных функциональных возможностях могут выступать: минимальная стоимость комплекса технических средств, минимальная стоимость обслуживания и др. При этом обязательно учитывается тот факт, что любая информационная система является постоянно развивающейся системой и ее комплекс технических средств должен иметь возможность при необходимости перестраиваться на решение новых задач.

При расчете параметров технических средств учитывают:

* предполагаемые объемы баз данных;
* сложность алгоритмов обработки данных по каждой задаче;
* количество пользователей и интенсивность их работы с базой данных;
* требуемый уровень надежности всех элементов системы и др. В качестве искомых величин выступают:
* технические характеристики всех составляющих комплекса технических средств (быстродействие процессора, объем оперативной и дисковой памяти и т.д.);
* способы организации вычислительных процессов, режимов работы;
* параметры, характеризующие эффективность работы технических средств и др.

В информационных системах на клиентских рабочих местах, называемых также рабочими станциями, обычно применяют персональные компьютеры. В качестве серверов в средних и крупных информационных системах используют специализированные многопользовательские мощные компьютеры — серверы. В информационных системах с небольшим количеством пользователей и малыми объемами информации в качестве сервера вполне может использоваться и персональный компьютер, обладающий приемлемыми техническими характеристиками. Первый  признак,  по  которому  разделяются  компьютеры,  —  платформа.

Сегодня на рынке представлено несколько основных платформ компьютеров, каждая из которых отличается как по назначению, так и по типу использованного «железа» и программ. Как правило, различные платформы компьютеров несовместимы между собой, — хотя в отдельных случаях программы, написанные для компьютеров одного типа, можно запустить на другом с использованием специальных программ-эмуляторов.

Остановимся более подробно на персональных компьютерах. Они широко используются в информационных системах. Основными достоинствами персональных компьютеров являются:

* небольшие физические габариты;
* мощные вычислительные возможности;
* простота эксплуатации пользователем-непрофессионалом;
* невысокая стоимость;
* отсутствие серьезных требований и ограничений по условиям эксплуатации. Рассмотрим самые популярные платформы персональных компьютеров.

**Платформы персональных компьютеров**

Платформа IBM

Платформа IBM-совместимых компьютеров включает громадный спектр самых различных компьютеров, от простеньких домашних до сложных серверов. Именно с IBM-совместимыми компьютерами вам придется сталкиваться в абсолютном большинстве случаев. Совершенно необязательно, что лучшие IBM- совместимые компьютеры изготовлены фирмой IBM — породившая этот стандарт фирма сегодня лишь один из великого множества производителей ПК.

С момента появления на свет ПК (начиная с середины 70-х годов) в мире существовало (и существует) множество видов этих устройств. Однако сейчас подавляющее большинство персональных компьютеров относятся к типу «IBM PC- совместимых».

Сегодня   все  чаще   говорят  не  об   «IBM-совместимых   компьютерах»,   а  о«платформе WIntel», подразумевая под этим сочетание аппаратного обеспечения — процессоров фирмы Intel и программной платформы — операционной системы Windows.

Современный  IBM-совместимый  ПК  похож  на   детский   конструктор   типа «сделай сам». Каждое из входящих в его состав устройств можно свободно поменять на другое — того же типа, но более совершенное. Благодаря этому становятся возможными две вещи — быстрая сборка компьютера непосредственно «под клиента» в любой, даже самой маленькой компьютерной фирме, а также простая (в большинстве случаев — силами самого пользователя) модернизация.

Сегодня уже нет ни одной детали, которая не была бы представлена четырьмя- пятью фирмами одновременно. Даже основа основ — процессоры — выпускаются сегодня не только знаменитой на весь мир корпорацией Intel, но и другими фирмами, — например, AMD.

Платформа Apple

Специалисты   по   компьютерной   истории   отдают   приоритет   в   создании персональных компьютеров именно компании Apple. С середины 70-х годов эта фирма представила несколько десятков моделей персональных компьютеров, — начиная с Apple I и заканчивая современным iMac — и уверенно противостояла мощной корпорации IBM. В середине 80-х компьютеры серии Macintosh стали самыми популярными персональными компьютерами в мире.

В отличие от IBM, компания Apple всегда делала ставку на закрытую архитектуру — комплектующие и программы для этих компьютеров выпускались лишь небольшим числом авторизованных производителей. За счет этого компьютеры Macintosh всегда стоили несколько дороже своих PC-совместимых конкурентов — что, впрочем, компенсировалось их высокой надежностью и удобством.

Именно на компьютерах Apple впервые появились многие новинки, со временем ставшие неотъемлемой частью персонального компьютера: графический интерфейс и мышь, звуковая подсистема и компьютерное видео и т.д. Даже  интерфейс самой Windows был частично скопирован с одной из ранних  операционных систем Apple.

Работа с графикой и сегодня остается основным козырем Apple — вот почему компьютеры Macintosh по-прежнему незаменимы в таких областях, как издательское дело, подготовка и дизайн полноцветных иллюстраций, обработка видео и звука, обучение. В этом качестве компьютеры Apple и используются сегодня в России

Несмотря на значительное падение интереса к Apple в начале 90-х, к концу десятилетия компьютеры Macintosh вновь привлекли к себе интерес пользователей после выхода моделей с новым, уникальным дизайном, рассчитанным на домашнего пользователя — настольной модели iMac и портативной — iBook (рис. 1.2).

## Серверы

Новое поколение информационных систем получило возможности использования мощных центральных сетевых компьютеров — серверов. Современные операционные системы компьютеров в существенной степени строятся на новой платформе, ориентированной на серверы. Разнотипные компьютеры — от дешевой настольной рабочей станции до мощного сервера — успешно объединяются в комплексы, обеспечивая надежные решения архитектуры информационных систем.

Серверы используют новые более мощные модели процессоров. Компьютерная индустрия планомерно переходит на 64-битные архитектуры серверов и компьютерных приложений. Это требует от пользователей освоения как новых процессоров, так и соответствующих операционных систем. Постепенно осуществляется перенос приложений на новую платформу и их оптимизация. Одно из преимуществ информационных систем с серверами в отличие от других платформ — совместимость процессоров различных поколений, что обеспечивает переносимость прикладных программ без их перекомпиляции. Это важно, поскольку крупные информационные системы, как правило, используют парк компьютеров различных поколений. Если компьютеры несовместимы, то увеличивается потребность в высококвалифицированных кадрах для новой  разработки и поддержки прикладного программного обеспечения.

Мировой лидер производства и поставки серверов — фирма IBM. Корпорация IBM, используя технологию медных микропроцессоров системы S80, создала семейство серверов IBM RS/6000 (модели S80, F80, Н80, М80 и др.), ориентированное на UNIX-платформу.

Разработанная в лабораториях IBM технология медных процессоров позволила резко повысить вычислительную мощность новых моделей. В предыдущих моделях в качестве проводника использовался алюминий. Медные процессоры имеют меньший размер и на 20-30% быстрее и эффективнее, чем их алюминиевые конкуренты.

Новые серверы IBM используют целый ряд особенностей, которые позволяют легко устанавливать и обслуживать их даже в условиях быстрорастущих компаний. В конструкцию серверов включены возможности для непрерывного наращивания, избыточные  вентиляторы   и  источники  питания,   жесткие  диски   с  возможностью «горячей» замены и встроенный на системной плате сервисный процессор.

Важнейшее направление использования серверной архитектуры  компьютерных систем — их объединение в высоконадежные и информационно- безопасные структуры — кластеры.

## Кластерная структура сервера

**Кластер**представляет собой многомашинный компьютерный комплекс, который с точки зрения пользователя:

* является единой системой;
* обеспечивает высокую надежность (отказоустойчивость);
* имеет общую файловую структуру;
* обладает        свойством        эффективной        масштабируемости        —        роста производительности при добавлении ресурсов;
* гибко перестраивается;
* управляется (администрируется) как единая система.

Иногда кластером называют комплекс из двух компьютеров, один из которых делает полезную работу, а другой включен и находится в горячем резерве. Это необходимо для того, чтобы в случае отказа основного компьютера можно было бы мгновенно продолжить вычисления на резервном. В этом случае пользователи, работающие в системе, даже не почувствуют последствия отказа сервера.

Главные же качества кластеров — высокая надежность и масштабируемость. В отличие от систем с горячим резервированием все компьютеры в кластере не простаивают, а выполняют полезную работу. В результате затраты на дополнительное оборудование являются платой не только за надежность, но и за производительность.

Каждый компьютер в кластере остается относительно независимым. Его можно остановить и выключить для проведения, например, профилактических работ или установки дополнительного оборудования, не нарушая работоспособности кластера в целом. Тесное взаимодействие компьютеров, образующих кластер, часто именуемых узлами кластера, гарантирует максимальную производительность и минимальное время обработки пользовательских приложений

При работе кластерной системы в составе АИС в случае сбоя программного обеспечения на одном узле приложение продолжает функционировать на других  узлах кластера. Профилактические и ремонтные работы, реконфигурацию и смену версий программного обеспечения в большинстве случаев можно осуществлять на узлах кластера поочередно, не прерывая работы АИС на других узлах кластера. Таким образом, в составе АИС кластер — это несколько компьютеров, соединенных коммуникационным каналом и имеющих доступ к общекластерным ресурсам, к которым прежде всего относятся дисковые накопители.

Общекластерные дисковые накопители обеспечивают возможность быстрого перезапуска приложений на разных узлах кластера и одновременной работы прикладных программ с одними и теми же данными, получаемыми с разных узлов кластера так, как если бы эти программы находились в оперативной памяти одного компьютера.

Коммуникационный канал кластера обеспечивает:

* скоординированное использование общекластерных ресурсов;
* взаимный контроль работоспособности узлов кластера;
* обмен данными о конфигурации кластера и другой специфической кластерной информацией.

С точки зрения пользователя кластер выглядит как единый сервер. Этот сервер имеет свое собственное имя (кластерное имя), с которым и работают пользователи. Более того, они могут даже не знать подлинные имена серверов, составляющих кластер.

## Сущность концепции открытых систем

Распространение распределенных информационных систем стало возможным благодаря концепции открытых систем. Основным смыслом концепции является упрощение совместимости вычислительных систем за счет международной и национальной стандартизации аппаратных и программных интерфейсов. Развитие концепции была обусловлено переходом к использованию локальных и глобальных сетей и необходимостью решения проблем совместной работы различных аппаратно- программных средств.

Ключевой особенностью открытых систем является независимость от конкретного поставщика. Ориентируясь на продукцию компаний, придерживающихся стандартов открытых систем, потребитель, приобретающий любой продукт такой компании, не попадает к ней в зависимость. Он может продолжить наращивание мощности системы путем приобретения продуктов любой другой компании, соблюдающей стандарты. Причем, это касается как аппаратных, так и программных средств.

Основой открытых систем является стандартизованная операционная система. Сегодня на эту роль претендуют операционные системы UNIX, Windows NT,  Windows 2000.

Технологии и стандарты открытых систем обеспечивают производство программных средств со свойствами мобильности и интероперабельности:

* свойство мобильности обеспечивает сравнительную простоту переноса программного обеспечения на другую аппаратно-программную платформу, соответствующую стандартам;
* интероперабельность означает возможность простого создания новых программных систем на основе использования готовых компонентов со стандартными интерфейсами.

Открытые системы обеспечивают решение проблемы поколений аппаратно- программных средств. Пользователи, по крайней мере, временно могут продолжать комплектовать системы, используя существующие компоненты. Они могут постепенно заменять компоненты системы на более совершенные, не нарушая при этом ее работоспособности.

**Виды серверного программного обеспечения.**

**Серверное ПО** – это ПО, предоставляющее услуги или функции на компьютере, выступающим в качестве среды.

**Виды серверного ПО**:

* **Файловый сервер** – предназначен для обеспечения доступа к файлам, хранящимся на серверных дисках организации.
* **Сервер баз данных** – обеспечивает хранение, обработку и доступ к базам данных компании, осуществляемы с клиентских компьютеров.
* **Сервер приложений** – осуществляет программную обработку данных, которые посылает ему пользователь, и выдает этому пользователю конечный результат.
* **Веб-сервер** – отвечает за выдачу интернет страницы. Может одновременно обрабатывать большое количество запросов.
* **Почтовый сервер** – предназначен для отправки, получения, хранения и распределения электронных писем.
* **Брандмауэр** (файервол) – обеспечивает защиту внутренней сети и ее ресурсов от интернет-атак.
* **Прокси-сервер** – служба, позволяющая выполнять клиентам косвенные запросы к другим сетевым службам.

Прокси-сервер скрывает от внешних пользователей структуру сети, обеспечивает доступ к сети по одному IP-адресу. Позволяет вмести с DNS-сервером производить автоматическую раздачу и назначение IP-адреса.

* **DNS-сервер** – служит для автоматического учета и выдачи уникальных IP-адресов всем узлам, которые к нему обращаются.
* **Сервер удаленного доступа** – позволяет получать через Интернет доступ к локальной сети.
* **Принт-сервер** – позволяет получить доступ к сетевому принтеру.

1. Охарактеризуйте особенности эксплуатации различных видов серверного программного обеспечения.

Сервером сети Интернет называется компьютер, на котором установлена специальная программа (она тоже называется сервером, web-сервером или http-cepвером), которая отображает web-страницы по запросу клиентской машины, атак же выполняет множество других полезных функций, которых мы коснемся чуть позже. Когда ваш домашний компьютер связывается с сервером и получает от него все необходимые данные, например код web-страницы, он выступает в роли клиента, а всю систему в этом случае принято называть связкой клиент-сервер. На этот термин следует обратить особое внимание, поскольку в последствии мы часто будем сталкиваться с ним.

Системой клиент-сервер называют механизм передачи информации между удаленным компьютером, предоставляющим свои ресурсы в распоряжение пользователей, и пользовательским компьютером, эксплуатирующим эти ресурсы. В данном случае компьютер, открывающий доступ к собственным ресурсам, носит название сервера, а получающий такой доступ клиента.

Серверы могут быть разными, причем отличия заключаются, прежде всего, в операционной системе, под управлением которой они работают.В настоящее время на большинстве интернетовских узлов используют два типа серверных программ: либо Internet Information Server, рассчитанный на работу под Windows NT, либо Apache, предназначенный для платформ, совместимых со стандартом UNIX. Как правило, серверы работают на линиях с большой пропускной способностью, например, в сетях с оптоволоконными каналами связи, что по финансовым соображениям доступно лишь крупным предприятиям.

Помимо соответствующей программы настоящий сервер должен иметь собственный домен, то есть адрес DNS, отвечающий стандартам Domain Name System.

Таким образом, сервер это компьютер с установленным на нем специальным программным обеспечением, имеющий собственное доменное имя. Владелец и администратор сервера могут гибко менять необходимые настройки, разрешать или запрещать доступ к его ресурсам, подключать, настраивать и запускать ряд дополнительных программ и функций, таких как скрипты CGI или приложения SSI, то есть полностью конфигурировать его работу по мере необходимости.

Тематическое содержание серверов может варьироваться в широком диапазоне в зависимости от целей, ради которых они были созданы, возможностей или фантазии владельца и многих других условий. Объединяет их все, пожалуй, только одно: полноценный сервер должен представлять собой то, что среди пользователей Интернета принято называть термином информационный портал , то есть в идеальном случае он является достаточно большим виртуальным пространством, состоящим из множества различных тематических разделов меньшего размера, либо некоторого количества самостоятельных проектов.

Файл-серверы и принт - серверы управляют доступом соответственно к файлам и принтерам, на серверах приложений выполняются прикладные части клиент - серверных приложений, а так же находятся данные доступные клиентам. Например, чтобы упростить извлечение данных, серверы хранят большие объемы информации в структурированном виде. Эти серверы отличаются от файл - серверов и принт - серверов.

В принт - серверах, файл или данные целиком копируются на запрашиваемый компьютер. А в сервере приложений на запрашиваемый компьютер посылаются только результаты запроса. Приложение-клиент на удаленном компьютере получает доступ к данным, хранимым на сервере приложений. Однако вместо всей базы данных на ваш компьютер с сервера загружаются только результаты запроса. В расширенной сети использование серверов различных типов становится наиболее актуальным. Необходимо поэтому учитывать всевозможные нюансы, которые могут проявиться при разрастании сети, с тем чтобы изменение роли определенного сервера в дальнейшем не отразилось на работе всей сети. Основным аргументом при работе в сети на основе выделенного сервера является, как правило, защита данных.

В таких сетях, например как Windows NT Server, проблемами безопасности может заниматься один администратор. Поскольку жизненно важная информация расположена централизованно, то есть, сосредоточена на одном или нескольких серверах, нетрудно обеспечить ее регулярное резервное копирование.

Благодаря избыточным системам данные на любом сервере могут дублироваться в реальном времени, поэтому в случае повреждения основной области хранения данных информация не будет потеряна легко воспользоваться резервной копией.

1. Опишите виды клиентского программного обеспечения. Опишите процесс установки, адаптации и сопровождения клиентского программного обеспечения.

**Программное обеспечение** – это совокупность программных и документальных средств для создания и эксплуатации систем обработки данных средствами вычислительной техники.

**Клиентское ПО** – это ПО, осуществляющее доступ к ресурсам, которые предоставляет сервер.

## Установка и сопровождение клиентского программного обеспечения

### Подготовка компьютера перед установкой

* Убедится, что конфигурация компьютера отвечает минимальным
* требованиям ПО.
* Проверить наличия свободного места на жёстком диске, если его не
* хватает, освободить.
* Отключить антивирусные программы.
* Остановить выполнение посторонних программ.

Установка различного ПО может существенно различаться, но чаще всего установка приложения сводится к стандартному набору действий в которых вам помогает мастер установки.

В случае если программа уже была когда то проинсталированна на ПК мастер установки спросит, что делать со старой версией удалить, заменить или восстановить.

После этого можно будет указать путь установки программы и завершить установку выполнив все действия указанные мастером.

Завершающим этапом установки, чаще всего, становится регистрация программного продукта.

### Поддержка программного обеспечения

#### Пакеты исправлений и заплатки

Время от времени производители программного обеспечения выпускают исправления к своим программным продуктам. Как правило, исправления корректируют выявленную проблему, а затраты на их тестирование могут варьироваться в зависимости от программного продукта.

Протестировав исправление и приняв решение о его развертывании, администратор копирует необходимые файлы в точку распространения ПО, производя замену старых файлов. Производители программного обеспечения распространяют обновления либо в виде нового пакета установщика, либо в качестве исправления установщика. В случае поставки нового пакета установщика администратор просто заменяет пакет, размещенный в точке распространения ПО. Если же поставляется исправление установщика, администратор следует указаниям производителя программного продукта по применению MSP-файла к существующему пакету, содержащему приложение.

После обновления файлов в точке распространения ПО администратор открывает расширение Установка программ (Software Installation) в объекте групповой политики, который управляет существующим программным обеспечением, и щелкает правой кнопкой мыши по исправляемому приложению для вызова контекстного меню. Он выбирает команду Развернуть приложение заново (Redeploy application), обеспечивающую копирование исправленных файлов при следующем применении групповой политики к тем пользователям, которые уже установили данное приложение.

#### Пакеты обновления

Между пакетами обновления и исправлениями нет больших отличий. Как правило, в состав пакета обновления входят несколько уже протестированных исправлений. Пакеты обновления распространяются реже, чем исправления, но чаще, чем полные обновления программного продукта.

Рекомендуется распространять пакет обновления в качестве исправления, если он содержит небольшое количество файлов. Распространение пакета с большим количеством файлов и управление им рекомендуется рассматривать в качестве полного обновления программного продукта.

В любом случае, следуйте инструкциям производителя заплатки и тестируйте ее в лаборатории или на небольшой группе пользователей, прежде чем осуществлять распространение заплатки для всех пользователей и компьютеров, управляемых заданным объектом групповой политики.

Пакеты обновлений подготовлены с учетом возможности их управления при помощи компонента установки и поддержки ПО, который является частью технологии IntelliMirror.

#### Обновления

Обновление подразумевает замену значительного числа файлов. Поскольку производятся большие изменения в программном продукте, обновление получает другой номер версии.

Производитель предоставляет новую версию в пакете установщика ПО, с заложенной в него информацией о том, какие более ранние версии он может обновлять. В пакете также должен содержаться сценарий процесса обновления, в котором описано, какие файлы должны быть заменены, удалены и добавлены.

Процесс обновления начинается с размещения файлов программного продукта (пакетов установщика ПО и файлов преобразования для них) в точке распространения ПО. Затем администратор назначает или публикует новую версию в расширении Установка программ (Software Installation). При этом администратор задает преемственную связь между версиями, если это необходимо. Если пакету установщика ПО с новой версией известно об обновляемом приложении, расширение Установка программ автоматически устанавливает преемственную связь между версиями.

Администраторам необходимо определиться с тем, будет обновление обязательным (немедленно вступит в силу для всех пользователей, работающих с текущей версией) или необязательным (пользователи смогут установить новую версию, когда сочтут это нужным).

1. Охарактеризуйте многоуровневую модель качества программного обеспечения
2. Опишите объекты уязвимости. Дестабилизирующие факторы и угрозы надежности

**Объектами уязвимости**, влияющими на надежность ПС, являются:

• динамический вычислительный процесс обработки данных, автоматизированной подготовки решений и выработки управляющих воздействий;

• информация, накопленная в базах данных, отражающая объекты внешней среды, и процессы ее обработки;

• объектный код программ, исполняемых вычислительными средствами в процессе функционирования ПС;

• информация, выдаваемая потребителям и на исполнительные механизмы, являющаяся результатом обработки исходных данных и информации, накопленной в базе данных.

На эти объекты воздействуют различные **дестабилизирующие факторы**, которые можно разделить на внутренние, присущие самим объектам уязвимости, и внешние, обусловленные средой, в которой эти объекты функционируют. **Внутренними источниками** угроз надежности функционирования сложных ПС можно считать следующие дефекты программ:

• системные ошибки при постановке целей и задач создания ПС, при формулировке требований к функциям и характеристикам решения задач, определении условий и параметров внешней среды, в которой предстоит применять ПС;

• алгоритмические ошибки разработки при непосредственной спецификации функций программных средств, при определении структуры и взаимодействия компонентов комплексов программ, а также при использовании информации баз данных;

• ошибки программирования в текстах программ и описаниях данных, а также в исходной и результирующей документации на компоненты и ПС в целом;

• недостаточную эффективность используемых методов и средств оперативной защиты программ и данных от сбоев и отказов и обеспечения надежности функционирования ПС в условиях случайных негативных воздействий.

**Внешними дестабилизирующими факторами** отражающимися на надежности функционирования перечисленных объектов уязвимости в ПС, являются:

• ошибки оперативного и обслуживающего персонала в процессе эксплуатации ПС;

• искажения в каналах телекоммуникации информации, поступающей от внешних источников и передаваемой потребителям, а также недопустимые для конкретной информационной системы характеристики потоков внешней информации;

• сбои и отказы в аппаратуре вычислительных средств;

• изменения состава и конфигурации комплекса взаимодействующей аппаратуры информационной системы за пределы, проверенные при испытаниях или сертификации и отраженные в

эксплуатационной документации.

Полностью исключить все эти факторы невозможно. Поэтому необходимо разрабатывать средства и методы уменьшения их влияния на надежность ПС. Степень влияния всех внутренних дестабилизирующих факторов и некоторых внешних на надежность ПС в наибольшей степени определяется качеством технологий проектирования, разработки, сопровождения и документирования ПС.  
Методы предотвращения угроз надежности:  
- предотвращение ошибок проектирования;  
- систематическое тестирование;  
- обязательная сертификация.  
Методы повышения надежности:  
- временная избыточность;  
- информационная избыточность;  
- программная избыточность.

1. Охарактеризуйте методы предотвращения угроз надежности

В современных автоматизированных технологиях создания сложных ПС с позиции обеспечения их необходимой и заданной надежности можно выделить методы и средства позволяющие:

— создавать программные модули и функциональные компоненты высокого, гарантированного качества;

— предотвращать дефекты проектирования за счет эффективных технологий и средств автоматизации обеспечения всего жизненного цикла комплексов программ и баз данных;

— обнаруживать и устранять различные дефекты и ошибки проектирования, разработки и сопровождения программ путем систематического тестирования на всех этапах жизненного цикла ПС;

— удостоверять достигнутые качество и надежность функционирования ПС в процессе их испытаний и сертификации перед передачей в регулярную эксплуатацию;

— оперативно выявлять последствия дефектов программ и данных и восстанавливать нормальное, надежное функционирование комплексов программ.

Внешние дестабилизирующие факторы имеют различную природу. Современные достижения микроэлектроники значительно снизили влияние сбоев и отказов вычислительных средств на надежность функционирования программных средств. Однако ошибки персонала, искажения данных в каналах телекоммуникации, а также случайные (при отказах части аппаратуры) и необходимые изменения конфигурации вычислительных средств остаются существенными внешними угрозами надежности ПС. Негативное влияние этих факторов может быть значительно снижено соответствующими методами и средствами защиты и восстановления программ и данных.

Степень влияния всех внутренних дестабилизирующих факторов и некоторых внешних угроз на надежность ПС в наибольшей степени определяется качеством технологий проектирования, разработки, сопровождения и документирования ПС и их основных компонент. При ограниченных ресурсах на разработку ПС для достижения заданных требований по надежности необходимо управление обеспечением качества в течение всего цикла создания программ и данных.

Для обнаружения и устранения ошибок проектирования все этапы разработки и сопровождения программных средств должны быть поддержаны методами и средствами систематического тестирования. Тестирование – это основной метод измерения качества, определения корректности и реальной надежности функционирования программ на каждом этапе разработки. Результаты тестирования должны сравниваться с требованиями технического задания или спецификацией.

Кроме того, обеспечение качества ПС предполагает формализацию и сертификацию технологии их разработки, а также выделение в специальный процесс поэтапное измерение и анализ текущего качества создаваемых и применяемых компонент.

Итак, основные методы предотвращения угроз надежности:

предотвращение ошибок проектирования в CАSE-технологиях;

систематическое тестирование;

обязательная сертификация

Комплексное применение этих методов позволяет значительно ослаблять влияние угроз. Таким образом, уровень достигнутой надежности зависит от ресурсов, выделяемых на его достижение и от качества технологий, используемых на всех этапах жизненного цикла программного средства.

1. Охарактеризуйте оперативные методы повышения надежности: временная, информационная, программная избыточность

Чтобы обеспечить высокую надежность функционирования ПС, необходимы вычислительные ресурсы для максимально быстрого обнаружения проявления дефектов и выполнения автоматических мероприятий, обеспечивающих быстрое восстановления нормального функционирования ПС. Для этих целей используются следующие оперативные методы повышения надежности:

временная избыточность,

информационная избыточность,

программная избыточность.

**Временная избыточность** состоит в использовании некоторой части производительности компьютера для контроля исполнения программ и восстановления вычислительного процесса. Для этого при проектировании информационной системы должен предусматриваться запас производительности, который будет затем использоваться на контроль и оперативное повышение надежности функционирования. Величина временной избыточности зависит от требований к надежности и находится в пределах от 5-10% производительности компьютера до 3-4-кратного дублирования производительности отдельной машины в многопроцессорных вычислительных системах. Временная избыточность используется для контроля и обнаружения искажений, на их диагностику и выработку решений по восстановлению вычислительного процесса или информации, а также на реализацию операций восстановления.

**Информационная избыточность** состоит в дублировании накопленных исходных и промежуточных данных, обрабатываемых программами. Избыточность используется для сохранения достоверности данных, которые в наибольшей степени влияют на нормальное функционирование ПС и требуют значительного времени для восстановления. Их защищают 2-3 кратным дублированием с периодичным обновлением.

**Программная избыточность** используется для контроля и обеспечения достоверности наиболее важных решений по обработке информации. Она заключается в сопоставлении результатов обработки одинаковых исходных данных программами, различающимися используемыми алгоритмами, и в исключении искажений при несовпадении результатов. Программная избыточность необходима также для реализации программ автоматического контроля и восстановления данных с использованием информационной избыточности и для функционирования всех средств обеспечения надежности, использующих временную избыточность.

Средства оперативного программного контроля включаются после использования прикладных и сервисных программ, поэтому средства программного контроля обычно не могут обнаружить непосредственно причину возникновения искажения вычислительного процесса или данных и фиксирует только последствия первичного искажения, т.е. вторичную ошибку. Результаты первичного искажения могут приобрести катастрофический характер при запаздывании их обнаружения и локализации. Для обеспечения надежности дефекты нужно обнаружить с минимальным запаздыванием, при этом желательны минимальные затраты аппаратных ресурсов, поэтому используются иерархические схемы контроля, при которых последовательно используется несколько методов в порядке углубления контроля и увеличения затрат до выявления источника искажения. Целесообразно акцентировать ресурсы на потенциально наиболее опасных дефектах и достаточно частых режимов восстановления: при искажениях программ и данных, при перегрузках по производительности и при параллельном использовании программ.

1. Охарактеризуйте математические модели описания статистических характеристик ошибок в программах

Детальный анализ проявления ошибок показывает их глубокую связь с методами структурного построения программ, типом языка программирования, степенью автоматизации технологии проектирования и многими другими факторами. Статистические характеристики различных типов ошибок трудно описать математическими моделями, и более доступны для математического описания обобщенные характеристики ошибок в комплексе программ. Путем анализа и обобщения экспериментальных данных реальных разработок предложено несколько математических моделей, описывающих основные закономерности изменения *суммарного количества вторичных ошибок* в программах. Модели имеют вероятностный характер, и достоверность прогнозов в значительной степени зависит от точности исходных данных и глубины прогнозирования по времени. Эти математические модели предназначены для оценки:

1. надежности функционирования комплекса программ в процессе отладки, испытаний и эксплуатации;
2. числа ошибок, оставшихся невыявленными в анализируемых программах;
3. времени, требующегося для обнаружения следующей ошибки в функционирующей программе;
4. времени, необходимого для выявления всех ошибок с заданной вероятностью.

Точное определение полного числа невыявленных ошибок в комплексе программ *прямыми методами измерения невозможно.* Однако имеются пути для приближенной статистической оценки их полного числа или вероятности ошибки в каждой команде программы. Такие оценки базируются на построении математических Моделей в предположении о жесткой корреляции между общим количеством и проявлениями ошибок в комплексе программ после его отладки в течение времени T, т. е. между:

* суммарным количеством первичных ошибок в комплексе программ (n0) или вероятностью ошибки в каждой команде программы (p0);
* количеством ошибок, выявляемых в единицу времени в процессе тестирования и отладки при постоянных усилиях на ее проведение *{dn/dt);*
* интенсивностью искажений результатов в единицу времени (λ) на выходе комплекса программ вследствие невыявленных первичных ошибок при функционировании программ.

Наиболее доступно для измерения количество вторичных ошибок в программе, выявляемых в единицу времени в процессе отладки. Возможна также непосредственная регистрация отказов и наиболее крупных искажений результатов, выявляемых средствами оперативного контроля в процессе функционирования программ. Все три показателя связаны некоторыми коэффициентами пропорциональности, значения которых зависят, в частности, от интервала времени, на котором производится сопоставление, от быстродействия ЭВМ, от эффективности средств автоматизации отладки и от некоторых других параметров. При фиксированных условиях разработки и функционирования конкретного комплекса программ эти коэффициенты имеют вполне определенное значения. Для другой подобной системы коэффициенты могут несколько измениться, однако оценки, полученные для нескольких конкретных систем, позволяют прогнозировать эти характеристики, а следовательно, и соответствующие показатели надежности ПС в зависимости от длительности отладки и ряда других факторов.

Описаны несколько математических моделей, основой которых являются различные гипотезы о характере проявления вторичных ошибок в программах. Эти гипотезы в той или иной степени апробированы при обработке данных реальных разработок, и их можно разделить на три группы. В **первую группу** входят очевидные допущения, статистическая проверка которых невозможна и нецелесообразна. Вторую группу составляют допущения, определяющие специфические характеристики модели и требующие статистической проверки и обоснования на базе экспериментальных исследований. В третью группу включены второстепенные допущения, расширяющие и уточняющие возможности применения модели и частично доступные экспериментальной проверке.

**Первая группа** допущений включает предположение о *наблюдаемости искажений данных,* программ или вычислительного процесса, обусловленных первичными ошибками в программах. Первичная ошибка, являющаяся причиной искажения результатов, либо фиксируется и исправляется после завершения этапа тестирования, либо вообще не обнаруживается, так как проявление ее последствий незначительно.

Предполагается, что множество тестов более или менее *равномерно покрывает* все множество реальных исходных данных и отсутствуют априорные сведения для искусственного повышения интенсивности ошибок при некоторых тестах. Тем самым состав тестов представляется случайным относительно области изменения входных данных программы и содержащихся в ней необнаруженных первичных ошибок.

Наличие большого числа разнообразных данных, необходимых для исполнения программ, и практически некоррелированное их изменение приводит к *внешне случайному выбору маршрута,*по которому исполняется программа в каждом конкретном случае. В результате интенсивность проявления ошибок при реальном функционировании программ зависит от среднего быстродействия ЭВМ и практически не зависит от конкретного распределения типов команд на маршрутах обработки данных между ошибками.

*Коллектив специалистов,* их квалификация и загруженность предполагаются *постоянными* на интервале проектирования и исследования характеристик ошибок. Также постоянным считается доступное машинное время для проведения проверок программ.

**Вторая группа** допущений при построении математических моделей ошибок является основной и проверена интегрально по обобщенным характеристикам частости обнаружения ошибок и дифференцирование путем анализа правомерности каждого допущения. Ниже рассмотрены допущения при построении экспоненциальной модели.

*Интервалы времени* между обнаруживаемыми искажениями результатов предполагаются *статистически независимыми.* Время измеряется по фактической наработке длительностей исполнения программ τ без учета дополнительных затрат календарного времени на локализацию, диагностику и исправление ошибок.

Предполагается, что *интенсивность проявления ошибок остается постоянной,* пока не произведено исправление первичной ошибки или не изменена программа по другой причине. Если каждая обнаруженная ошибка исправляется, то значения интервалов времени между их проявлениями изменяются по экспоненциальному закону. Интегральная проверка распределения интервалов времени между обнаружениями ошибок показала, что оно достаточно хорошо аппроксимируется экспонентой.

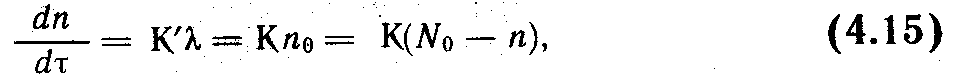
Логично предположить, что *интенсивность обнаружения ошибок пропорциональна суммарному числу первичных ошибок,*имеющихся в данный момент в комплексе программ. Это допущение подтверждено расчетом значений суммарного числа ошибок для хорошо отлаженных и переданных в эксплуатацию комплекса программ на ряд предыдущих моментов времени, когда проводилась отладка.

Каждая обнаруженная ошибка подлежит исправлению, поэтому предполагается, что *частота исправления ошибок пропорциональна частоте их обнаружения.* Однако некоторые исправления, в свою очередь, содержат ошибки. Кроме того, некоторые ошибки являются связанными, и при обнаружении проявления одной ошибки следует исправление нескольких первичных ошибок. Из-за этого частота обнаружения ошибок и частота их исправления не равны, а должны быть связаны некоторым коэффициентом пропорциональности. Коэффициенты корреляции для исследованных комплексов программ довольно высокие — от 0,52 до 0,82 при среднем значении около 0,68, т. е. достаточно хорошо подтверждают гипотезу.

**Третья группа** допущений детализирует использование ресурсов на корректировку программ и повышение**их**качества.

Приведенные предположения позволяют построить экспоненциальную математическую модель распределения моментов обнаружения ошибок в программах и установить связь между интенсивностью обнаружения ошибок при отладке *dn/dτ,* интенсивностью проявления ошибок при нормальном функционировании программ λ и числом первичных ошибок *п.* Предположим, что в начале отладки комплекса программ при *τ*=0 в нем содержалось N0 первичных ошибок. После отладки в течение времени осталось *п0*первичных ошибок и устранено *п* ошибок (n0 + n = N0*).* Время *τ* соответствует длительности исполнения программы на ЭВМ для обнаружения ошибок и не учитывает время, необходимое для анализа результатов и проведения корректировок. Календарное время *τ*к отладочных и испытательных работ с реальным комплексом программ значительно больше, так как после тестирования программ, на которое затрачивается машинное время т, необходимо время на анализ результатов, на обнаружение и локализацию ошибок, а также на их устранение.

При постоянных усилиях на отладку интенсивность обнаружения искажений вычислительного процесса, программ или данных вследствие еще не выявленных ошибок пропорциональна количеству оставшихся первичных ошибок в комплексе программ. Предположение о сильной корреляции между значениями *по* и *dn/dτ* проверено анализом реальных характеристик процесса обнаружения ошибок. Тогда

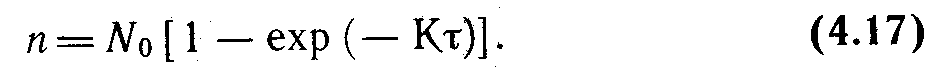
(11.1)\_

где коэффициенты К и К' учитывают масштаб времени, используемый для описания процесса обнаружения ошибок, быстродействие ЭВМ и другие параметры. Значение коэффициента К' можно определить как изменение темпа проявления искажений при переходе от функционирования программ на специальных тестах к функционированию на типовых исходных данных. В начале отладки это различие может быть значительным, однако при завершении отладки и при испытаниях тестовые данные практически совпадают с исходными данными при нормальной эксплуатации. Поэтому ниже К' полагается равным единице (К'=1).

Таким образом, интенсивность обнаружения ошибок в программе и абсолютное число устраненных первичных ошибок связываются уравнением

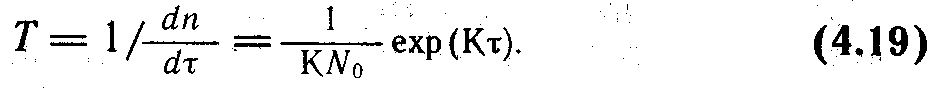
**(11.2)**

Так как выше предполагалось, что в начале отладки при *τ*= 0 отсутствуют обнаруженные ошибки, то решение уравнения (11.2) имеет вид

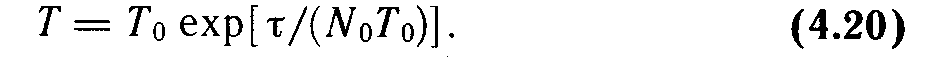
**(11.3)**

пропорционально интенсивности их обнаружения ***dn/dτ*** с точностью до коэффициента К.

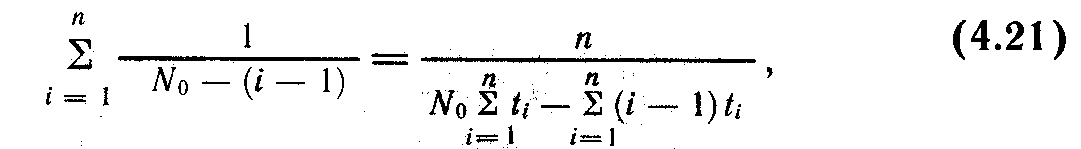
Наработка между проявлениями ошибок, которые, рассматриваются как обнаруживаемые искажения программ, данных или вычислительного процесса, равны величине, обратной интенсивности обнаружения ошибок:

**(11.4)**

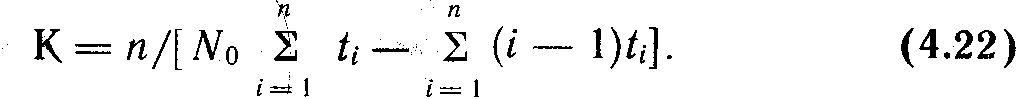
Если учесть, что до начала отладки в комплексе программ содержалось ***N0***первичных ошибок и этому соответствовала наработка T0 , то функцию наработки между проявлениями ошибок от длительно-

**(11.5)**

Если известны все моменты обнаружения ошибок***ti*** и каждый раз в эти моменты обнаруживается и достоверно устраняется одна первичная ошибка, то, используя метод максимального правдоподобия, получим уравнение для определения значения начального количества первичных ошибок ***N0***

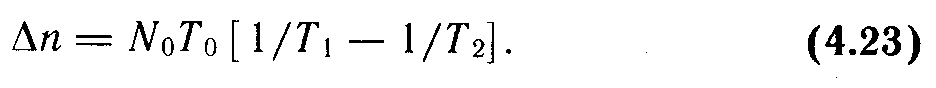
**(11.6)**

и выражение для расчета коэффициента пропорциональности

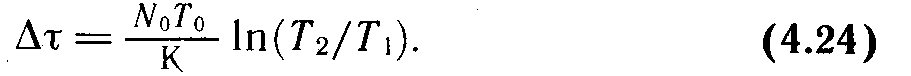
**(11.7)**

Необходимые для расчетов К и ***N0*** экспериментальные значения ***ti***определяются в процессе отладки данного или аналогичных комплексов программ, созданных тем же коллективом разработчиков и при такой же технологии. В результате можно рассчитать число оставшихся в программе первичных ошибок и среднюю наработку***Т*** до обнаружения следующей ошибки.

В процессе отладки и испытания программ для повышения наработки между проявлениями ошибок от величины **Т1** до значения **Т2** необходимо обнаружить и устранить **Δn** ошибок. Эту величину можно определить, выразив число обнаруживаемых ошибок через длительность наработки на проявление ошибок, для чего подставим (11.4) в {11.2). Тогда

(**11.8)**

Аналогичными несложными преобразованиями можно получить затраты времени на проведение отладки**Δτ,** которые позволяют устранить **Δn**ошибок и соответственно повысить наработку между очередными обнаружениями ошибок от значения **Т1** до **Т2:**

**(11.9)**

Следует подчеркнуть статистический характер приведенных соотношений. Неравномерность выбора маршрутов исполнения программы при нормальной эксплуатации, разное влияние конкретных типов ошибок в программах на проявление их при функционировании, а также сравнительно небольшие значения **n** и **Δn**, особенно на заключительных этапах отладки, приводят к тому, что флюктуации интервалов времени между обнаружением ошибок **Δτ** могут быть весьма значительными

1. Охарактеризуйте анализ рисков и характеристик качества программного обеспечения при внедрении.
2. Вредоносные программы: классификация, жизненной цикл, методы обнаружения

К *вредоносным программам*(иначе называемым *разрушающими программными воздействиями*, malware) относятся компьютерные вирусы и программные закладки.

Впервые термин *компьютерный вирус*ввел в употребление специалист из США Ф.Коэн в 1984 г.

Компьютерный вирус

Автономно функционирующая программа, обладающая одновременно тремя свойствами:

o способностью к включению своего кода в тела других файлов и системных областей памяти компьютера;

o последующему самостоятельному выполнению;

o самостоятельному распространению в компьютерных системах.

Программная закладка

Внешняя или внутренняя по отношению к атакуемой компьютерной системе программа, обладающая определенными разрушительными функциями по отношению к этой системе.

Классификация компьютерных вирусов

1. По способу распространения в компьютерной системе:

n файловые вирусы, заражающие файлы одного или нескольких типов;

n загрузочные вирусы, заражающие загрузочные сектора жестких дисков и дискет;

n комбинированные вирусы, способные заражать и файлы, и загрузочные сектора дисков.

1. По способу заражения других объектов компьютерной системы:

резидентные вирусы, часть кода которых постоянно находится в оперативной памяти компьютера и заражает другие объекты

нерезидентные вирусы, которые заражают другие объекты в момент открытия уже зараженных ими объектов.

1. По деструктивным возможностям:

n безвредные вирусы, созданные в целях «обучения», однако снижающие эффективность работы компьютерной системы за счет потребления ее ресурсов;

n неопасные вирусы, создающие различные звуковые и видеоэффекты;

n опасные и очень опасные вирусы, вызывающие сбои в работе программного и (или) аппаратного обеспечения компьютера, потерю программ и данных, а потенциально – вывод из строя аппаратуры КС и нанесение вреда здоровью пользователей.

1. По особенностям реализуемого алгоритма:

n вирусы-спутники, создающие для заражаемых файлов одноименные файлы с кодом вируса и переименовывающие исходные файлы (при открытии зараженного файла фактически открывается файл с кодом вируса, в котором после выполнения предусмотренных автором действий открывается исходный файл);

n паразитические вирусы, которые обязательно изменяют содержимое заражаемых объектов;

n вирусы-невидимки («стелс»-вирусы), в которых путем перехвата обращений операционной системы к зараженным объектам скрывается факт присутствия вируса в компьютерной системе (при собственном обращении к дисковой памяти вирусы-невидимки также используют нестандартные средства для обхода средств антивирусной защиты);

n вирусы-призраки (полиморфные вирусы), каждая следующая копия которых в зараженных объектах отличается от предыдущих (не содержит одинаковых цепочек команд за счет применения шифрования на различных ключах базового кода вируса).

Защита от руткитов

o Существующие сегодня специализированные программы, предназначенные для обнаружения руткитов, и традиционные антивирусы не дают стопроцентной гарантии безопасности.

o Обладая исходным кодом этих программ, можно создать любые модификации руткитов или включить часть кода в любую шпионскую программу.

o Главная цель руткитов не прочно закрепиться в системе, а проникнуть в нее.

Предотвращение внедрения программных закладок

Организационные меры:

o Минимизация времени работы в системе с полномочиями администратора.

o Создание отдельной учетной записи для работы в Интернете (с минимальными правами: запуск браузера и сохранение файлов в выделенной папке).

o Осмотрительная работа с почтовыми и офисными программами и др.

Методы предупреждения вирусного заражения

o физическое или логическое (для отдельных учетных записей) отключение накопителей для съемных дисков;

o разграничение прав отдельных пользователей и групп на доступ к папкам и файлам операционной системы и других пользователей;

o ограничение времени работы в компьютерной системе привилегированных пользователей;

o использование, как правило, только лицензионного программного обеспечения, приобретенного у официальных представителей фирм-правообладателей;

o выделение не подсоединенного к локальной сети компьютера для тестирования полученного из ненадежных источников программного обеспечения;

o использование встроенной в Microsoft Office защиты от потенциально опасных макросов, разрешающей в зависимости от установленного уровня выполнение макросов, содержащихся в документах из надежных расположений, подписанных доверенными издателями или разрешенных пользователем после получения соответствующего предупреждения.

Методы обнаружения компьютерных вирусов

1. Просмотр (сканирование) проверяемых объектов (системных областей дисковой и оперативной памяти, а также файлов заданных типов) в поиске сигнатур (уникальных последовательностей байтов) известных вирусов. Недостатки: необходимость постоянного обновления баз данных сигнатур известных вирусов, неспособность обнаружить новые компьютерные вирусы.
2. Инспекция (обнаружение изменений в объектах компьютерной системы) путем сравнения их вычисленных при проверке хеш-значений с эталонными (или проверки ЭЦП для этих объектов). Недостатки: не все изменения проверяемых объектов вызываются вирусным заражением, не может помочь при записи на жесткий диск компьютера пользователя уже зараженного файла.
3. Эвристический анализ – проверка системных областей памяти и файлов с целью обнаружения фрагментов исполнимого кода, характерного для компьютерных вирусов. Анализируются тысячи различных характеристик каждого файла. Недостатки: длительность процедуры проверки, возможность ложных сообщений о найденных вирусах.
4. Мониторинг − постоянное присутствие в оперативной памяти компьютера с целью сканирования всех открываемых или получаемых извне файлов и контроля всех «подозрительных» действий других программ (например, обнаружение и блокирование потенциально опасных результатов web-поиска, ссылок на URL). Недостатки: снижение эффективности работы системы, возможность выполнения контролируемых действий незараженными программами.
5. Вакцинирование – присоединение к защищаемому файлу специального модуля контроля, следящего за целостностью данного файла с помощью вычисления его хеш-значения и сравнения с эталоном. Недостатки: возможность обхода вирусами-невидимками , неприменимость для защиты файлов документов.
6. Блокирование потенциально опасных действий пользователя:
   1. Установка параметров безопасности и конфиденциальности в обозревателе Интернета.
   2. Установка защиты от записи в загрузочные сектора с помощью программы BIOS Setup.
   3. Определение недоступных для изменения областей дисковой памяти с помощью драйвера PCI-контроллера.

Удаление обнаруженных вирусов

1. Автоматическое − с помощью заранее разработанного алгоритма «лечения» зараженных известным вирусом объектов.
2. Автоматическое или «ручное» удаление зараженного объекта (с его последующим восстановлением по сохраненной незараженной резервной копии). Иногда возможно автоматическое восстановление объекта соответствующей программой (например, файла normal.dot программой Microsoft Office Word).
3. Полная переустановка системы, включая форматирование дисковой памяти, восстановление главного загрузочного сектора, установку операционной системы и прикладного программного обеспечения, восстановление файлов данных с резервных носителей информации.
4. Антивирусные программы: классификация, сравнительный анализ

Для защиты от вирусов можно использовать общие средства защиты информации, такие как дублирование информации, создание резервных копий, разграничение доступа. Разграничение доступа позволяет не только предотвратить несанкционированное использование информации, но и защитить данные от вредных действий вирусов, за счет ограничения доступа к файлам.

      Одним из самых удобных методов защиты от компьютерных вирусов является использование специализированных программ. Рассмотрим основные типы антивирусных программ.

* **Программы-детекторы** обеспечивают поиск и обнаружение вирусов в оперативной памяти и на внешних носителях, и при обнаружении выдают соответствующее сообщение. Различают детекторы *универсальные* и *специализированные*.
* **Программы-доктора (фаги)** не только находят зараженные вирусами файлы, но и "лечат" их, т.е. удаляют из файла тело программы вируса, возвращая файлы в исходное состояние. В начале своей работы фаги ищут вирусы в оперативной памяти, уничтожая их, и только затем переходят к "лечению" файлов. Среди фагов выделяют полифаги, т.е. программы-доктора, предназначенные для поиска и уничтожения большого количества вирусов. Учитывая, что постоянно появляются новые вирусы, программы-детекторы и программы-доктора быстро устаревают, и требуется регулярное обновление их версий.
* **Программы-ревизоры** относятся к самым надежным средствам защиты от вирусов. Ревизоры запоминают исходное состояние программ, каталогов и системных областей диска тогда, когда компьютер не заражен вирусом, а затем периодически или по желанию пользователя сравнивают текущее состояние с исходным. Обнаруженные изменения выводятся на экран монитора. Как правило, сравнение состояний производят сразу после загрузки операционной системы. При сравнении проверяются длина файла, код циклического контроля (контрольная сумма файла), дата и время модификации, другие параметры.
* **Программы-фильтры (сторожа)** представляют собой небольшие резидентные программы, предназначенные для обнаружения подозрительных действий при работе компьютера, характерных для вирусов.
* **Программы-вакцины (иммунизаторы)** - это резидентные программы, предотвращающие заражение файлов. Вакцины применяют, если отсутствуют программы-доктора, "лечащие" этот вирус. Вакцинация возможна только от известных вирусов. Вакцина модифицирует программу или диск таким образом, чтобы это не отражалось на их работе, а вирус будет воспринимать их зараженными и поэтому не внедрится. Существенным недостатком таких программ является их ограниченные возможности по предотвращению заражения от большого числа разнообразных вирусов.
* Обеспечение информационной безопасности систем является одним из первостепенных вопросов. В современном обществе особенно значительную роль играет защита информации, так как интернет кишит вирусами и даже самые простые из них могут нанести серьезный вред компьютеру и хранящимся на нем данным. Эти угрозы могут носить самый разнообразный характер - нарушать работу системы путем уничтожения важных системных файлов, красть важную информацию, пароли, документы. Это приводит к печальным последствиям - от переустановки системы до потери важных данных или денег. [1,2] Поэтому очень важным становится вопрос выбора антивирусной программы для компьютера, способной защитить важные данные. В этой статье мы рассмотрим некоторые популярные антивирусы и попытаемся выбрать среди них самый оптимальный для рядового пользователя (большая часть из них будут бесплатными, так как все же, именно, бесплатные антивирусы являются наиболее доступными для широкой аудитории пользователей). Итак, нами будут рассмотрены 4 антивируса - Avast Free Antivirus, Panda Antivirus, 360 Total Security, ESET NOD32. Начнем мы с представления и краткой информации о каждом из них.
* **Avast Free Antivirus** - один из самых известных бесплатных антивирусов. Avast включает в себя функцию AutoSandbox, автоматически помещающую подозрительные файлы в так называемую "песочницу", где можно полностью проанализировать файл с последующим лечением при надобности. Данная функция спасает большой процент информации от автоматического удаления, что в свою очередь позволяет предотвратить неполадки и ошибки, возникающие в результате случайного повреждения системных файлов. Еще у Avast есть удаленная поддержка, позволяющая пользователям подключаться друг к другу для взаимопомощи. В целом, Avast Free Antivirus - хороший выбор для среднестатистического пользователя, обеспечивающий хорошую безопасность и защиту системе.
* **360 Total Security**- чуть менее, чем Avast, но все же достаточно известный антивирус, делающий ставку на надежность и простоту, по словам его создателей. Он придется по душе самым разным категориям пользователей. Для новичков будет удобным сильная автоматизация функций антивируса, защищающих компьютер без непосредственного участия пользователя. Для опытных пользователей тут есть гибкие настройки приложения, возможность оптимизировать работу системы через антивирус и возможность создания нескольких профилей с разными настройками.
* **Panda Antivirus**позиционируется разработчиками, как простой в использовании, но надежный и результативный антивирус. Взглянем на его функционал. Большая часть функций автоматизирована, включая автоматическое сканирование компьютера на наличие угроз. Есть функция антивирусного загрузочного диска Panda Cloud Cleaner, позволяющая вылечить систему, когда она не может сама загрузиться из-за заражения.
* **ESET NOD32 -**достаточно эффективный антивирус с агрессивным подходом к обнаружению и нейтрализации угроз. Вот только некоторые его функции: несколько ступеней защиты от различных вариаций нежелательного программного обеспечения и вирусов; фаервол с возможностью кастомизации для шифрования соединения; родительский контроль; сканирование подключаемых устройств. К сожалению, такое обилие функций существенно повышает нагрузку на систему.
* Итак, для наглядности представим результаты проведенного анализа характеристик и функционала антивирусных программ в табличном виде (табл. 1).
* Из таблицы 1 мы видим, что если говорить о количестве различных функций, то безусловным лидером будет ESET NOD32, а наименее надежны Panda Antivirus и 360 Total Security. Теперь протестируем взаимодействие антивирусов с системой. Результаты тестов представлены в таблице 2.
* **Таблица 1**-**Результаты сравнительного анализа характеристик и функционала антивирусных программ**

|  |
| --- |
|  |
| Функционал антивируса | Avast Free Antivirus | 360 Total Security | Panda Antivirus | ESET NOD32 |  |
| Антивирусный сканер и антивирусный монитор | + | + | + | + |  |
| Защита персональных данных | + | + | + | + |  |
| Эвристический алгоритм | + | + | + | + |  |
| Наличие фаервола | - | - | - | + |  |
| Защита E-mail | + | - | - | + |  |
| Возможность работы в облаке | + | + | + | + |  |
| Система обнаружения и предотвращения вторжений | + | - | - | + |  |
| Антиспам | - | - | - | + |  |
| Система обновлений | + | + | + | + |  |
| Веб-защита | + | + | - | + |  |
| Поведенческий блокиратор | + | + | + | + |  |
|  |  |  |  |  |  |

* **Таблица 2**-**Результаты тестов на взаимодействие антивирусов с системой**

|  |
| --- |
|  |
| Категория тестирования | Avast Free Antivirus | 360 Total Security | Panda Antivirus | ESET NOD32 |  |
| Время загрузки системы | 2 мин. | 2 мин. | 1 мин. | >2 мин. |  |
| Время сканирования системных папок | 10 мин. | 18 мин. | 20 мин | 12 мин. |  |
| Загрузка процессора | 4 % | 5 % | 6 % | 11 % |  |
| Использование памяти | 41 mb | 122 mb | 40 mb | 110 mb |  |
|  |  |  |  |  |  |

* В тестах себя наиболее "легким" для системы показал **Avast Free Antivirus. 360 Total Security**и **Panda Antivirus**немного уступают в скорости сканирования, в то время как **ESET NOD32**почти не отстает. По использованию памяти фаворитами являются Avast и Panda. ESET NOD32 и 360 Total Security потребляют значительно больший объем памяти.
* Таким образом, самым оптимальным вариантом антивируса является Avast Free Antivirus, показавший хороший результат и при обзоре функционала, и при тестах. антивирусная обзор безопасность оптимальное
* ESET NOD32 также является хорошим вариантом, но, как было ранее сказано, существенно грузит систему.
* 360 Total Security и Panda Antivirus являются неплохим выбором для новичков благодаря автоматизации множества функций и приятному простому интерфейсу, но все же уступают первым двум антивирусам в плане настроек безопасности.

1. Антивирусные программы: Виды антивирусных программ, Встраивание антивирусов в BIOS компьютеров

У пользователей современных персональных компьютеров есть возможность свободного доступа ко всему аппаратному и программному обеспечению. Но как раз эта возможность и представляет главную опасность в виде заражения компьютерными вирусами. Под компьютерным вирусом понимается специальная программа, которая способна самостоятельно внедряться в другие в другие программные приложения, делать собственные копии и пересылать их в различные файлы, систему компьютера и разные сети для препятствия работе программ, удаления или нарушения целостности файловых систем и каталогов, организации препятствий нормальному функционированию компьютера. Вирусы возможно классифицировать по среде их распространения, а именно:

Вирусы сети. Передаются по разнообразным сетям, соединяющим компьютеры. Вирусы файлов. Заражают, как правило, разные модули, с расширением «исполняемые» (.exe). Возможно внедрение файловых вирусов и в файлы других типов, но, обычно, из таких файлов они редко могут выйти наружу и поэтому не размножаются. Вирусы загрузки. Засылаются в сектора загрузки дисков (Boot-сектор) или в сектора, которые содержат программу, загружающую системный диск (Master Boot Record). Комбинированные вирусы. Они способны заражать как файлы, так и сектора загрузки дисков. Макровирусы. Пишутся на языке высокого уровня и заражают документальные файлы приложений, имеющие внутренние языки и автоматизирующие работу (макроязыки). Примером можно считать программный набор Microsoft Office. Вирусы семейства троянов. Это программы, которые замаскированы под нужные пользователям приложения, но по сути это источники вирусов для компьютера. Виды антивирусных программ. Чтобы найти и удалить компьютерные вирусы, а также защитить компьютер от их проникновения, существует несколько типов специальных программ, позволяющих находить и обезвреживать вирусные вложения. Эти программы получили название антивирусных. Они делятся на следующие типы: Детекторные программы. Лечебные программы (доктора), или фаги. Программы для ревизии (ревизоры). Программы фильтрации (фильтры). Программы вакцинации, или программы иммунизации. Детекторные программы ищут характерные для данного типа вирусов признаки в оперативной памяти и в файлах, а при нахождении вируса информируют об этом пользователя специальным сообщением. К недостаткам этого типа антивирусов следует отнести то, что они способны обнаружить лишь известные данной программе вирусы. Лечебные программы, или доктора, а также программы вакцинации, способны обнаруживать заражение файлов и вылечивать их, то есть удалять из тела файла вирусную программу, предавая «больному» файлу первозданный вид. Но сначала доктора-антивирусы производят поиск заражений в оперативной памяти, очищают её от вирусов и лишь потом приступают к работе с фалами. Среди докторских антивирусов можно выделить полифаги, то есть антивирусные программы, которые предназначены для обнаружения и ликвидации большого числа вирусных заражений. Самые распространённые из них, это: Kaspersky Antivirus. Norton AntiVirus. Doctor Web. Но новые разновидности вирусов обнаруживаются практически постоянно, поэтому антивирусы детекторы и доктора довольно скоро стареют, и по этой причине необходимо периодически обновлять антивирус. Программы, выполняющие функции ревизора, можно отнести к одним из самых надёжных антивирусных средств. Ревизоры помнят начальное состояние приложений, каталогов и системных программ на диске в «здоровом» состоянии компьютера. В дальнейшем, по мере необходимости, производится сравнение текущего состояния компьютера с начальным. Если найдены какие-либо существенные отличия, то они отображаются на экране дисплея. Обычно, сравнение состояний выполняется непосредственно после запуска операционной системы. Процедура сравнения заключается в проверке длины файла и его контрольной суммы, проверке даты и времени модификаций, других характеристик. У программ с функцией ревизора присутствуют очень непростые алгоритмы работы, которые могут обнаружить и «невидимые» (стелс) вирусы, и способны увидеть отличия, внесённые вирусом в проверяемую программу. К этой категории программ можно отнести достаточно известную программу Kaspersky Monitor. Программы фильтрации (или фильтры, или программа-«сторож») являются относительно маленькими резидентными программами, которые предназначены для выявления неадекватных операций в работе компьютера, что характерно при заражениях. К таким операция можно отнести: Попытка корректировать файлы, имеющие расширения СОМ. и ЕХЕ. Операции, корректирующие атрибуты файла. Операции прямой записи по абсолютным адресам на диск. Попытки записи в секторы загрузки диска. Операция загрузки резидентной программы. Если обнаружено, что какая-то программа пытается выполнить одно из указанных выше действий, то антивирус выдаёт эту информацию в виде сообщения и запрашивает у пользователя варианты дальнейшей работы. Такие программы очень хороши, поскольку могут выявить заражения на ранних его стадиях, то есть ещё до размножения вируса. Но они не могут вылечить файлы и диски. Для этих целей необходимо использовать другие антивирусы, предназначенные именно для этого. К числу недостатков программ фильтрации следует причислить их назойливое внимание. К примеру, программа всё время сообщает о каждой попытке копировать исполняемый файл. Кроме того, возможно она будет конфликтовать с другими приложениями. Программы вакцинации являются резидентными программами, которые предотвращают проникновение вирусов в файлы. Вакцины используют, когда нет программы лечения (доктора), способной вылечить от данного вируса. Выполнить вакцинацию можно только от уже известного вируса. Вакцинация изменяет программу или диск так, что это не влияет на их работоспособность, но вирус считает это заражением и уже не выполняет проникновение в объект. На сегодняшний день антивирусы типа вакцин находят достаточно небольшое использование. Следует в заключение отметить, что вовремя обнаруженное заражение вирусными программами файлов и дискового пространства, а затем абсолютно полная ликвидация найденных вирусов на всех компьютерах гарантирует блокировку начала эпидемии вирусов остальных компьютерах.

В системные платы компьютеров тоже встраивают простейшие средства защиты от вирусов. Эти средства позволяют контролировать все обращения к главной загрузочной записи жестких дисков, а также к загрузочным секторам дисков и дискет. Если какая-либо программа пытается изменить содержимое загрузочных секторов, срабатывает защита и пользователь получает антивирусный модуль, который осуществляет мониторинг подозрительных действий в системе, выполняемых другими программами. Эти программы, перехватывающие вирусоопасные ситуации и сообщающие об их возникновении. Такие ситуации как вызов на открытие, запись в загрузочный сектор диска, попытка программы остаться резидентно в памяти

К достоинствам относится способность блокировать и обнаруживать соответствующее предупреждение.

Антивирусные программные комплексы. Обычно современный комплекс включает в себя монитор, сканер, ревизор и планировщик. Планировщик используется для координации работы.

1. Охарактеризуйте состав мероприятий по защите персональных данных. Профилактические меры защиты.

### **К мерам по внутренней защите персональных данных относятся следующие действия:**

• ограничение числа работников (с регламентацией их должностей), которым открыт доступ к персональным данным. Кого может включать этот перечень? Абсолютно всех, кто имеет доступ к личным делам, т. е. сотрудников отделов кадров или ответственных за кадровое делопроизвод-   
ство, работников бухгалтерии, секретарей-делопроизводителей, специалистов, которые заключают договоры с физическими лицами, а также инженеров, программистов, юристов;

• назначение ответственного лица, обеспечивающего исполнение организацией законодательства в рассматриваемой сфере;

• утверждение перечня документов, содержащих персональные данные;

• издание внутренних документов по защите персональных данных, осуществление контроля за их соблюдением;

• ознакомление работников с действующими нормативами в области защиты персональных данных и локальными актами; проведение систематических проверок соответствующих знаний работников, обрабатывающих персональные данные, и соблюдения ими требований нормативных документов по защите конфиденциальных сведений. Следует иметь в виду, что все сотрудники, которые имеют доступ к персональным данным других людей, должны быть ознакомлены с особенностями законодательства в области защиты персональных данных;

• рациональное размещение рабочих мест для исключения несанкционированного использования защищаемой информации;

• утверждение списка лиц, имеющих право доступа в помещения, в которых хранятся персональные данные;

• утверждение порядка уничтожения информации;

• выявление и устранение нарушений требований по защите персональных данных;

• проведение профилактической работы с сотрудниками по предупреждению разглашения ими персональных данных.

### **Среди мер по внешней защите персональных данных следует выделить такие:**

• введение пропускного режима, порядка приема и учета посетителей;

• внедрение технических средств охраны, программных средств защиты информации на электронных носителях и др.

Несмотря на то что законом не установлены конкретные требования к количеству и содержанию локальных актов, принимаемых в организации по вопросам обработки и защиты персональных данных, практика реализации данного нормативного акта сформировала необходимый минимум документов, которые должны быть разработаны в компании:

• общий документ, определяющий политику фирмы в отношении обработки персональных данных, например положение о персональных данных;

•  список лиц, обрабатывающих персональные данные;

• приказ о назначении сотрудника, ответственного за организацию обработки персональных данных. Указанное лицо должно осуществлять внутренний контроль за соблюдением компанией и ее работниками законодательства о персональных данных, в том числе требований к их защите, доводить до сведения персонала положения законодательства о персональных данных, локальных актов по вопросам их обработки, а также требования к защите таких данных, организовывать прием и обработку обращений и запросов субъектов персональных данных и (или) контролировать прием и обработку таких обращений и запросов;

• положение о правовых, организационных и технических мерах защиты персональных данных от неправомерного или случайного доступа к ним, их уничтожения, изменения, блокирования, копирования, предоставления, распространения, а также от иных неправомерных действий в отношении персональных данных. В данном положении рекомендуется прописать конкретные меры по защите персональных данных (введение пропускного режима, применение программных средств защиты информации – паролей, антивирусных программ, хранение персональных данных обособленно от других сведений, на отдельных материальных носителях и в специально оборудованных помещениях с ограниченным доступом и т. д.);

•  локальный акт, устанавливающий процедуры,направленные на предотвращение и выявление нарушений законодательства в сфере защиты персональных данных, устранение последствий таких нарушений. Так, в компании могут быть разработаны план мероприятий по внутреннему контролю безопасности персональных данных, инструкция о порядке проведения служебного расследования по фактам нарушений законодательства в сфере защиты персональных данных, вестись журнал антивирусных проверок и контроля работы с персональными данными, журнал обучения, инструктажа и аттестации по вопросам защиты персональных данных.

### **Какие контрольные органы вправе затребовать персональные данные**

Суды и другие правоохранительные органы могут беспрепятственно получать от компаний персональные данные сотрудников, клиентов или контрагентов без согласия последних. Но ответ на вопрос о том, имеют ли аналогичные права другие контролирующие структуры и какие именно, приходится искать не только в законах, но и в судебной практике.

1. Файрвол: задачи, сравнительный анализ, настройка

**Фаервол** (от английского firewall — противопожарная стена), он же брандмауэр (тот же перевод с немецкого brandmauer), он же межсетевой экран — компонент программного либо программно-аппаратного обеспечения, который фильтрует сетевой трафик, выполняет защитную функцию для фрагментов сети или отдельных узлов.  
Его работа напоминает пункт пропуска на границе государства, либо пост охраны при входе на закрытую территорию.

Брандмауэр можно представить как набор фильтров, через который последовательно проходит трафик. Каждый фильтр проверяет элементы потока на соответствие определенному правилу. Набор правил фильтрации(ruleset) определяет, какой пакет пересечет экран (операция allow – «разрешить»), а какой будет отброшен (операция deny – «отказать» без уведомления, что сервис недоступен, либо reject – «отклонить» с уведомлением).

Отбор пакетов происходит по одному из двух принципов:

1. Проходит все, что не запрещено. В правилах прописаны признаки пакетов, которые нельзя пропускать.
2. Проходит только то, что разрешено. В правилах прописано, какие пакеты можно пропускать.

Второй принцип дает большую безопасность ценой усложнения администрирования системы.

В зависимости от того, на уровне каких сетевых протоколов работает экран, его можно отнести к одному из следующих типов:

* пакетные фильтры (анализируют заголовки (но не передаваемые данные) пакетов, уязвимы перед фрагментированными пакетами и фальсификацией данных, например подменой IP адреса);
* шлюзы сеансового уровня (работают как посредники между защищаемым хостом и другими узлами сети, пропускают пакеты только в рамках установленного соединения, как и предыдущий тип, не анализируют содержимое пакетов, что является потенциальной уязвимостью);
* посредники прикладного уровня (также выступают посредниками в соединении, но дополнительно способны анализировать передаваемые данные, блокировать нежелательные команды, надежнее предыдущих, но более ресурсозатратны);
* инспекторы состояния (являются слиянием вышеперечисленных типов с поправками в пользу производительности).

Для брандмауэров может создавать затруднения шифрованный либо тунеллированный трафик, который невозможно проанализировать на соответствие фильтрам. Для подобных случаев самым безопасным решением будет отбрасывать такие данные.

Далее рассмотрим персональные фаерволы, программы, которые устанавливаются на ПК для защиты от сетевых угроз.

Персональные брандмауэры работают по тем же принципам, что и межсетевые, но имеют ряд особенностей:

* возможность определить, какое ПО пытается установить сетевое соединение;
* разные правила для разных пользователей;
* наличие «режима обучения», который позволяет вручную настроить разрешения и запреты сетевой активности для всех программ.

## Примеры известных программ-фаерволов

Операционная система имеет встроенного защитника, а многие антивирусные программы имеют фаервол в составе, но это не помешает вам при желании выбрать, установить и настроить брандмауэр на своё усмотрение.

Популярные бесплатные фаерволы для Windows:

* Comodo Firewall;
* Outpost Firewall Free;
* ZoneAlarm Free Firewall;
* PC Tools Firewall Plus;
* Emsisoft Online Armor Free;
* Privatefirewall;
* TinyWall (не так популярен как остальные, но не конфликтует со встроенным в систему брандмауэром и эффективен против троянских программ).

## Как настроить брандмауэр для Windows?

**Настройка фаервола** — комплексная задача, для выполнения которой нужен багаж знаний. Необходимо проработать последовательный набор правил-фильтров, которые могли бы обеспечить защиту и в то же время не мешать работе полезных программ. Каждое правило имеет ряд свойств, и все они должны быть установлены грамотно.

Доступен ряд методов упрощения задачи:

* использовать специальное ПО для настройки брандмауэра (**Windows Firewall Control** — бесплатная компактная утилита);
* использование других программ персональных фаерволов с упрощенной настройкой (как **PC Tools Firewall Plus**);
* работа в «режиме обучения», которая позволяет разрешать или запрещать доступ в каждом конкретном случае.

Отдельно разберем, как разрешать доступ для отдельных приложений:

1. Находим Брандмауэр Windows в панели управления.
2. Открываем «Разрешение взаимодействия с приложением или компонентом в брандмауэре Windows».
3. «Изменить параметры», «Разрешить другое приложение».
4. Вводим расположение программы, выбираем типы сетей, «Добавить».

## Как отключить брандмауэр в Windows?

При использовании другого полноценного фаервола встроенный можно выключить.

Самый быстрый способ — использовать командную строку:

* напишите в «Выполнить» (Win + R), команду «cmd»;
* в открывшемся поле пишете ***netsh advfirewall set allprofiles state off***и жмете «Enter»;
* готово, брандмауэр выключен.

Если потребуется, команда для включения выглядит так:

***netsh advfirewall set allprofiles state on***

**Outpost Firewall**— программа для защиты компьютера от хакерских атак из Интернета. Кроме этого, Аутпост обеспечивает блокировку загрузки рекламы и активного содержимого веб-страниц, а тем самым — их более быструю загрузку. От российской компании Agnitum.

Outpost Firewall — выпускается в двух вариантах: *Outpost Firewall Pro*(лицензия Shareware) и *Outpost Firewall Free*(лицензия Freeware)

Возможности программы

* Фильтрация входящих и исходящих сетевых соединений
* Глобальные правила для протоколов и портов
* Создания правил сетевого доступа для известных приложений на основе предустановок
* Политики блокировки задают реакцию Outpost на соединение, отсутствующее в правилах — автоматически отклонить его, разрешить или выдать запрос на создание правила, кроме того, блокировка/разрешение всех соединений
* Контроль компонентов, контроль скрытых процессов и контроль памяти процессов позволяют устанавливать ограничения на сетевую активность для отдельных приложений и процессов, определяя, какие именно — входящие или исходящие — соединения разрешены для конкретных приложений
* Визуальное оповещение о событиях (например, о блокировании соединения, попытке сетевой атаки) с помощью всплывающих окон
* Наглядное отображение сетевой активности
* Журнал действий программы, в том числе и подключаемых модулей
* Внутренняя защита (например, от попыток остановить сервис) и возможность задать пароль на изменение конфигурации

Подключаемые  модули

В состав дистрибутива Outpost Firewall Pro входят следующие плагины(программный модуль):

* *Реклама*— позволяет блокировать интернет-рекламу по ключевым словам и типичным размерам рекламных баннеров
* *Интерактивные элементы*— контролирует деятельность следующих активных web-элементов: ActiveX, Java-апплеты, сценарии Java Script и, Cookies, всплывающие окна,  скрытые фреймы, Flash, анимированные GIF
* *Детектор атак*— обнаруживает и блокирует попытки сетевых атак
* *Фильтрация почтовых вложений*— обнаруживает и переименовывает потенциально опасные вложения в электронной почте
* *DNS*— кэширование наиболее часто используемых записей DNS для ускорения доступа к ним
* *Разное*— блокировка сайтов с нежелательным содержимым по ключевым словам и URL
* *Anti-Spyware*— блокировка шпионского программного обеспечения и предотвращение отправки через сеть Интернет конфиденциальной информации

Награда: Outpost Firewall Pro был признан лучшим продуктом 2008 года в категории *персональный брандмауэр*по версии журнала Мир ПК

**Comodo Internet Security**- бесплатный файрвол, разработанный американской компанией. Это комплексное решение, включающее наряду с файрволом защиту от троянов и вирусов и другие функции. Программа включает огромное число разнообразных настроек. Платная версия имеет дополнительные функции по защите, оптимизации и настройке ПК и шифровке данных, передаваемых по беспроводным соединениям.

Возможности программы

1. Проактивная защита.
2. Защита от интернет-атак.
3. Защита от переполнения буфера.
4. Защита от несанкционированного доступа.
5. Защита важных системных файлов и записей реестра от внутренних атак.
6. Обнаружение переполнения буфера, которое происходит в HEAP памяти.
7. Обнаружение нападений ret2libc (Атака **возврата в библиотеку**(англ. *Return-to-libc attack*))

**Kaspersky Internet Security**— пакет программ для комплексной защиты персональных компьютеров. Программа обеспечивает: антивирусную защиту, защиту от сетевых атак, спама, неизвестных угроз, интернет-мошенничества и позволяет контролировать доступ пользователей компьютера к интернету.

Защита компьютера в реальном времени обеспечивается следующими компонентами защиты[1]:

* **Файловый Антивирус:**Контролирует файловую систему компьютера. Он проверяет все открываемые, запускаемые и сохраняемые файлы на вашем компьютере и на всех присоединенных дисках. Каждое обращение к файлу перехватывается Kaspersky Internet Security, и файл проверяется на присутствие известных вирусов. Дальнейшая работа с файлом возможна только в том случае, если файл не заражен или был успешно вылечен программой. Если файл по каким-либо причинам невозможно вылечить, он будет удален; при этом его копия будет сохранена в резервном хранилище, или помещён на карантин.
* **Почтовый Антивирус:**Проверяет все входящие и исходящие почтовые сообщения вашего компьютера. Он анализирует электронные письма на присутствие вредоносных программ. Письмо будет доступно адресату только в том случае, если оно не содержит опасных объектов. Кроме того, компонент анализирует почтовые сообщения на предмет фишинг-мошенничества.
* **Веб-антивирус:**Перехватывает и блокирует выполнение скрипта, расположенного на веб-сайте, если он представляет угрозу. Строгому контролю также подвергается весь HTTP-трафик. Кроме того, компонент анализирует веб-страницы на предмет фишинг-мошенничества.
* **IM-антивирус:**Обеспечивает безопасность работы с интернет-пейджерами. Компонент защищает информацию, поступающую на ваш компьютер по протоколам интернет-пейджеров. IM-антивирус обеспечивает безопасную работу со многими программами, предназначенными для быстрого обмена сообщениями.
* **Контроль программ:**Регистрирует действия, совершаемые программами в системе, и регулирует деятельность программ, исходя из того, к какой группе компонент относит данную программу. Для каждой группы программ задан набор правил. Эти правила регламентируют доступ программ к различным ресурсам.
* **Сетевой экран:**Обеспечивает безопасность вашей работы в локальных сетях и интернете. Компонент производит фильтрацию всей сетевой активности согласно правилам двух типов: правилам для программ и пакетным правилам.
* **Проактивная защита:**Позволяет обнаружить новую вредоносную программу ещё до того, как она успеет нанести вред. Компонент основан на контроле и анализе поведения всех программ, установленных на вашем компьютере. На основании выполняемых действий Kaspersky Internet Security принимает решение о том, является программа потенциально опасной или нет. Таким образом, ваш компьютер защищен не только от уже известных вирусов, но и от новых, ещё не исследованных.
* **Защита от сетевых атак:**Запускается при старте операционной системы и отслеживает во входящем трафике активность, характерную для сетевых атак. Обнаружив попытку атаки на компьютер, Kaspersky Internet Security блокирует любую сетевую активность атакующего компьютера в отношении вашего компьютера.
* **Анти-спам:**Встраивается в установленный на вашем компьютере почтовый клиент и контролирует все поступающие почтовые сообщения на предмет спама. Все письма, содержащие спам, помечаются специальным заголовком. Предусмотрена также возможность настройки Анти-спама на обработку спама (автоматическое удаление, помещение в специальную папку и т. д.). Также компонент анализирует почтовые сообщения на предмет фишинг-мошенничества.
* **Мониторинг сети:**Компонент, предназначенный для просмотра информации о сетевой активности в реальном времени.
* **Анти-фишинг:**Компонент, встроенный в веб-антивирус, анти-спам и IM-антивирус, который позволяет проверять веб-адреса на принадлежность к спискам фишинговых и подозрительных веб-адресов.
* **Анти-баннер:**Блокирует рекламную информацию, размещенную на специальных баннерах, встроенных в интерфейс различных программ, установленных на вашем компьютере, и находящихся в интернете.
* **Родительский контроль:**Компонент программы, выполняющий функции контроля доступа пользователей компьютера к веб-ресурсам. Основной задачей Родительского контроля является ограничение доступа, в первую очередь, к веб-сайтам, предназначенным для взрослой аудитории, затрагивающим темы порнографии, оружия, наркотиков, провоцирующим жестокость, насилие и т. д., а также к веб-сайтам, которые являются потенциальной причиной потери времени (чаты, игровые ресурсы) или денег (интернет-магазины, аукционы).
* **Безопасная среда:**Позволяет запускать приложения в безопасном для системы окружении (т. н. песочница). Это может быть использовано для запуска потенциально опасного ПО или для анонимного веб-серфинга.
* **Проверка ссылок:**Дополнение для браузеров Internet Explorer и Mozilla Firefox, проверяющее ссылки на просматриваемой веб-странице на предмет заражения веб-страниц, на которые они ссылаются

Нововведения в Kaspersky Internet Security 2011

**Мониторинг  активности:**Компонент защиты «Мониторинг активности» отслеживает активность программ в системе и предоставляет расширенную информацию другим компонентам защиты. Кроме того, благодаря сохраняемой истории активности программ, компонент может выполнять откат действий вредоносной программы при обнаружении вредоносной активности различными компонентами защиты.

* **Усовершенствованная функциональность Безопасной среды:**Безопасный запуск программ представляет собой изолированный рабочий стол, на котором вы можете запускать подозрительные программы без вреда для основной операционной системы.
* **Веб-фильтр:**Для усиления защиты при работе в интернете добавлены новые модули: Веб-фильтр — включает в себя модуль проверки ссылок, уже известный с предыдущей версии программы, а также предоставляет возможность блокировать доступ к небезопасным веб-сайтам, что позволяет вам оставаться в пределах безопасной зоны интернета.
* **Гео-фильтр:**Позволяет вам разрешить или запретить доступ к веб-сайтам на основе их принадлежности к доменам. Это помогает, например, запретить доступ к веб-сайтам, принадлежащим к региональным доменам с высокой степенью зараженности.
* **Контроль программ и Kaspersky Security Network:**Позволяет эффективнее определять статусы программ и настраивать правила для программ, используя данные Kaspersky Security Network, основанные на статистике работы Контроля программ на компьютерах множества пользователей.
* **Проверка во время простоя компьютера:**С помощью проверки во время простоя компьютера проверка компьютера на вирусы теперь может выполняться в те промежутки времени, когда вы не работаете за компьютером, и останавливаться, когда вы возвращаетесь к работе за компьютером. Это позволяет регулярно выполнять проверку и в то же время не снижать быстродействие компьютера тогда, когда он вам нужен.
* **Расширена функциональность Родительского контроля:**Теперь можно контролировать доступ пользователя к компьютеру и интернету, запуск пользователем компьютерных программ, ограничивать просмотр веб-страниц с нежелательным содержимым и загрузку файлов из интернета, контролировать общение пользователя в социальных сетях и через интернет-пейджеры, а также просматривать отчеты о действиях контролируемого пользователя. Для оптимизации настройки Родительского контроля предусмотрены экспорт и импорт параметров работы компонента для учетной записи.**Kaspersky Gadget:**При использовании программы на компьютере под управлением операционной системы Windows Vista или Windows 7 вам доступен Kaspersky Gadget.

Гаджет предназначен для быстрого доступа к основным функциям программы: индикации состояния защиты компьютера, проверке объектов на вирусы, просмотру отчетов о работе программы и т. д.

1. Охарактеризуйте методы и средства защиты компьютерных систем: Групповые политики. Аутентификация. Учетные записи

**Групповая политика** — это набор правил или настроек, в соответствии с которыми производится настройка рабочей среды приёма/передачи ([Windows](https://ru.wikipedia.org/wiki/Windows), X-unix и другие операционные системы с поддержкой сети). Групповые политики создаются в [домене](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BC%D0%B5%D0%BD_Windows_NT) и реплицируются в рамках домена. ***Объект групповой политики*** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Group Policy Object, GPO*) состоит из двух физически раздельных составляющих: ***контейнера групповой политики*** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Group Policy Container, GPC*) и ***шаблона групповой политики*** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Group Policy Template, GPT*). Эти два компонента содержат в себе все данные о параметрах рабочей среды, которая включается в состав объекта групповой политики. Продуманное применение объектов GPO к объектам каталога [Active Directory](https://ru.wikipedia.org/wiki/Active_Directory) позволяет создавать эффективную и легко управляемую компьютерную рабочую среду на базе ОС [Windows](https://ru.wikipedia.org/wiki/Windows). Политики применяются сверху вниз по [иерархии каталога](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%82%D0%B0%D0%BB%D0%BE%D0%B3_(%D1%84%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0)) Active Directory. По умолчанию в иерархии каталога Active Directory создаются две групповые политики: Default Domain Policy (политика домена по умолчанию) и Default Domain Controller’s Policy (политика контроллера домена по умолчанию). Первая из них назначается домену, а вторая — контейнеру, в состав которого входит [контроллер домена](https://ru.wikipedia.org/wiki/Active_Directory). Если вы хотите создать свой собственный объект GPO, вы должны обладать необходимыми полномочиями. По умолчанию правом создания новых GPO обладают группы Enterprise Administrators ([Администратор](https://ru.wikipedia.org/wiki/%D0%90%D0%B4%D0%BC%D0%B8%D0%BD%D0%B8%D1%81%D1%82%D1%80%D0%B0%D1%82%D0%BE%D1%80) предприятия) и Domain Administrators (Администраторы домена).

Стандартным примером аутентификации можно назвать вход пользователя в операционную систему Windows по паролю, когда после ввода пароля компьютер сравнивает данные, введённые с клавиатуры, с информацией, которая хранится в памяти машины. В случае если данные совпадают, процедура завершается успешно и пользователь получает разрешение на вход в компьютер.

Таким образом, аутентификация - процедура, в ходе которой пользователь должен предъявить системе секретную информацию, известную только ему одному.

Отметим, что механизм стандартной аутентификации на компьютере по паролю - достаточно несовершенен. К примеру, под легального пользователя может замаскироваться любой другой человек, который тем или иным образом узнал пароль владельца ПК. Именно поэтому вход в операционную систему Windows при помощи пароля не гарантирует надёжной защиты информации, хранящейся на вашем компьютере или ноутбуке.

Чтобы минимизировать подобные риски специалисты стали применять дополнительные виды информации, которые требуется предъявить при входе в систему. Такая информация может носить различный характер и называется факторами аутентификации.

**Три фактора аутентификации**

* Знание чего либо - пароль, парольная фраза, ПИН-код
* Обладание чем либо - физический [USB-токен](https://erim.ru/elektronnye-klyuchi/tokens/), [смарт-карта](https://erim.ru/elektronnye-klyuchi/kontaktnye-karty/), [ключ iButton](https://erim.ru/elektronnye-klyuchi/elektronnye-klyuchi-ibutton/)
* Биометрические характеристики - голос, рисунок сетчатки глаза, отпечаток пальца

Аутентификацию, которая использует только один из вышеназванных факторов, называют однофакторной. Аутентификацию, в процессе которой применяется несколько факторов, называют многофакторной.

Пример однофакторной аутентификации мы уже привели выше - это вход в компьютер с помощью пароля, а традиционным примером многофакторной аутентификации можно назвать использование банковской карточки для получения денежных средств в банкомате (двухфакторная аутентификация). В этом случае для того, чтобы снять деньги со своего банковского счёта, системе нужно предъявить не только банковскую карту, но и ввести правильный ПИН-код. Только после этого банкомат выдаст запрошенную денежную сумму.

Ещё один пример двухфакторной аутентификации - применение смарт-карты или [электронного ключа JaCarta](https://erim.ru/elektronnye-klyuchi/tokens/jacarta-pki/) или [eToken](https://erim.ru/elektronnye-klyuchi/tokens/elektronnye-klyuchi-etoken/) в комбинации с паролем для [входа в компьютер по USB-токену или смарт-карте](https://erim.ru/zaschita-informatsii/apparatnyy-dostup-k-pk-i-programmam/). В этом случае для загрузки компьютера требуется не только предъявить физический носитель (eToken или JaCarta), но и ввести его пароль, задействовав, таким образом, два фактора аутентификации.

Трёхфакторная аутентификация, в свою очередь, использует все три фактора, описанные выше. Очень часто её называют "аутентификацией типа 123". В настоящий момент трёхфакторная аутентификация является одним из самых надёжных инструментов защиты информации, хранящейся на компьютерах и ноутбуках.

Учетная запись или аккаунт на ПК означает совокупность личных сведений, которые пользователь вносит в компьютерную систему. Они доступны только ему. Это можно сравнить со своей комнатой, если вы проживаете в квартире с семьей. Как правило, учетка содержит данные, необходимые для идентификации личности в ходе подключения к системе, авторизации – логин и пароль.

В качестве логина обычно берется имя пользователя. Пароль же придумывают самостоятельно и состоит он из разных символов (цифр, букв, знаков). Для интернет-пользователей учетная запись то же самое, что Личный кабинет – онлайн-ресурс, где хранится персональная и прочая информация. Для большей защиты аккаунта от взлома часто предусмотрены дополнительные меры безопасности при прохождении аутентификации. К примеру, секретный вопрос, ответ на который знает только хозяин учетки.

Также в учетной записи часто содержатся дополнительные сведения о создателе странички: фотография, Ф. И. О., пол, псевдоним, национальность, группа крови, дата рождения, адрес проживания, № телефона и другие подробности.

Учетные записи бывают нескольких видов:

|  |  |
| --- | --- |
| **Тип учетки** | **Описание** |
| Администратор | Предоставляет полноценный контроль над ПК. |
| Стандарт | Предназначена для постоянного использования ПК. Внутри нее допустимо хозяйничать по своему усмотрению, но какие-то глобальные изменения, затрагивающие остальных пользователей компьютера, сделать невозможно. |
| Гость | Предполагает предоставление временного доступа к ПК постороннему человеку. Здесь пользователь имеет минимум прав. |

Если проще изъясняться, то создание УЗ – это процедура регистрации, необходимая для получения аккаунта на любой интернет-площадке (сайте, форуме и т. д.). После этого пользователю открывается доступ к функционалу сервиса.

## Как создаются учетные записи в Windows

Рассмотрим, как создать учетную запись на ПК с операционной системой Windows 7. В других версиях алгоритм будет таким же.

Последовательность действий:

1. Нажмите в панели задач на «Пуск» и выберите пункт «Панель управления», затем – «Добавление и удаление учетных записей..».
2. Откроется вкладка с учетными записями, где вы кликните по пункту «Создание учетной записи».
3. В следующем окошке укажите имя новичка и тип учетки. Нажмите на клавишу «Создание…».
4. На этом процедура создания УЗ завершена. Но, желательно запоролить аккаунт. Для этого войдите в него, далее снова запустите «Панель управления» и перейдите в «Учетные записи пользователей и семейная безопасность» - «Изменение пароля..».
5. В возникшем окошке обозначьте предполагаемые действия с УЗ. Щелкните по созданию пароля и вбейте его (придумываете самостоятельно). Аналогично можно поменять рисунок. Параметры, помеченные значком в виде щита, в состоянии изменить только администратор.

Аккаунт «Гость» по умолчанию отключен. Активировать его имеет право администратор ПК. Также у него нельзя поменять пароль и сменить тип.

## Как удалить учетку

В доступе администратора ПК удаление учетных записей иных пользователей. Для этого в панели управления задачами щелкните по записи «Удаление учетной записи». Система спросит, удалять ли личные файлы или сохранить. Во втором случае они останутся на рабочем поле в папке под названием учетки, которая была удалена.

1. Опишите процесс тестирования защиты программного обеспечения

Тестирование программного обеспечения является наиболее важной фазой жизненного цикла разработки программного обеспечения. Тестируемое программное обеспечение проходит различные фазы, которые, согласно исследованию, являются тестовым анализом: планирование тестирования; подготовка тестового примера данных, среды; выполнение теста; регистрация ошибок; отслеживание и закрытие; В прошлом было проведено много исследований для оптимизации всего процесса тестирования с целью улучшения качества программного обеспечения за минимальное время. После оценки всех доступных процессов тестирования было обнаружено, что для разных типов приложений используются разные модели разработки, и для тестирования используются разные методы тестирования. На основе исследований, проведенных в ходе изучения этой статьи, было проанализировано, что каждая компания модифицирует свой процесс тестирования в соответствии с потребностями и выполняет тестирование в зависимости от критичности приложений. Наиболее важные компоненты каждого приложения должны быть тщательно протестированы, чтобы убедиться, что их функции, производительность и функции безопасности работают должным образом. В этой статье говорится об обеспечении качества всех типов программных приложений путем выполнения определенных типов методов тестирования и оптимизированных процессов тестирования программного обеспечения. В соответствии с исследованиями и проведенными исследованиями типы тестирования могут быть разделены на три основных метода тестирования: функциональное тестирование, тестирование производительности и безопасности, а также основной процесс тестирования программного обеспечения, называемый анализом, подготовкой, выполнением и закрытием. Существует множество доступных типов тестирования, таких как тестирование «черного ящика», тестирование «белого ящика», тестирование на основе состояния, тестирование безопасности, тестирование на приемлемость, приемочное тестирование, системное тестирование, альфа- и бета-тестирование, а также тестирование на основе конфигурации, проверка и валидация. Основываясь на проведенных исследованиях и исследованиях, эта статья классифицировала их всех по трем типам тестирования высокого уровня, а именно: Функциональность, Производительность и Безопасность (FPS). Последний раздел посвящен заключению, которое показывает актуальность нашего оптимизированного процесса тестирования программного обеспечения и FPS как основа для методов тестирования. Оптимизированный процесс тестирования программного обеспечения Этапы STLC связаны с обнаружением и исправлением любой ошибки с использованием различных методов тестирования программного обеспечения. В этом разделе представлены обязательные этапы жизненного цикла тестирования, без которых жизненный цикл программного обеспечения не будет выполнен эффективно. Тестирование в основном дает критику или сравнение, которое определяет поведение системы в отношении ее спецификаций, механизмов, принципов, характеристик и соответствующих стандартов. Процесс тестирования программного обеспечения может быть настроен в соответствии с потребностями клиента или проекта. Процесс оптимизации, который можно использовать при тестировании программного обеспечения, — это анализ, планирование и подготовка, выполнение и закрытие. Процесс программного обеспечения обеспечивает поток системы и повышает гарантию продукта, который будет произведен. Этап анализа тестирования Первый этап, который является этапом анализа, является основным этапом процесса тестирования программного обеспечения. Эта фаза включает анализ функциональных и нефункциональных требований, например, бизнес-требования, функциональная документация и документ технической спецификации и т. д. Сбор требований должен быть согласован с клиентами, чтобы идентифицировать фактические и ожидаемые результаты тестирования, которые в основном являются нефункциональными требованиями, такими как: удобство использования, масштабируемость, тестируемость, производительность и безопасность. Все требования, которые не могут быть проверены из-за ограничений системы и среды тестирования, должны быть сообщены бизнес-команде. На этом этапе команда тестирования рассматривает и анализирует требования и определяет тесты, которые должны быть выполнены, и устанавливает приоритеты для тестирования — членов команды. Требования к среде тестирования включают требования к аппаратному и программному обеспечению, в соответствии с которыми необходимо тестировать необходимое программное обеспечение, и параллельно разработчики программного обеспечения начинают с планирования и разработки. Этап планирования и подготовки теста Этап подготовки теста включает подготовку плана тестирования, тестовый набор, данные теста и подготовку среды тестирования. План тестирования является первым документом, который должен быть подготовлен, в котором описываются объем, цели, характеристики, подлежащие тестированию, функции, которые не должны тестироваться, типы тестирования, которые должны быть выполнены, роли и обязанности группы тестирования, критерии входа и выхода и предположения. Одновременно команды тестирования начинают готовить тестовые наборы и тестовые данные. Тестовый пример — это документ, в котором описаны шаги, необходимые для тестирования любой функциональности с ожидаемым и фактическим результатом. Если фактический результат не совпадает с ожидаемым, открывается ошибка. Для каждого требования подготавливаются положительные и отрицательные тестовые примеры, что обеспечивается матрицей прослеживаемости требований (RTM). RTM — это документ, который сопоставляет требования с тестовыми примерами, чтобы гарантировать, что тестирование выполнено на 100 % Все действительные и недействительные наборы тестовых данных должны быть подготовлены для каждого тестового случая и подготовлен документ с тестовыми данными. Тестовые данные также генерируются на основе некоторых алгоритмов и инструментов. Подготовка тестового набора включает в себя различные этапы, которые начинаются с создания тестового набора, выбора тестового набора, оценки и определения приоритетов тестового набора. Эрик Д. Свайн предложил метод генерации тестовых случаев с использованием соответствующих диаграмм последовательности, а также определяет ограничения для определенных артефактов. В то же время методы генерации тестовых наборов полезны для выявления синхронизации и зависимости вариантов использования и сообщений, взаимодействия объектов и ошибок работы. Подготовка тестовой среды является одним из наиболее важных этапов, которые обычно готовятся отдельными средами обработки команды. После завершения части кода код проверяется инструментом управления конфигурацией, а затем готовится тестовая сборка, где тестировщики должны начать выполнение теста. Этап выполнения теста На этом этапе тестеры выполняют программное обеспечение в соответствии с контрольными примерами. Везде, где фактические и ожидаемые результаты не совпадают, тестировщик открывает ошибки и назначает их разработчикам. Закрытие теста Закрытие теста — это важный этап, который включает в себя все отчеты о тестировании, гарантирующие, что вся система, интеграция, приемочное тестирование пользователя пройдены, и принято решение о том, все ли требования проверены, и нет критической ошибки, ожидающей исправления ИЛИ проверки. Просмотр всех тестовых артефактов выполняется менеджером. Как только все артефакты рассмотрены и утверждены, выполняется выпуск программного обеспечения. Проводится дальнейший анализ первопричин, чтобы провести мозговой штурм по поводу того, что прошло хорошо, что не удалось, и областей улучшения. Существуют различные типы функционального тестирования, которые могут выполняться на различных уровнях тестирования, то есть модульное тестирование, интеграционное тестирование (тестирование сверху вниз и тестирование снизу вверх) и тестирование системы. Существует много тестов, проводимых на различных уровнях тестирования, таких как тестирование черного ящика, тестирование белого ящика, тестирование серого ящика, регрессионное тестирование, нечеткое тестирование, тестирование варианта использования, предварительное тестирование, тестирование на дым., тестирование работоспособности, приемочное тестирование, альфа-тестирование, бета-тестирование и т. д.

1. Охарактеризуйте средства и протоколы шифрования сообщений

Конфиденциальность – данные не могут быть получены или прочитаны неавторизованными пользователями. Целостность информации – уверенность в том, что информация 100% останется нетронутой и не будет изменена злоумышленником. Доступность информации – получение доступа к данным, когда это необходимо. Шифрование – преобразование информации в целях сокрытия от неавторизованных лиц, но в то же время с предоставлением авторизованным пользователям доступа к ней. Средства шифрования  - аппаратные, программные и программно-аппаратные шифровальные(криптографические) средства, реализующие алгоритмы криптографического преобразования  информации для ограничения доступа к ней, в том числе при ее хранении, обработке и передаче (Постановление Правительства РФ от 16.04.2012 N 313). Основные способы шифрования: Симметричное Асимметричное Хеширование Цифровая подпись. Основные способы шифрования: Симметричное Асимметричное Хеширование Цифровая подпись. Коммуникационный протокол устанавливает последовательность действий участников при передаче информации или информационном обмене. Обычный коммуникационный протокол обеспечивает установку сеанса, выбор маршрута, обнаружение искажений и восстановление передаваемой информации и т. п. Безопасность протокола выражается в обеспечении гарантий выполнения таких свойств, характеризующих безопасность, как доступность, конфиденциальность, целостность и др. Протокол, обеспечивающий поддержку хотя бы одной из функций-сервисов безопасности, называется защищённым или, точнее, протоколом обеспечения безопасности ( security protocol ). Защитные механизмы либо дополняют, либо встраиваются в коммуникационный протокол. Аутентификация источника данных Аутентификация сторон Конфиденциальность  данных Невозможность отказа Невозможность отказа с доказательством получения Невозможность отказа с доказательством источника Целостность данных Обеспечение целостности соединения без восстановления Обеспечение целостности соединения с восстановлением Разграничение доступа. 1.Классификация по числу участников: двусторонний; трёхсторонний; многосторонний. 2.Классификация по числу передаваемых сообщений: интерактивный (есть взаимный обмен сообщениями); не интерактивный (только однократная передача ). 3.Классификация по целевому назначению протокола : протокол обеспечения целостности сообщений; с аутентификацией источника; без аутентификации источника; протокол (схема) цифровой подписи; протокол индивидуальной / групповой цифровой подписи; с восстановлением / без восстановления сообщения; протокол цифровой подписи вслепую; протокол цифровой подписи с доказуемостью подделки; протокол идентификации (аутентификации участников); односторонней аутентификации; двусторонней (взаимной) аутентификации ; 3.Классификация по целевому назначению протокола : конфиденциальная передача; обычный обмен сообщениями; широковещательная / циркулярная передача; честный обмен секретами; забывающая передача; протокол привязки к биту (строке); протокол распределения ключей; протокол (схема) предварительного распределения ключей; протокол передачи ключа (обмена ключами); протокол совместной выработки ключа (открытого распределения ключей); протокол парный / групповой; протокол (схема) разделения секрета; протокол (распределения ключей для) телеконференции. Классификацию криптографических протоколов можно проводить также и по другим признакам: по типу используемых криптографических систем: на основе симметричных криптосистем; на основе асимметричных криптосистем; смешанные; по способу функционирования: интерактивный / неинтерактивный ; однопроходный / двух- / трёх- и т. д. проходный ; протокол с арбитром (протокол с посредником); двусторонний / с доверенной третьей стороной (с центром доверия). Групповые протоколы предполагают одновременное участие групп участников, например: протокол разделения секрета ( secret sharing protocol ) — если все группы, имеющие на это право, формируют одинаковые ключи; протокол телеконференции — если у различных групп должны быть разные ключи; протокол групповой подписи ( group signature protocol ) — предполагается одновременное участие заранее определенной группы участников, причем в случае отсутствия хотя бы одного участника из группы формирование подписи невозможно. Примитивный криптографический протокол ( primitive cryptographic protocol ) — это криптографический протокол, который не имеет самостоятельного прикладного значения, но используется как базовый компонент при построении прикладных криптографических протоколов. Как правило, он решает какую-либо одну абстрактную задачу. Например, протокол обмена секретами, протокол привязки к биту, протокол подбрасывания монеты (по телефону ). Прикладной криптографический протокол ( application cryptographic protocol ) предназначен для решения практических задач обеспечения функций — сервисов безопасности с помощью криптографических систем. Прикладные протоколы, как правило, обеспечивают не одну, а сразу несколько функций безопасности. Такие протоколы, как IPsec, на самом деле являются большими семействами различных протоколов, включающими много разных вариантов для различных ситуаций и условий применения. Примерами прикладных протоколов являются: система электронного обмена данными; система электронных платежей; система электронной коммерции; поддержка правовых отношений; игровые протоколы. Атаки, направленные против криптографических алгоритмов Атаки против криптографических методов, применяемых для реализации протоколов Атаки против самих протоколов (активные или пассивные). Аутентификация ( нешироковещательная ): аутентификация субъекта аутентификация сообщения защита от повтора Аутентификация при рассылке по многим адресам или при подключении к службе подписки/уведомления: неявная (скрытая) аутентификация получателя аутентификация источника Авторизация  (доверенной третьей стороной ). Свойства совместной генерации ключа: аутентификация ключа подтверждение правильности ключа защищенность от чтения назад формирование новых ключей защищенная возможность договориться о параметрах безопасности Конфиденциальность Анонимность: защита идентификаторов от прослушивания ( несвязываемость ) защита идентификаторов от других участников. Ограниченная защищенность от атак типа «отказ в обслуживании» Инвариантность отправителя Невозможность отказа от ранее совершенных действий: подотчётность доказательство источника доказательство получателя Безопасное временное свойство.

1. Опишите мероприятия по техническому обеспечению безопасности персональных данных
2. Охарактеризуйте основные мероприятия обеспечения безопасности персональных данных.

**Основные мероприятия по обеспечению безопасности персональных данных в учреждениях образования**

Исходя из требований законодательства образовательным учреждениям в течение 2009 года необходимо.

1. Определить (или уточнить) состав и категории обрабатываемых персональных данных.

2. Осуществить (или уточнить) классификацию действующих информационных систем, обрабатывающих персональные данные.

3. Провести необходимые организационные и технические мероприятия для обеспечения защиты:

- персональных данных, обрабатываемых без использования средств автоматизации;

- информационных систем, обрабатывающих персональные данные.

4. Декларировать соответствие или провести аттестационные (сертификационные) испытания информационных систем, обрабатывающих персональные данные.

Мероприятия по обеспечению безопасности персональных данных осуществляются на основе законодательства Российской Федерации, нормативных и методических документов.

В части предварительных организационных мероприятий по защите персональных данных всем подведомственным Рособразованию учреждениям и организациям следует:

- определить перечень, цели и порядок обработки персональных данных,

- назначить ответственных за работу с персональными данными,

- подготовить должностные инструкции сотрудников, обрабатывающих персональные данные,

- обеспечить размещение и охрану средств хранения и обработки персональных данных.

Для информационных систем классов К1 и К2 дополнительно потребуется принять предусмотренные методическими документами ФСТЭК России и ФСБ России меры по защите информации от утечки по техническим каналам.

1. Приведите примеры и опишите программные средства резервного копирования данных.

На сегодняшний день существует множество программных продуктов для обеспечения технологии резервного копирования данных.

Рассмотрим следующие программы:

- Acronis TrueImage Home.

- Paragon DriveBackup Server Edition.

- Symantec Backup Exec.

- DejaDup.

**Acronis TrueImage Home**

Acronis True Image Home - популярная программа для резервного копирования данных. Программа решает проблему резервного копирования данных, гарантируя полную сохранность всей информации, хранящейся на жестких дисках компьютера. С ее помощью можно производить резервное копирование отдельных файлов и папок, любых категорий пользовательских данных по выбору, а также целых дисков или их разделов. В случае какого-либо сбоя, нарушившего доступ к информации или работу системы, а также в случае нечаянного удаления нужных файлов, легко можно восстановить работу системы и утраченную информацию.

Возможности программы:

1) Создание точной копии жесткого диска;

2) Резервное копирование сообщений электронной почты, параметров и данных Outlook;

3) Защита параметров конфигурации установленных приложений;

4) Резервное копирование и восстановление данных по категориям: музыка, видео, документы и т.п.;

5) Try&Decide™ (Пробный режим): удобный инструмент для безопасной установки и испытания новых приложений;

6) Автоматическое резервное копирование по расписанию: в определенное время или при возникновении определенных событий;

7) Защита личных данных.

**Paragon DriveBackup Server Edition**

Paragon DriveBackup Server Edition - надежная защита операционной системы и данных жесткого диска как персональных компьютеров, так и корпоративных сетей. Помимо инструментов резервного копирования и восстановления, пользователь получает уникальную возможность управлять задачами резервного копирования удаленно, на других сетевых компьютерах. PARAGON DriveBackup предоставляет полный набор инструментов для создания бэкапа, восстановления и копирования жестких дисков и их разделов. Используя программу, можно осуществить резервное копирование данных, в том числе автоматическое резервирование диска, тем самым надежно сохранив архив на удобном носителе, а также создать Recovery CD, позволяющий запустить систему и восстановить информацию после любого системного сбоя.

Возможности программы:

1) Резервное копирование Windows систем без прерывания работы с использованием Microsoft Volume Shadow Copy Service (MS VSS) или Paragon Hot Processing;

2) Организация автоматизированной системы защиты данных, полностью отвечающей политике резервного копирования «настроить и забыть»;

3) Эффективные инструменты миграции Windows систем на различные аппаратные платформы, как физические, так и виртуальные (P2V, P2V Восстановление, V2P, V2V, P2P), способствуют минимизации затрат на обновление центров хранения данных, позволяет более рационально использовать ресурсы аппаратного и программного обеспечения;

4) Комплексные среды восстановлении данных на основе Linux/DOS и WinPE\*\*\* доступны на CD/DVD/Blu-ray, флэш-носителе или из Архивной капсулы, позволяют быстро восстановить работоспособность системы. Кроме того, с их помощью можно извлечь информацию с поврежденного жесткого диска или выполнить развертывание архива на машину без предустановленной операционной системы

5) Интеграция всего функционала виртуализации компании, включая уникальную технологию монтирования виртуальных дисков для проведения любого типа миграции и обслуживания виртуальных сред;

6) Восстановление на гранулярном уровне сокращает время недоступности ресурсов и сервисов, если нет необходимости производить полное восстановление данных из архива;

7) Устранение проблем загрузки без восстановления данных (изменение реестра ОС Windows в режиме оффлайн, коррекция MBR, BCD, файла Boot.ini и т.д.).

**Symantec Backup Exec**

BackupExec 2012 представляет собой единый интегрированный продукт, предназначенный для защиты виртуальных и физических сред, упрощения процедур резервного копирования и расширения возможностей восстановления. Благодаря поддержке технологии Symantec V-Ray продукт BackupExec 2012 позволяет восстанавливать целые серверы, важные приложения Microsoft и виртуальные среды VMware или MicrosoftHyper-V, существенно сокращая время простоя.

Возможности программы:

1) Простое восстановление виртуальных систем, приложений, баз данных, файлов и папок или отдельных объектов из единой резервной копии за считанные секунды благодаря запатентованной технологии V-Ray;

2) Резервное копирование виртуальных систем без агента;

3) Интегрированные средства восстановления «с нуля» на другой аппаратной платформе с преобразованием физической среды в виртуальную (P2V);

4) Лучшие в своем классе средства дедупликации данных позволяют оптимизировать любую стратегию резервного копирования за счет дедупликации в системе клиента, на сервере резервного копирования или в программно-аппаратных комплексах;

5) Передовая обновленная консоль администрирования обеспечивает более эффективную настройку резервных копий, управление политиками резервного копирования, аварийное восстановление и преобразование резервных копий серверов в виртуальные системы для немедленного аварийного восстановления;

6) Защита самых различных операционных систем, платформ, приложений и баз данных в виртуальных и физических средах с поддержкой устройств хранения на дисках и магнитных лентах.

**Deja Dup**

**Deja Dup** - это простой комплект инструментов для создания резервных копий файлов, обеспечивающий «правильную» реализацию выполнения резервных копий (шифрование и сжатие данных, хранение на удалённых серверах и периодичность этого процесса), в качестве основы приложения используется duplicity (мощная и компактная консольная утилита для автоматического шифрования и создания бекапов).

Возможности Deja Dup:

1) Есть возможность делать резервные копии: на локальном жёстком диске, USB, на удаленном ресурсе - FTP, WebDAV, а также поддержка облачных хранилищ: Amazon S3, UbuntuOne;

2) Можно выставить расписание резервного копирования;

3) Есть возможность выбора или исключения каталогов для резервного копирования;

4) Имеет архивирование / сжатие данных;

5) А также инкрементальное резервное копирование, которое позволяет восстанавливать файлы из отдельных резервных копий;

6) Есть выбор по времени хранения данных: минимум 6 месяцев, минимум год, вечно

1. Классификация программ резервного копирования. Опишите наиболее популярные программы резервного копирования. Проведите сравнение программ.

**Классификация программ резервного копирования**

Задача резервного копирования (backup) заключается в создании копии данных на любом другом носителе информации (магнитной ленте, жёстком диске, flash носителе и т.п.), предназначенной для последующего восстановления в оригинальном или новом месте их расположения в случаях повреждения, потери или кражи.

Основные требования, предъявляемые к *программам резервного копирования*:

1. ***Внедрение****: возможность быстрой установки и настройки программного обеспечения. Важным фактором для этого может служить: доступный русскоязычный мануал, наличие поддержки от производителя или интегратора, доступность материалов в интернете, что является показателем популярности решения, наличие минимум специфических настроек, понятные интерфейсы и др.*
2. ***Эксплуатация****: минимизация трудозатрат человека в процессе резервного копирования и восстановления данных, автоматизации процессов резервного копирования и восстановления данных в случаях сбоя.*
3. ***Функциональные характеристики****: скорость работы, поддержка различных физических и виртуальных платформ, поэлементное восстановление приложений и баз данных, дедупликация резервных копий, централизованное управление, поддержка ленточных и дисковых систем хранения, масштабируемость решения, безопасность хранение данных.*
4. ***Стоимость решения****: начальная стоимость приобретения программного обеспечения, стоимость ежегодной поддержки, стоимость расширения при увеличении парка систем.*

Исходя из основных требований к системам резервного копирования можно составить общую классификацию решений. Имея в данной области наколенный опыт и знания, попробуем глобально определить несколько категорий:

* ***Целевая аудитория и назначение****: частные решения для домашних пользователей; корпоративные решения для небольших частных предприятий; решения для крупных международных компаний или государственных структур с большой и разнообразной аппаратной и системной инфраструктурой.*
* ***Уровень надежности системы****: надежный разработчик, с большим опытом и авторитетом, решение проверенное временем с неоднократным выходом новых релизов.*
* ***Масштабируемость решения****: возможность расширения количества серверов, приложений, виртуальный сред, баз данных.*
* ***Функциональные характеристики****: возможность дифференциального, инкрементального или в виде образа резервного копирования, работа с виртуальными машинами, выборочное восстановление данных, функции дедупликации и архивирования, возможность автоматического аварийного восстановления.*

На сегодняшний день на рынки систем резервного копирования представлены более двух десятков программных и аппаратных решений от различных производителей. Но лишь часть из них представляет действительный интерес для применения в масштабах корпоративных систем. В нашей статье мы рассмотрим программные решения от компании Symantec, EMC, Acronis, Veeam и CommVault. Они по нашему мнению и мнению сообщества являются наиболее интересными и популярными продуктами для корпоративного сегмента.

резервного копирования и восстановления данных, *Symantec Backup Exec* является наиболее популярным решением как в России так и за ее рубежом. Symantec Backup Exec с новой технологией V-Ray (для виртуальных серверов), предоставляет максимально полное решение, предназначенное для различных ИТ сред любого масштаба и сложности. Это решение предлагается в трех вариантах: программное обеспечение для установки на существующем оборудовании, программно-аппаратный комплекс или в виде услуги как облачный сервис.

Функциональные характеристики: масштабируемая, аппаратно-независимая, клиент-серверная архитектура с функциями централизованного управления, аварийное восстановление после сбоя, поэлементное восстановление данных, интегрированная технология дедупликации и репликации данных, резервное копирование физических и виртуальных систем, русскоязычный интерфейс, шифрование бэкапируемых данных.

**Acronis Backup & Recovery**

обеспечивает резервное копирование и восстановление серверов, портативных и настольных компьютеров. Решение в основном предназначено для небольших предприятий с ограниченным или вовсе отсутствующим ИТ-персоналом.

К основным возможностям программы можно отнести: восстановление виртуальных машин, операционных систем, приложений и файлов, поддержка различных систем хранения резервной копии (FTP, NAS, SAN, ленточные устройства и облачное хранилище Acronis), шифрование резервной копии, функции автоматизации резервного копирования и восстановления.

**Veeam Backup & Replication**

 Решение *Veeam Backup & Replication* разработано специально для виртуальной среды — оно не использует агентов и обеспечивает быстрое резервное копирование и восстановление данных виртуальных машин на базе VMware vSphere или Microsoft Hyper-V независимо от операционной системы или типа хранилища. Используя один файл лицензии и одну консоль, можно обеспечить защиту всей виртуальной инфраструктуры с помощью таких инновационных возможностей, как мгновенное восстановление виртуальных машин, восстановление отдельных файлов и объектов приложений, практически непрерывная защиты данных и многих других.

Лицензирование и поддержка *Veeam Backup & Replication* значительно проще и дешевле, чем у традиционных средств резервного копирования. Решение лицензируется по количеству сокетов и предоставляет возможности резервного копирования и репликации, дедупликацию, поддержку нескольких гипервизоров и централизованное управление. Цена решения не зависит от количества копируемых приложений, виртуальных машин или хостов. Veeam Backup & Replication обеспечивает комплексную защиту виртуальной инфраструктуры, а его цена составляет лишь долю от стоимости других инструментов.

Отличительные особенности: восстановление виртуальных машин и любых виртуализованных приложений, автоматическая верификация возможности восстановления из каждой резервной копии.

1. Дать определение понятия политики безопасности. Описать обеспечение ИБ в нормальных ситуациях и обеспечение ИБ в чрезвычайных ситуациях.

Организационные мероприятия по защите процессов перера­ботки информации заключаются в разработке и реализации ад­министративных и организационно-технических мер при подго­товке и эксплуатации системы в нормальных условиях.

Организационные меры, по мнению зарубежных специалистов, несмотря на постоянное совершенствование технических мер, составляют значительную часть (около 50 %) системы защиты. Они применяются в случаях, когда ВС не может непосредственно кон­тролировать использование информации. Кроме того, для повышения эффективности защиты полезно иногда продублировать технические меры организационными.

Организационные меры по защите систем процессов перера­ботки информации должны охватывать этапы проектирования, разработки, изготовления, испытаний, подготовки к эксплуата­ции и саму эксплуатацию системы.

В соответствии с требованиями технического задания в орга­низации-проектировщике наряду с техническими средствами раз­рабатываются и внедряются организационные мероприятия по защите процессов переработки информации на этапе создания системы.

Под *этапом создания*понимаются проектирование, разработ­ка, изготовление и испытание системы. При этом следует отли­чать мероприятия по защите процессов переработки информа­ции, проводимые организацией-проектировщиком, разработчиком и изготовителем в процессе создания системы и рассчитан­ные на защиту от утечки информации в данной организации, и мероприятия, закладываемые в проект и разрабатываемую доку­ментацию на систему, которые касаются принципов организации защиты в самой системе и из которых следуют организационные мероприятия, рекомендуемые в эксплуатационной документации организацией-разработчиком, на период ввода и эксплуатации системы.

Выполнение этих рекомендаций является определенной гаран­тией защиты процессов переработки информации.

К организационным мероприятиям по защите процессов пе­реработки информации в процессе создания системы относятся:

введение на необходимых участках проведения работ с режи­мом секретности;

разработка должностных инструкций по обеспечению режима секретности в соответствии с действующими в стране инструкци­ями и положениями;

при необходимости выделение отдельных помещений с охран­ной сигнализацией и пропускной системой;

разграничение задач по исполнителям и выпуску документа­ции;

присвоение грифа секретности материалам, документации, аппаратуре и хранение их под охраной в отдельных помещениях с учетом и контролем доступа исполнителей;

постоянный контроль за соблюдением исполнителями режима и соответствующих инструкций;

установление ответственных лиц за утечку информации;

другие меры, устанавливаемые главным конструктором при создании конкретной системы.

Организационные мероприятия, закладываемые в инструкцию по эксплуатации на систему и рекомендуемые организации-по­требителю, должны быть предусмотрены на периоды подготовки и эксплуатации системы.

В целях защиты функционирования КС в чрезвычайных ситуа­циях от случайных угроз применяются уже известные средства повышения надежности аппаратуры и программного обеспечения, а для защиты процессов переработки информации — средства повышения ее достоверности. Для предотвращения аварийной ситуации применяют специальные меры.

Методы и средства повышения надежности ВС и достоверно­сти информации в настоящее время достаточно хорошо разрабо­таны. Эти методы и средства косвенным образом помогают суще­ственно сократить влияние случайных воздействий на информацию.

Проблема надежности автоматизированных систем решается тремя путями:

повышением надежности деталей и узлов;

построением надежных систем из менее надежных элементов за счет структурной избыточности (дублирование, утроение эле­ментов, устройств, подсистем и т.п.);

применением функционального контроля (ФК) с диагности­кой отказа, увеличивающего надежность функционирования сис­темы путем сокращения времени восстановления отказавшей ап­паратуры.

Задачами ФК системы являются: своевременное обнаружение сбоев, неисправностей и программных ошибок; исключение их влияния на дальнейший процесс обработки информации и указа­ние места отказавшего элемента, блока программы для последу­ющего быстрого восстановления системы.

Существующие методы ФК вычислительных систем могут быть разделены на программный, аппаратный и комбинированный (со­четание программного с аппаратным).

Программный контроль подразделяют на программно-логический, алгоритмический и тестовый.

Наиболее распространенная форма *программно-логического кон­троля —*это двойной счет со сравнением полученных результатов. *Алгоритмический контроль*заключается в том, что задача, решен­ная по какому-либо алгоритму, проверяется повторно по сокра­щенному алгоритму с достаточной степенью точности.

Программно-логический контроль позволяет надежно обнару­живать сбои, и для его осуществления не требуется дополнитель­ного оборудования. Однако при нем более чем вдвое снижается производительность ВС, не обнаруживаются систематические сбои, нельзя указать место отказа и тем более сбоя, требуется дополнительная емкость памяти для программы вычислений. При алгоритмическом контроле производительность ВС выше, в ос­тальном он обладает теми же недостатками и, кроме того, имеет ограниченное применение, так как не всегда удается найти для основного алгоритма сокращенный, который был бы значитель­но короче основного.

*Тестовый контроль,*как правило, применяют для проверки работоспособности комплекса средств автоматизации при помо­щи испытательных программ. Тестовый контроль, в отличие от программно-логического, проверяет не процесс переработки информации, а пребывание КС или ее части в работоспособном со­стоянии. Кроме того, тестовый контроль не всегда обнаруживает сбои и во время проверки не может решать задачи по рабочей программе.

В настоящее время широкое применение находят методы а п-паратного и комбинированного контроля.

Аппаратный контроль, в отличие от программного, может обес­печивать указание о наличии сбоя или неисправности непосред­ственно в момент его возникновения. Аппаратный контроль в КС делится на контроль по модулю, контроль при дублировании обо­рудования и контроль при троировании оборудования с исполь­зованием мажоритарных элементов.

*Контроль по модулю*основывается на следующих принципах. Из теории чисел известно, что целое положительное число можно представить в виде сравнения:

(читается: *А*сравнимо с остатком *rа*модуля *М).*Это сравнение устанавливает следующее соотношение между числами *А, га*и *М:*

**А ≡ Мl + rа**,

где *А, М, l*и *rа*— целые числа; *А —*любое контролируемое *n*-разряд­ное число; *М—*модуль, или делитель; *l* — частное; *rа —*остаток от деления *А*на модуль *М*(контрольный код числа *А).*

При данном методе контроля каждому контролируемому чле­ну придается еще *т*дополнительных разрядов, в которые записы­вается контрольный код, т. е. остаток *rа.*Если записать все числа в виде сравнения, то после этого их можно будет складывать, пере­множать, а результаты записывать в следующем виде:

Эти выражения означают, что сумма (произведение) чисел сравнима с суммой (произведением) остатков этих чисел по мо­дулю *М.*

Техническая реализация контроля по модулю заключается в разработке специальных схем, которые в литературе получили название «сверток». Эффективность контроля повышается с уве­личением модуля. Однако с увеличением *М*непропорционально возрастает количество дополнительного оборудования и усложняются схемы контроля.

*Контроль при дублировании оборудования*позволяет путем срав­нения выходных сигналов обнаружить отказ аппаратуры. Высокая эффективность такого контроля основывается на том, что вероят­ность одновременного отказа двух одинаковых элементов ничтож­но мала. Недостатком этого метода является не всегда имеющаяся возможность определить, какой из каналов является исправным, и поэтому, чтобы процесс функционирования оставался исправ­ным, приходится одновременно в каждом из каналов применять методы контроля, например контроль по модулю.

*Контроль при троировании оборудования*с элементами «голосо­вания» позволяет наряду с увеличением вероятности безотказной работы увеличить и достоверность функционирования при помо­щи мажоритарных элементов. Данный метод требует, разумеется, увеличения объемов оборудования.

В настоящее время существует много разнообразных методов контроля, имеющих в зависимости от конкретных требований и условий различную степень применяемости. Некоторые из этих методов являются специализированными для определенных ти­пов устройств и систем, другие — приспособлены для проверки определенных видов операций и применяются в различных типах Устройств.

Поскольку результат воздействия на информацию зависит от числа ошибок в данный момент времени, рассмотрим вероятность Появления этих событий. Ввод, хранение и обработка информа­ции в КС осуществляются при помощи кодов чисел и слов по определенному алгоритму. Появление сбоев приводит к тому, что в коде может возникнуть одиночная или групповая ошибка (двухкратная, трехкратная и т.д.). Ошибка может считаться одиночной, если она возникла в одном разряде кода числа или слова.

Способность средств ФК обеспечить своевременно (до начала последующей обработки) обнаружение и блокировку ошибок за­данной кратности определяет уровень достоверности контроля обработки информации. Существенную роль для качества ФК иг­рает плотность распределения его средств обнаружения по всей «площади» контролируемой КС, т. е. полнота ее охвата ФК.

Сравнительная характеристика методов ФК учитывает следу­ющие факторы:

надежность обнаружения;

возможность исправления ошибок после сбоев без вмешатель­ства оператора;

время, затрачиваемое на устранение случайных ошибок;

количество дополнительного оборудования;

способы применения (параллельно или с прерыванием обработки информации);

влияние контроля на быстродействие вычислительной систе­мы или ее производительность;

указание места неисправности с необходимой точностью.

Одним из основных условий эффективного функционирова­ния автоматизированной системы является обеспечение требуемого уровня достоверности информации. Под *достоверностью инфор­мации*понимают некоторую функцию вероятности ошибки, т.е. события, заключающегося в том, что реальная информация в си­стеме о некотором параметре не совпадает в пределах заданной точности с истинным значением.

Необходимая достоверность достигается использованием раз­личных методов, реализация которых требует введения в системы обработки данных информационной, временной или структур­ной избыточности. Достоверность при обработке данных достига­ется путем контроля и выявления ошибок в исходных и выводи­мых данных, их локализации и исправления. Условие повыше­ния достоверности — снижение доли ошибок до допустимого уровня. В конкретных КС требуемая достоверность устанавливает­ся с учетом нежелательных последствий, к которым может при­вести возникшая ошибка, и тех затрат, которые необходимы для ее предотвращения.

Методы контроля при обработке информации в КС классифи­цируют по различным параметрам: по числу операций, охватыва­емых контролем, — единичный (одна операция), групповой (груп­па последовательных операций), комплексный (контролируется, например, процесс сбора данных); по частоте контроля — непрерывный, циклический, периодический, разовый, выборочный, по отклонениям; по времени контроля — до выполнения основ­ных операций, одновременно с ними, в промежутках между ос­новными операциями, после них; по виду оборудования контроля — встроенный, контроль с помощью дополнительных техни­ческих средств, безаппаратный; по уровню автоматизации — ручной, автоматизированный, автоматический.

Различают системные, программные и аппаратные методы кон­троля достоверности.

*Системные методы*включают в себя: оптимизацию структуры обработки; поддержание характеристик оборудования в заданных пределах; повышение культуры обработки; обучение и стимули­рование обслуживающего персонала; создание оптимального числа копий и (или) предыстории программ исходных и текущих данных; определение оптимальной величины пакетов данных и ско­рости первичной обработки, процедур доступа к массивам дан­ных и др.

*Программные методы*повышения достоверности информации состоят в том, что при составлении процедур обработки данных в них предусматривают дополнительные операции, имеющие мате­матическую или логическую связь с алгоритмом обработки дан­ных. Сравнение результатов этих дополнительных операций с результатами обработки данных позволяет установить с определен­ной вероятностью наличие или отсутствие ошибок. На основании этого сравнения, как правило, появляется возможность испра­вить обнаруженную ошибку.

*Аппаратные методы*контроля и обнаружения ошибок могут выполнять практически те же функции, что и программные. Ап­паратными методами обнаруживают ошибки быстрее и ближе к месту их возникновения, а также ошибки, недоступные для про­граммных методов.

Все перечисленные методы контроля обработки данных бази­руются на использовании определенной избыточности. При этом различают методы контроля со структурной, временной и инфор­мационной избыточностью.

*Структурная избыточность*требует введения в состав КС до­полнительных элементов (резервирование информационных мас­сивов и программных модулей, реализация одних и тех же функ­ций различными программами, схемный контроль в технических средствах КС и т.д.).

*Временная избыточность*связана с возможностью неоднократ­ного повторения определенного контролируемого этапа обработ­ки данных.

Обычно этап обработки данных повторяют неоднократно и результаты обработки сравнивают между собой. В случае обнару­жения ошибки производят исправления и повторную обработку данных.

*Информационная избыточность*может быть естественной и ис­кусственной. Естественная информационная избыточность отра­жает объективно существующие связи между элементами обра­ботки, наличие которых позволяет судить о достоверности ин­формации. Искусственная информационная избыточность характеризуется введением дополнительных информационных разря­дов в цифровом представлении обрабатываемых данных и допол­нительных операций в процедуре их обработки, имеющих мате­матическую или логическую связь с алгоритмом обработки дан­ных.

На основании анализа результатов дополнительных операций и процедур обработки данных, а также дополнительных инфор­мационных разрядов выявляется наличие или отсутствие ошибок определенного типа, а также возможности их исправления.

В работах по повышению достоверности информации рассмат­риваются совместно помехоустойчивость и надежность систем пе­редачи и обработки информации с позиций качества таких систем.

В зависимости от характера информации, особенностей алго­ритмов системы, а также от задач, стоящих перед ее адресатами, можно определить следующие зависимости содержания инфор­мации от ошибок при ее передаче:

смысловой объем информации в сообщении уменьшается про­порционально числу искаженных разрядов в кодовой комбина­ции данного сообщения;

искажение одного или нескольких разрядов приводит почти к полной потере остальной части информации, содержащейся в смысловом отрезке сообщения.

Проанализируем способность средств ФК и повышения досто­верности информации к защите от случайных разрушений, моди­фикации и утечки информации.

Отказы, сбои в аппаратуре и ошибки в программном обеспе­чении могут привести к нарушению функционирования ВС, раз­рушению и изменению информации на ложную.

Если рассматривать искажение информации (без ее модифика­ции) как разрушение информации, то условием его возникнове­ния может считаться однократная ошибка, несмотря на то, что пропадание одной буквы не всегда ведет к потере информации.

Для возникновения случайной утечки информации при ее обра­ботке в ВС необходимо, чтобы в результате случайных воздействий был перепутан адрес получателя или в правильный адрес была выдана другая информация, для него не предназначенная. В первом случае, например, заменилась одна из букв на другую (модифи­кация), во втором — адресация ячеек памяти ОЗУ, из которого считывалась информация до ее передачи получателю (тоже моди­фикация).

Таким образом, можно полагать, что в нашем случае утечка информации — это частный случай ее модификации.

Вероятность указанных событий зависит от многих факторов, но, анализируя приведенные относительные условия их наступле­ния, можно дать им сравнительную оценку. Вероятность разруше­ния информации от случайных воздействий больше, чем ее моди­фикации, а вероятность модификации информации больше веро­ятности ее утечки. Эта оценка необходима для выработки подхода к функциональному контролю с позиций защиты информации, ко­торый выражается в предъявлении к средствам ФК дополнитель­ных требований, выполнение которых может потребовать допол­нительных средств. Дополнительные требования заключаются в ре­ализации уменьшения вероятности модификации и утечки инфор­мации существующими средствами повышения надежности и дос­товерности информации. Для выполнения этой задачи в настоящее время применяют специальные системотехнические решения:

изоляция областей доступа к информации;

специальная организация работы с данными, хранящимися в памяти вычислительной системы.

Изоляция областей доступа к информации ВС осуществляется также для поддержки разграничения санкционированного доступа.

Информация, содержащаяся в КС, может быть поделена меж­ду пользователями, что требует размещения ее в непересекающихся областях, отведенных для ее хранения. В каждой из этих областей хранится совокупность информационных объектов, подлежащих в равной степени защите. В процессе эксплуатации системы необ­ходимо обеспечить надежное разграничение доступа к информа­ции. Для этой цели помимо организации доступа с помощью си­стемы паролей в систему при проектировании закладываются до­полнительные меры по изоляции областей доступа, нарушение которых по причине отказов и программных ошибок не приводи­ло бы к НСД информации.

При наличии в системе общего поля памяти, которое необхо­димо для решения поставленных задач, схемы защиты допускают обмен информацией между пользователями. Тогда применяются списковые и мандатные схемы защиты. В списковых схемах систе­ма охраны снабжается списком всех лиц, имеющих право доступа к информации (для получения права доступа достаточно предъя­вить свой идентификатор). *Мандатные схемы*— схемы, в которых система охраны реализует только один вид мандата, а пользова­тель должен иметь набор мандатов для доступа к каждому из не­обходимых ему объектов.

Наибольшую опасность составляют многократные ошибки, при­водящие к модификации самой информации и команд, осуществ­ляющих ее обработку. При этом уровень безопасности информации находится в прямой зависимости от числа одновременно возникаю­щих ошибок. Способность средств функционального контроля к их обнаружению определяет уровень безопасности информации. По­скольку вероятность появления четырехкратной ошибки относительно мала, то вероятность обнаружения двух- и трехкратных ошибок и будет мерой безопасности информации от отказов аппаратуры.

Защита процессов переработки информации в чрезвычайных ситуациях, особенно в аварийных, заключается в создании средств предупреждения, контроля и организационных мер по исключе­нию НСД на комплексе средств автоматизации в условиях отка­зов его функционирования, отказов системы защиты информа­ции, систем жизнеобеспечения людей на объекте размещения и при возникновении стихийных бедствий.

Отказ функционирования КС может повлечь за собой отказ системы защиты процессов переработки информации, может от­крыться доступ к ее носителям (магнитным лентам, барабанам, дискам и т.д.), что, в свою очередь, может привести к преднаме­ренному разрушению, хищению или подмене носителя. Несанкционированный доступ к внутреннему монтажу аппаратуры мо­жет привести к подключению посторонней аппаратуры, разруше­нию или изменению принципиальной электрической схемы.

Отказ системы жизнеобеспечения может привести к выводу из строя обслуживающего и контролирующего персонала. Стихийные бедствия (пожар, наводнение, землетрясение, удары молнии и т.д.) могут также привести к указанным последствиям. Аварийная ситу­ация может быть создана преднамеренно нарушителем. В последнем случае применяются организационные мероприятия.

Наиболее частой и типичной естественной угрозой является пожар. Он может возникнуть по вине обслуживающего персонала, при отказе аппаратуры, а также в результате стихийного бедствия.

**Вопросы к экзамену по ПМ 11 «Разработка, администрирование и защита баз данных»**

1. Приведите классификацию баз данных. Определения, основные функции, виды.
2. Основы реляционной алгебры. Определения высказываний, запись, примеры.
3. Иерархическая модель данных. Основные понятия, графическое изображение, примеры.
4. Сетевая модель данных. Основные понятия, графическое изображение, примеры.
5. Реляционная модель данных. Основные понятия, графическое изображение, примеры.
6. Первая, вторая, третья нормальные формы реляционной модели данных. Определение, требования, примеры.
7. НФБК, четвертая нормальная форма реляционной модели данных. Определение, требования, примеры.
8. Модель сервера приложений. Архитектура, компоненты, преимущества.
9. Модель клиент. Архитектура, компоненты, преимущества.
10. Создание структуры таблицы. Типы данных таблиц базы данных. Ключевое поле таблиц баз данных.
11. Заполнение таблиц данными. Установление связей между таблицами. Назначение, главная и подчиненная таблица, последовательность действий.
12. Запрос на выборку. Назначение, создание, примеры. Запрос с параметром. Назначение, создание, примеры. Создание запроса на выборку с логическими операциями в условиях отбора.
13. Запрос на создание таблицы. Запрос на удаление. Назначение, создание, примеры. Запрос на обновление. Запрос на добавление. Назначение, создание, примеры.
14. *Опишите основные принципы построения концептуальной, логической и физической модели данных.*
15. *Расскажите про структуры данных СУБД, общий подход к организации представлений, таблиц, индексов и кластеров*
16. *Расскажите про основные принципы структуризации и нормализации базы данных.*
17. *Опишите методы описания схем баз данных в современных СУБД. Расскажите про структуры данных СУБД.*
18. *Расскажите про методы организации целостности данных в БД.*
19. *Опишите современные инструментальные средства проектирования схемы базы данных.*
20. *Введение в SQL и его инструментарий. Подготовка систем для установки SQL-сервера. Расскажите про установку и настройку SQL-сервера, автоматизация управления SQL*
21. *Расскажите про выполнение мониторинга SQLServer с использование оповещений и предупреждений.*
22. *Расскажите про поиск и решение типичных ошибок, связанных с администрированием.*
23. *Расскажите про способы контроля доступа к данным и управления привилегиями.*
24. *Опишите алгоритм проведения процедуры резервного копирования.*
25. *Опишите модели восстановления SQL-сервера.*
26. *Расскажите про резервное копирование баз данных, восстановление баз данных*
27. *Расскажите про аутентификацию и авторизацию пользователей, назначение серверных ролей и ролей баз данных, авторизацию пользователей при получении доступа к ресурсам.*