МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Кузбасский государственный технический университет имени Т. Ф. Горбачева»

Кафедра информационной безопасности

Составители
Е. В. Прокопенко
И. В. Чичерин

БЕЗОПАСНОСТЬ ОПЕРАЦИОННЫХ СИСТЕМ

Методические материалы

Рекомендованы учебно-методической комиссией специальности
10.05.03 Информационная безопасность автоматизированных систем
в качестве электронного издания для использования
в образовательном процессе

Кемерово 2018

Рецензенты
Стенин Д. В.– кандидат технических наук, доцент, директор ИИТМА
Сыркин И. С. – кандидат технических наук, доцент кафедры информационных и автоматизированных производственных систем

Прокопенко Евгения Викторовна
Чичерин Иван Владимирович

Безопасность операционных систем: методические материалы [Электронный ресурс]: для обучающийся специальности 10.05.03 Информационная безопасность автоматизированных систем очной формы обучения / сост. Е. В. Прокопенко, И. В. Чичерин; КузГТУ. – Кемерово, 2018. – Систем. требования: Pentium IV; ОЗУ 8 Мб; Windows XP; мышь. – Загл. с экрана.

1. Архитектура современных ОС
Введение в операционные системы. Процессы. Алгоритмы и механизмы синхронизации. Тупики. Управление памятью. Файловая система. Система ввода-вывода. Механизмы синхронизации процессов и потоков.

Появление общего программного обеспечения в ЭВМ относят к 1953 г., когда в СССР появилась одна из первых теоретических работ по автоматизации программирования для цифровых ЭВМ (А.П. Ершов), а в Массачусетсом технологическом институте (США) была создана экспериментальная "операционная система", применявшаяся в учебных целях. Затем появились специализированные операционные системы (ОС) для обслуживания оборонных вычислительных систем реального времени. Однако эти разработки имели экспериментальный, исследовательский характер и широкого распространения в то время не получили. Тем не менее, потребности практического использования ЭВМ в различных предметных областях, необходимость более эффективного использования ЭВМ. Повышение производительности труда разработчиков программного обеспечения, а также стремление расширить рынок сбыта ЭВМ вызвали стремительный прогресс в создании теории и инструментальных средств общего программного обеспечения вычислительных систем.

Построение вычислительных машин основано на трех принципах:

1. Принцип цифрового представления данных (чисел, команд, обозначение операции, букв, слов и т.д.). Единицами данных в ЭВМ являются бит, байт, слово и т.п.

2. Принцип адресности данных – все данные и любые объекты программы хранятся в ячейках памяти, имеющих адрес.

3. Принцип программного управления (Ч. Беббндж, 1834 г.) – управление вычислительным процессом осуществляется с помощью программы, находящейся в памяти ЭВМ.

Все универсальные вычислительные машины, в том числе и персональные компьютеры, имеют структуру, показанную на рис. 1.

Впервые такую структуру вычислительных машин предложил Джон фон Нейман в 1945 г. поэтому ЭВМ с такой структурой называют машинами фон Неймана.

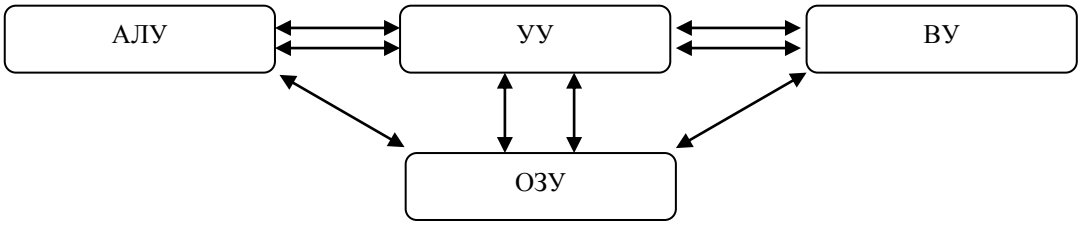

Рис. 1. Общая структура универсальной ЭВМ: АЛУ – арифметико-логическое устройство; УУ – устройство управления: ВУ – внешние устройства; ОЗУ – оперативное запоминающее устройство

Конкретная аппаратурная реализация схемы изменялась от поколения к поколению ЭВМ. Например, в современных компьютерах АЛУ и УУ объединены в единое устройство – *центральный процессор.* Кроме того, в ЭВМ ввели *систему прерываний.* Появились многопроцессорные ЭВМ, позволяющие осуществлять параллельную обработку данных в компьютере. Тем не менее, функциональная структура существующих компьютеров в основном соответствует структуре машины фон Неймана.

Архитектура вычислительной системы – общая логическая организация цифровой вычислительной системы, определяющая процесс обработки данных в конкретной

вычислительной системе и включающая методы кодирования данных, состав, назначение, принципы взаимодействия технических средств и программного обеспечения.

Большинство из выпускаемых сейчас компьютеров выполнено в соответствии с принципом открытой архитектуры, впервые примененном в персональной ЭВМ IBM PC (фирма IBM, 1981 г.).

Классификация программных средств

Программное обеспечение вычислительных систем принято делить на следующие виды:

1) общее (системное) программное обеспечение (ОПО):

2) специальное программное обеспечение (СПО). Введем ряд определений.

В состав общего программного обеспечения вычислительных систем входят:

• программные средства управления обработкой данных, включая операционные системы;

• обслуживающие (сервисные) программы (утилиты):

• инструментальные программные средства.

Специальное ПО делят на следующие виды:

• прикладные программы (приложение) общего назначения;

• прикладные программы пользователя.

Прикладные программы общего назначения можно разделить на следующие группы:

- программа офисного назначения;
- программа экономического назначения;
- издательские системы:
- компьютерная графика, видео, анимация и звук;
- системы управления базами данных;
- прочие прикладные программы общего назначения.

Можно видеть, что современные компьютеры и их программное обеспечение глубоко внедрилось практически во все сферы человеческой деятельности: науку, производство, экономику, право и т.д.

Функционирование прикладных программ любого назначения происходит под управлением и при участии программ, относящихся к категории системного программного обеспечения.

Место и функции системного программного обеспечения

Системное ПО играет роль "прослойки" между пользователем и техническими средствами вычислительной системы. На различных этапах работы с компьютером в качестве такой "прослойки" выступают разные программы и пакеты программ системного ПО, выполняя при этом отличающиеся назначением функции.

Основой системного ПО является операционная система.

Операционная система (ОС) цифровой вычислительной системы – система программ, предназначенная для обеспечения определенного уровня эффективности цифровой вычислительной системы за счет автоматизированного управления ее работой и предоставляемого пользователям набора услуг.

Основными функциями ОС являются:

1) автоматическое выполнение действий по запуску задач в обработку и их завершению;

2) диспетчеризация (планирование обработки задач);

3) распределение памяти между различными задачами;

4) управление ходом выполнения задач в вычислительной системе;

5) распределение задачам необходимых ресурсов ВС;
6) синхронизация выполнения задач;
7) поддержка выполнения операции ввода/вывода данных;
8) ведение учета работы системы (при необходимости).

Выполнение своих функции ОС осуществляется с помощью соответствующих программных комплексов управления, которые носят название супервизорных программ (супервизоров или менеджеров).

Супервизорная программа – машинная программа, являющаяся обычно частью операционной системы, которая управляет выполнением других машинных программ и регулирует поток работ в системе управления данными.

Супервизор – часть управляющей программы, координирующая распределение ресурсов вычислительной системы.

В целом современные операционные системы представляют собой иерархическую структуру (рис. 2).

В основе иерархии находится аппаратура вычислительной машины, называемая иногда "чистой машиной" или "голым железом". На следующем уровне иерархии (иногда на следующих нескольких уровнях) находятся некоторые функции ядра операционной системы. В совокупности с этими функциями ядра (называемыми еще "примитивами") компьютер становится *расширенной машиной,* т.е. машиной, которая представляет для операционной системы и пользователей не только свой машинный язык, но и ряд дополнительных возможностей.

Выше над ядром расположены программы ОС для обеспечения выполнения задач пользователя (управления внешними устройствами, обслуживание операций ввода/вывода и т.п.). На вершине иерархии находятся программы пользователя. В подобных иерархических системах принято, как правило, следующее ограничение: допускается обращение только сверху вниз в иерархии, т.е. средства каждого уровня могут обращаться только к тем функциям, которые находятся на ближайшем нижележащем уровне.

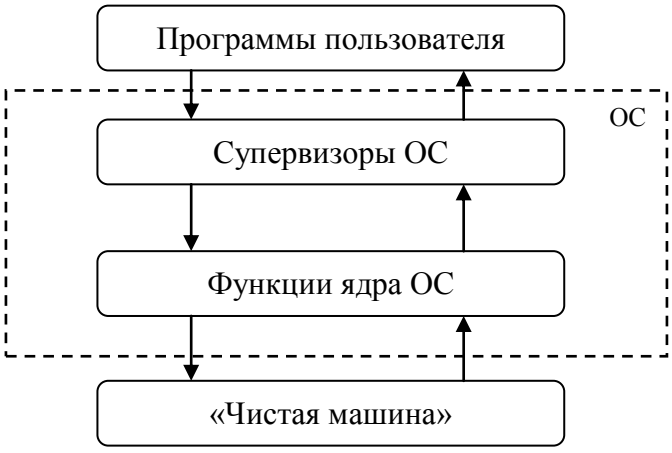

Рис. 2. Структура операционной системы

Обслуживающие (сервисные) программы (утилиты) предназначены для выполнения различных вспомогательных функций и разделяются на следующие типы: программы-упаковщики (архиваторы); антивирусные программы; программы резервирования; программы диагностики компьютера; программы оптимизации дисков; программы динамического сжатия дисков.

Инструментальные программные средства, называемые также средствами разработки приложений и системами программирования, являются орудием автоматиза-

ции разработок программного обеспечения ЭВМ, обеспечивающим повышение производительности труда разработчиков и надежности ПО.

К инструментальным программным средствам относятся:

- компиляторы и интерпретаторы;
- автономные отладчики (дебагеры. от англ. Debug "удаление насекомых");
- интегрированные оболочки;
- средства создания приложений типа клиент-сервер и т.п.

Существующие инструментальные программные средства обеспечивают разработчиков ПО всем необходимым набором функций для создания мощного программного обеспечения решения прикладных задач любой мощности для практически всех предметных областей.

Принципы работы вычислительной системы

Всякая вычислительная система создается для решения некоторого множества вычислительных или информационных задач, которые в совокупности называются задачами обработки данных. Для успешного решения любой задачи в вычислительной системе необходимо иметь:

1) программу, реализующую алгоритм решения задачи;

2) аппаратные средства ВС для ввода программы, выполнения программы, получения дополнительной информации и вывода результатов:

3) дополнительные программные средства, необходимые для решения прикладной задачи (стандартные программы).

Существует три вида систем обработки данных (СОД), отличающихся друг от друга требованиями к скорости получения результатов решения задач:

1) системы реального времени (СРВ), в которых требования к скорости обработки информации очень высокие из-за необходимости решения задач в темпе реального времени (примером являются системы навигации и управления летательными аппаратами);

2) системы оперативной обработки (СОО), в которых планирование заданий на обработку данных осуществляется, исходя из требования минимальности времени выполнения каждого полученного задания. Примером такого вида систем является система обработки данных для персонала боевых расчетов пунктов управления;

3) системы пакетной обработки (СПО), в которых основным требованием является минимизация простоя оборудования при решении поставленных задач.

Запуск прикладной программы в работу, предоставление ей необходимых аппаратных мощностей и программных средств осуществляется операционной системой.

Режимы работы операционных систем

Режимы обработки данных

Порядок представления прикладной программе перечисленных средств определяется режимов обработки данных, реализованных в операционной системе ЭВМ. Различают однопрограммные и мультипрограммные режимы обработки данных и, соответственно, работы ОС.

К однопрограммным режимам относятся:

– режим непосредственного доступа (РНД):

– пакетный однопрограммный режим (ПШ). Мультипрограммными режимами обработки данных являются:

– пакетный мультипрограммный режим (ПМП);

– режим разделения времени (РРВ).

Однопрограммные режимы обработки данных

Режим непосредственного доступа широко применялся в ЭВМ первого поколения и используется при работе с современными персональными компьютерами. РНД характерен тем, что ЭВМ предоставляется только одному пользователю, который осуществляет взаимодействие с машиной посредством пульта управления (сейчас – клавиатура, мыши и дисплеи). Время решения каждой задачи в РНД складывается из времени 7"$_{вв}$ ввода программы и данных в ЭВМ, времени T_p работы процессора над решением задачи, времени $T_{ву}$ обмена данными с внешними устройствами (включая вывод результатов и обработки), времени $T_{оп}$ обслуживания ЭВМ и задачи оператором ЭВМ при ее подготовке к запуску и по окончании решения задачи:

$$T_{НД} = T_{ВВ} + T_{ЦП} + T_{ВУ} + T_{ОП}.$$

Коэффициент загрузки процессора при одной задаче составляет

$$\eta_{НД} = T_{ЦП} / T_{НД}.$$

Полное время решения N задач и коэффициент загрузки:

$$T_{НД}(N) = \sum_{i=1}^{N} [T_{ВВ}(i) + T_{ЦП}(i) + T_{ВУ}(i) + T_{ОП}(i)];$$

$$\eta_{НД}(N) = \left[\sum_{i=1}^{N} T_{ЦП}(i) \right] \Big/ T_{НД},$$

где i – номер задачи.

В РНД наличие ОС не обязательно. Недостатками РНД являются:

1) аппаратура и программы ЭВМ используются не эффективно:

2) велики затраты времени программиста на управление машиной;

3) предъявляются высокие требования к подготовке пользователя как оператора вычислительной машины.

Пакетный однопрограммный режим применяется в ВС, начиная с ЭВМ второго поколения. Несколько заданий для решения задач обработки собираются в один пакет, называемый пакет заданий (ПЗ). Пакет гаданий оператор ЭВМ вводит в ЭВМ, где ПЗ сначала записывается во внешнюю память (магнитные диски, магнитные барабаны и т.п.). Затем операционная система машины последовательно считывает задания, входящие в ПЗ. и осуществляет выполнение необходимых в соответствиями с заданиями действий для решения задач пользователей. После завершения очередного задания происходит обращение к ОС, которая активирует начато выполнения следующего. После завершения последнего задания пакета оператор ЭВМ загружает в машину новый пакет заданий.

Режим П1П обладает следующими положительными чертами:

1) более высокая пропускная способность;

2) отсутствие специальных требований к аппаратуре ЭВМ;

3) возможна его реализация на любой ЭВМ.

К недостаткам режима П1П относятся:

1) необходимо наличие операционной системы;

2) пользователь физически отделен от ЭВМ и решаемой им задачи;

3) увеличивается реакция пользователя на полученные результаты решения;

4) последовательный порядок выполнения заданий пакетов не позволяет увеличить загрузку оборудования вычислительной системы.

Многопрограммные режимы обработки данных

Пакетный мультипрограммный режим широко применяется в ЭВМ третьего и последующих поколении. ПМП является режимом классического мультипрограммирования, при котором в вычислительной системе находятся в обработке сразу несколько заданий. На входе в систему формируется набор пакетов гаданий, которые оператор ЭВМ загружает в систему. После окончания ввода первого ПЗ операционная система начинает его обработку, не дожидаясь ввода второго и последующих ПЗ. Задания, принадлежащие одному пакету*, выполняются последовательно (т.е. в режиме П1П). Задания, принадлежащие разным пакетам, выполняются параллельно. Первым начинает выполняться первое задание первого пакета. По мере освобождения ресурсов ОС активизирует выполнение заданий из других пакетов в порядке их следования внутри ПЗ.

Пакетный мультипрограммный режим обеспечивает наивысшую пропускную способность вычислительной системы, что достигается при наличии в ЭВМ следующих аппаратных средств:

1) автономно управляемые внешние устройства:

2) развитая система прерывания программ;

3) средства защиты памяти от взаимного влияния программ.

Основным недостатком режима ПМП является практически полное устранение пользователя из системы и, как следствие, отсутствие связи пользователя со своей задачей.

Режим разделения бремени существенно отличается от классического мультипрограммирования, реализованного в ПМП, и является в настоящее время основным режимом функционирования операционных систем. Главное в режиме разделения времени – это предоставление каждой задаче (или пользователю, работающему в диалоге с машиной) ресурсов ЭВМ на некоторый ограниченный интервал времени (квант). По истечении кванта времени данная программа свертывается операционной системой, развертывается следующая по очереди программа (или подключается следующий терминал пользователя), которой предоставляются ресурсы ЭВМ, и т.д.

Режимы и дисциплины обслуживания

Порядок обслуживания заданий (заявок на работу) в операционных системах с мультипрограммированием, т.е. реализующих режимы ПМП или РРВ, определяется принятыми в них режимами обслуживания и дисциплинами обслуживания,

Режимы обслуживания

Режимом обслуживания называется правило отбора заявок на обслуживание. Режимы обслуживания делятся на три вида:

1) режим одиночного отбора заявок:

2) режим группового отбора, когда на обслуживание отбирается вся очередь заявок определенного типа:

3) смешанный режим отбора, когда для одних классов заявок производится одиночный отбор, а для других групповой.

Дисциплины обслуживания

Дисциплиной обслуживания называется правило отбора заявок на обслуживание при заданном режиме обслуживания. Для каждого из режимов обслуживания может быть применен один из следующих видов дисциплин обслуживания:

- бесприоритетное обслуживание:
- обслуживание с приоритетом;
- обслуживание по расписанию.

Разновидности дисциплины бесприоритетного обслуживания:

1) ОПП – обслуживание в порядке поступления ("первый пришел – первый обслужен". FIFO);

2) ООП – обслуживание в обратном порядке ("первый пришел – последний обслужен", LIFO);

3) ОСП – обслуживание в случайном порядке.

При бесприоритетном обслуживании считается, что все заявки имеют равное право на обслуживание. Если требуется, чтобы заявки некоторого типа имели преимущества перед другими на их обслуживание операционной системой, то применяется дисциплина обслуживания с приоритетами:

1) ДОП – дисциплина обслуживания с относительными приоритетами, когда приоритет заявки влияет только на ее место в очереди заявок на обслуживание;

2) ДАП – дисциплина с абсолютными приоритетами, когда высоко приоритетная заявка получает преимущества не только перед заявками, стоящими в очереди, но и перед заявкой, получающей обслуживание;

3) ДСП – дисциплина со смешанными приоритетами, при которой к одним группам заявок применяются относительные приоритеты, а к другим – абсолютные:

4) ДДП – дисциплина обслуживания с динамическими приоритетами, когда значение приоритетов заявок может изменяться (расти) по мере их нахождения в очереди, обеспечивая тем самым первоочередное обслуживание заявок, долго находящихся в системе.

Дисциплина обслуживания по расписанию обеспечивает заданный пользователем порядок обработки задании независимо от очередности их поступления в систему. Она применяется в тех случаях, когда результаты решения одной задачи являются входными данными для другой.

Классификация операционных систем

Операционные системы могут различаться особенностями реализации внутренних алгоритмов управления основными ресурсами компьютера (процессорами, памятью, устройствами), особенностями использованных методов проектирования, типами аппаратных платформ, областями использования и многими другими свойствами.

Особенности алгоритмов управления ресурсами

От эффективности алгоритмов управления локальными ресурсами компьютера во многом зависит эффективность всей сетевой ОС в целом. Поэтому, характеризуя сетевую ОС, часто приводят важнейшие особенности реализации функций ОС по управлению процессорами, памятью, внешними устройствами автономного компьютера. Так, например, в зависимости от особенностей использованного алгоритма управления процессором, операционные системы делят на многозадачные и однозадачные, многопользовательские и однопользовательские, на системы, поддерживающие многопроцессорную обработку и не поддерживающие ее, на многопроцессорные и однопроцессорные системы.

Поддержка многозадачности

По числу одновременно выполняемых задач операционные системы могут быть разделены на два класса:

1) однозадачные (например, MS DOS. MSX);

2) многозадачные (ОС ЕС. OS/2, Unix, Windows 95).

Однозадачные ОС в основном выполняют функцию предоставления пользователю виртуальной машины, делая более простым и удобным процесс взаимодействия пользователя с компьютером. Однозадачные ОС включают средства управления периферийными устройствами, средства управления файлами, средства общения с пользователем.

Многозадачные ОС, кроме вышеперечисленных функций, управляют разделением совместно используемых ресурсов, таких как процессор, оперативная память, файлы и внешние устройства.

Поддержка многопользовательского режима. По числу одновременно работающих пользователей ОС делятся на:

1) однопользовательские (MS DOS. Windows 3.x, ранние версии OS/2);

2) многопользовательские (Unix, Windows NT).

Главным отличием многопользовательских систем от однопользовательских является наличие средств зашиты информации каждого пользователя от несанкционированного доступа других пользователей. Следует заметить, что не всякая многозадачная система является многопользовательской, и не всякая однопользовательская ОС является однозадачной.

Вытесняющая и невытесняющая многозадачность. Важнейшим разделяемым ресурсом является процессорное время. Способ распределения процессорного времени между несколькими одновременно существующими в системе процессами (или нитями) во многом определяет специфику ОС. Среди множества существующих вариантов реализации многозадачности можно выделить две группы алгоритмов:

1) невытесняющая многозадачность (NetWare. Windows 3.x);

2) вытесняющая многозадачность (Windows NT. OS/2, Unix).

Основным различием между вытесняющим и невытесняющим вариантами много задачи ости является степень централизации механизма планирования процессов. В первом случае механизм планирования процессов целиком сосредоточен в операционной системе, а во втором – распределен между системой и прикладными программами. При невытесняющей многозадачности активный процесс выполняется ло тех пор, пока он сам, по собственной инициативе, не отлает управление операционной системе для того, чтобы та выбрала из очереди другой готовый к выполнению процесс. При вытесняющей многозадачности решение о переключении процессора с одного процесса на другой принимается операционной системой, а не самим активным процессом.

Поддержка многонитевости

Важным свойством операционных систем является возможность распараллеливания вычислений в рамках одной задачи. Многонитевая ОС разделяет процессорное время не между задачами, а между их отдельными ветвями (нитями).

Многопроцессорная обработка. Другим важным свойством ОС является отсутствие или наличие в ней средств поддержки многопроцессорной обработки – мультипроцессорование. Мультипроцессирование приводит к усложнению всех алгоритмов управления ресурсами.

В наши дни становится общепринятым введение в ОС функций поддержки многопроцессорной обработки данных. Такие функции имеются в операционных системах Solaris 2.x фирмы Sim. Open Server 3.x компании Santa Crus Operations, OS/2 фирмы ЮМ, Windows NT фирмы Microsoft и NetWare 4.1 фирмы Novell.

Многопроцессорные ОС могут классифицироваться по способу организации вычислительного процесса в системе с многопроцессорной архитектурой: асимметричные ОС и симметричные ОС. Асимметричная ОС целиком выполняется только на одном из процессоров системы, распределяя прикладные задачи по остальным процессорам. Симметричная ОС полностью децентрализована и использует весь пул процессоров, разделяя их между системными и прикладными задачами.

Выше были рассмотрены характеристики ОС, связанные с управлением только одним типом ресурсов – процессором. Важное влияние на облик операционной систе-

мы, в целом, на возможности ее использования в той или иной области оказывают особенности и других подсистем управления локальными ресурсами – подсистем управления памятью, файлами, устройствами ввода-вывода.

Специфика ОС проявляется и в том, каким образом она реализует сетевые функции: распознавание и перенаправление в сеть запросов к удаленным ресурсам, передача сообщений по сети, выполнение удаленных запросов. При реализации сетевых функций возникает комплекс задач, связанных с распределенным характером хранения и обработки данных в сети: ведение справочной информации обо всех доступных в сети ресурсах и серверах, адресация взаимодействующих процессов, обеспечение прозрачности доступа, тиражирование данных, согласование копий, поддержка безопасности данных.

Особенности аппаратных платформ

На свойства операционной системы непосредственное влияние оказывают аппаратные средства, на которые она ориентирована. По типу аппаратуры различают операционные системы персональных компьютеров, мини-компьютеров, мейн-фреймов, кластеров и сетей ЭВМ. Среди перечисленных типов компьютеров могут встречаться как однопроцессорные варианты, так и многопроцессорные. В любом случае специфика аппаратных средств, как правило, отражается на специфике операционных систем.

Очевидно, что ОС большой машины является более сложной и функциональной, чем ОС персонального компьютера. Так в ОС больших машин функции по планированию потока выполняемых задач, очевидно, реализуются путем использования сложных приоритетных дисциплин и требуют большей вычислительной мощности, чем в ОС персональных компьютеров. Аналогично обстоит дело и с другими функциями.

Сетевая ОС имеет в своем составе средства передачи сообщений между компьютерами по линиям связи, которые совершенно не нужны в автономной ОС. На основе этих сообщений сетевая ОС поддерживает разделение ресурсов компьютера между удаленными пользователями, подключенными к сети. Для поддержания функций передачи сообщений сетевые ОС содержат специальные программные компоненты, реализующие популярные коммуникационные протоколы, такие как IP, IPX. Ethernet и другие.

Многопроцессорные системы требуют от операционной системы особой организации, с помощью которой сама операционная система, а также поддерживаемые ею приложения могли бы выполняться параллельно отдельными процессорами системы. Параллельная работа отдельных частей ОС создает дополнительные проблемы для разработчиков ОС, так как в этом случае гораздо сложнее обеспечить согласованный доступ отдельных процессов к общим системным таблицам, исключить эффект гонок и прочие нежелательные последствия асинхронного выполнения работ.

Другие требования предъявляются к операционным системам кластеров. Кластер – слабо связанная совокупность нескольких вычислительных систем, работающих совместно для выполнения общих приложений, и представляющихся пользователю единой системой. Наряду со специальной аппаратурой для функционирования кластерных систем необходима и программная поддержка со стороны операционной системы, которая сводится в основном к синхронизации доступа к разделяемым ресурсам, обнаружению отказов и динамической реконфигурации системы. Одной из первых разработок в области кластерных технологий были решения компании Digital Equipment на базе компьютеров VAX. Недавно этой компанией заключено соглашение с корпорацией Microsoft о разработке кластерной технологии, использующей Windows NT. Несколько компаний предлагают кластеры на основе Unix-машин.

Наряду с ОС, ориентированными на совершенно определенный тип аппаратной платформы, существуют операционные системы, специально разработанные таким образом, чтобы они могли быть легко перенесены с компьютера одного типа на компьютер другого типа, так называемые мобильные ОС. Наиболее ярким примером такой ОС является популярная система Unix. В этих системах аппаратно-зависимые места тщательно локализованы, так что при переносе системы на новую платформу переписываются только они. Средством, облегчающим перенос остальной части ОС, является написание ее на машино-независимом языке, например на С, который и был разработан для программирования операционных систем.

Особенности областей использования

Многозадачные ОС подразделяются на три типа в соответствии с использованными при их разработке критериями эффективности:

1) системы пакетной обработки (например, ОС ЕС);
2) системы разделения времени (Unix. VMS);
3) системы реального времени (QNX. RT/11).

Системы пакетной обработки предназначались для решения задач в основном вычислительного характера, не требующих быстрого получения результатов. Главной целью и критерием эффективности систем пакетной обработки является максимальная пропускная способность, т.е. решение максимального числа задач в единицу времени. Для достижения этой пели в системах пакетной обработки используются следующая схема функционирования: в начале работы формируется пакет заданий, каждое задание содержит требование к системным ресурсам; из этого пакета заданий формируется мультипрограммная смесь, т.е. множество одновременно выполняемых задач. Для одновременного выполнения выбираются задачи, предъявляющие отличающиеся требования к ресурсам, так, чтобы обеспечивалась сбалансированная загрузка всех устройств вычислительной машины; так, например, в мультипрограммной смеси желательно одновременное присутствие вычислительных задач и задач с интенсивным вводом-выводом. Таким образом, выбор нового задания из пакета заданий зависит от внутренней ситуации, складывающейся в системе, т.е. выбирается "выгодное" задание. Следовательно, в таких ОС невозможно гарантировать выполнение того или иного задания в течение определенного периода времени. В системах пакетной обработки переключение процессора с выполнения одной задачи на выполнение другой происходит только в случае, если активная задача сама отказывается от процессора, например, из-за необходимости выполнить операцию ввода-вывода. Поэтому одна задача может надолго занять процессор, что делает невозможным выполнение интерактивных задач. Таким образом, взаимодействие пользователя с вычислительной машиной, на которой установлена система пакетной обработки, сводится к тому, что он приносит задание, отдает его диспетчеру-оператору, а в конце дня после выполнения всего пакета заданий получает результат. Очевидно, что такой порядок снижает эффективность работы пользователя.

Системы разделения времени призваны исправить основной недостаток систем пакетной обработки – изоляцию пользователя-программиста от процесса выполнения его задач. Каждому пользователю системы разделения времени предоставляется терминал, с которого он может вести диалог со своей программой. Так как в системах разделения времени каждой задаче выделяется только квант процессорного времени, ни одна задача не занимает процессор надолго, н время ответа оказывается приемлемым. Если квант выбран достаточно небольшим, то у всех пользователей, одновременно работающих на одной и той же машине, складывается впечатление, что каждый из них единолично использует машину. Ясно, что системы разделения времени обладают меньшей пропускной способностью, чем системы пакетной обработки, так как на вы-

полнение принимается каждая запущенная пользователем задача, а не та, которая "выгодна" системе, и, кроме того, имеются накладные расходы вычислительной мощности на более частое переключение процессора с задачи на задачу. Критерием эффективности систем разделения времени является не максимальная пропускная способность, а удобство и эффективность работы пользователя.

Системы реального времени применяются для управления различными техническими объектами такими, например, как станок, спутник, научная экспериментальная установка или технологическими процессами такими, как гальваническая линия, доменный процесс и т.п. Во всех этих случаях существует предельно допустимое время, в течение которого должна быть выполнена та или иная программа, управляющая объектом, в противном случае может произойти авария: спутник выйдет из зоны видимости, экспериментальные данные, поступающие с датчиков, будут потеряны, толщина гальванического покрытия не будет соответствовать норме. Таким образом, критерием эффективности для систем реального времени является их способность выдерживать заранее заданные интервалы времени между запуском программы и получением результата (управляющего воздействия). Это время называется временем реакции системы, а соответствующее свойство системы – реактивностью. Для этих систем мультипрограммная смесь представляет собой фиксированный набор заранее разработанных программ, а выбор программы на выполнение осуществляется, исходя из текущего состояния объекта или в соответствии с расписанием плановых работ.

Некоторые операционные системы могут совмещать в себе свойства систем разных типов, например, часть задач может выполняться в режиме пакетной обработки, а часть – в режиме реального времени или в режиме разделения времени. В таких случаях режим пакетной обработки часто называют фоновым режимом.

Особенности методов построения

При описании операционной системы часто указываются особенности ее структурной организации и основные концепции, положенные в ее основу.

К таким базовым концепциям относятся. Способы построения ядра системы – монолитное ядро или микроядерный подход. Большинство ОС использует монолитное ядро, которое компонуется как одна программа, работающая в привилегированном режиме и использующая быстрые переходы с одной процедуры на другую, не требующие переключения из привилегированного режима в пользовательский и наоборот. Альтернативой является построение ОС на базе микроядра, работающего также в привилегированном режиме и выполняющего только минимум функций по управлению аппаратурой, в то время как функции ОС более высокого уровня выполняют специализированные компоненты ОС – серверы, работающие в пользовательском режиме. При таком построении ОС работает более медленно, так как часто выполняются переходы между привилегированным режимом и пользовательским, зато система получается более гибкой – ее функции можно наращивать, модифицировать или сужать, добавляя, модифицируя или исключая серверы пользовательского режима. Кроме того, серверы хорошо защищены друг от друга, как и любые пользовательские процессы.

Построение ОС на базе объектно-ориентированного полхода дает возможность использовать все его достоинства, хорошо зарекомендовавшие себя на уровне приложении, внутри операционной системы, а именно аккумуляцию удачных решений в форме стандартных объектов, возможность создания новых объектов на базе имеющихся с помощью механизма наследования, хорошую защиту данных за счет их инкапсуляции во внутренние структуры объекта, что делает данные недоступными для несанкционированного использования извне, структурированность системы, состоящей из набора хорошо определенных объектов.

Наличие нескольких прикладных сред дает возможность в рамках одной ОС одновременно выполнять приложения, разработанные для нескольких ОС. Многие современные операционные системы поддерживают одновременно прикладные среды MS DOS, Windows, Unix (POSDC). OS/2 или хотя бы некоторого подмножества из этого популярного набора. Концепция множественных прикладных сред наиболее просто реализуется в ОС на базе микроядра, над которым работают различные серверы, часть которых реализуют прикладную среду той или иной операционной системы.

Распределенная организация операционной системы позволяет упростить работу пользователей и программистов в сетевых средах. В распределенной ОС реализованы механизмы, которые дают возможность пользователю представлять и воспринимать сеть в виде традиционного однопроцессорного компьютера. Характерными признаками распределенной организации ОС являются: наличие единой справочной службы разделяемых ресурсов, единой службы времени, использование механизма вызова удаленных процедур (RPC) для прозрачного распределения программных процедур по машинам, много-нитевой обработки, позволяющей распараллеливать вычисления в рамках одной задачи и выполнять эту задачу сразу на нескольких компьютерах сети, а также наличие других распределенных служб.

Кроме того, операционные системы можно также разделить на группы, используя различные признаки классификации (табл. 1).

Таблица 1. Классы операционных систем

Признак классификации	**Описание класса**
1. Мощность аппаратных средств	персональные компьютеры; мини ЭВМ; большие ЭВМ; суперЭВМ
2. Количество ЭВМ, обслуживаемых ОС	автономная ЭВМ; многомашинная ВС; сети ЭВМ (локальные и глобальные)
3. Тип системы обработки данных	система оперативной обработки; система пакетной обработки; система реального времени
4. Режим обработки данных	однопрограммный; пакетный мультипрограммный; разделения времени
5. Режим обслуживания заявок	одиночный отбор; групповой отбор; смешанный отбор
6. Дисциплина обслуживания заявок	без приоритетов; с приоритетами

Основные принципы построения операционных систем

Частотный принцип реализации системных программ основан на выделении в алгоритмах и в обрабатываемых массивах ОС действий и данных по частоте их использования. Следствием применения частотного принципа в современных ОС – наличие многоуровневого планирования при организации работы ОС.

Принцип модульности отражает технологические и эксплуатационные свойства системы, предусматривая оформление функционально законченных компонентов ОС в виде отдельных модулей.

Принцип функциональной избирательности предусматривает выделение некоторого множества важных модулей, которые должны быть постоянно в "горячем" режиме

для обеспечения эффективного управления вычислительным процессом. Этот выделенный набор модулей называют *ядром* ОС. При формировании состава ядра ОС ищут компромисс между двумя разноречивыми требованиями: в состав ядра должны войти наиболее часто используемые модули; объём памяти, занимаемый ядром ОС, должен быть как можно меньше. Программы ядра ОС постоянно находятся в оперативной памяти ЭВМ и называются *резидентными.* Программы ОС, подгружаемые в ОЗУ по мере необходимости из внешней памяти, называются *транзитными.*

Принцип генерируемости определяет такой способ исходного представления системной программы ОС, который позволяет настраивать эту системную программу, исходя из конкретной конфигурации аппаратных средств и круга решаемых проблем.

Принцип функциональной избыточности предусматривает обеспечение возможности выполнения одной и той же работы различными средствами.

Принцип перемещаемости предусматривает такое построение модулей ОС, при котором результаты работы не зависят от места их расположения.

Принцип защиты информации определяет необходимость разработки мер, ограждающих программы и данные пользователя от искажений или нежелательных влияний друг от друга, а также пользователей на ОС и обратно.

Принцип независимости программ от внешних устройств заключается в том. что связь программ с конкретными внешними устройствами осуществляется не на уровне подготовки программных устройств (трансляции или компиляции исходного кода, генерации выполняемого модуля), а в период планирования операционной системой ее выполнения.

Принцип открытости и наращиваемости ОС предусматривает возможность доступа к ней для анализа пользователями, специалистами, обслуживающим персоналом, а также изменения конфигурации ОС и ее мощности без осуществления процессов генерации.

Пользовательский интерфейс операционных систем

Как любое техническое устройство, компьютер обменивается информацией с человеком посредством набора определенных правил, обязательных как для машины, так и для человека. Эти правила в компьютерной литературе называются интерфейсом. Интерфейс может быть понятным и непонятным, дружественным и нет. К нему подходят многие прилагательные. Но в одном он постоянен: он есть, и никуда от него не денешься.

Интерфейс, по определению – это правила взаимодействия операционной системы с пользователями, а также соседних уровней в сети ЭВМ. От интерфейса зависит технология общения человека с компьютером.

Классификация интерфейсов

Как уже указывалось выше, интерфейс – это, прежде всего, набор правил. Как любые правила, их можно обобщить, собрать в "кодекс", сгруппировать по общему признаку. Таким образом, мы пришли к понятию "вид интерфейса" как объединение по схожести способов взаимодействия человека и компьютеров.

Современными видами интерфейсов являются:

1. Командный интерфейс – называется так по тому, что в этом виде интерфейса человек подает "команды" компьютеру, а компьютер их выполняет и выдает результат человеку. Командный интерфейс реализован в виде пакетной технологии и технологии командной строки.

2. WIMP-интерфейс (Window – окно. Image – образ. Menu – меню. Pointer – указатель). Характерной особенностью этого вида интерфейса является то, что диалог с

пользователем ведется не с помощью команд, а с помощью графических образов – меню, окон, других элементов. Хотя и в этом интерфейсе подаются команды машине, но это делается "©посредственно", через графические образы. Этот вид интерфейса реализован на двух уровнях технологий: простой графический интерфейс и "чистый" WIMP – интерфейс.

3. SILK-интерфейс (Speech – речь. Image – образ. Language – язык, Knowlege – знание). Этот вид интерфейса наиболее приближен к обычной, человеческой форме общения. В рамках этого интерфейса идет обычный "разговор" человека и компьютера. При этом компьютер находит для себя команды, анализируя человеческую речь и находя в ней ключевые фразы Результат выполнения команд он также преобразует в понятную человеку форму. Этот вид интерфейса наиболее требователен к аппаратным ресурсам компьютера, и поэтому его применяют в основном для военных целей.

4. Общественный интерфейс – основан на семантических сетях.

Пакетная технология

Исторически вид пакетной технологии появился первым. Она существовала уже на релейных машинах Зюса и Цюзе (Германия, 1937 г.). Идея ее проста: на вход компьютера подается последовательность символов, в которых по определенным правилам указывается последовательность запушенных на выполнение программ. После выполнения очередной программы запускается следующая и т.д. Машина по определенным правилам находит для себя команды и данные. В качестве этой последовательности может выступать, например, перфолента, стопка перфокарт, последовательность нажатия клавиш электрической пишущей машинки (типа CONSUL). Машина также выдает свои сообщения на перфоратор, алфавитно-цифровое печатающее устройство (АЦПУ), ленту пишущей машинки.

Такая машина представляет собой "черный ящик" (точнее "белый шкаф"), в который постоянно подается информация и которая также постоянно "информирует" мир о своем состоянии. Человек здесь имеет малое влияние на работу машины – он может лишь приостановить работу машины, сменить программу и вновь запустить ЭВМ. Впоследствии, когда машины стали помощнее и могли обслуживать сразу нескольких пользователей, вечное ожидание пользователей типа: "Я послал данные машине. Жду, что она ответит. И ответит ли вообще?" – стало, мягко говоря, надоедать. К тому же вычислительные центры, вслед за газетами, стали вторым крупным "производителем" макулатуры. Поэтому с появлением алфавитно-цифровых дисплеев началась эра по-настоящему пользовательской технологии – командной строки.

Технология командной строки

При технологии командной строки в качестве единственного способа ввода информации от человека к компьютеру служит клавиатура, а компьютер выводит информацию человеку с помощью алфавитно-цифрового дисплея (монитора). Эту комбинацию (монитор + клавиатура) стали называть терминалом, или консолью.

Команды набираются в командной строке, которая представляет собой символ приглашения и мигающий прямоугольник – курсор. При нажатии клавиши на месте курсора появляются символы, а сам курсор смешается вправо. Это очень похоже на набор команды на пишущей машинке. Однако, в отличие от нее, буквы отображаются на дисплее, а не на бумаге, и неправильно набранный символ можно стереть. Команда заканчивается нажатием клавиши Enter (или Return.) После этого осуществляется переход в начало следующей строки. Именно с этой позиции компьютер выдает на монитор результаты своей работы. Затем процесс повторяется.

Технология командной строки уже работала на монохромных алфавитно-цифровых дисплеях. Поскольку вводить позволялось только буквы, цифры и знаки препинания, то технические характеристики дисплея были не существенны. В качестве монитора можно было использовать телевизионный приемник и даже трубку осциллографа.

Обе эти технологии реализуются в виде командного интерфейса – машине подаются на вход команды, а она как бы "отвечает" на них.

Преобладающим видом файлов при работе с командным интерфейсом стали текстовые файлы – их и только их можно было создать при помощи клавиатуры. На время наиболее широкого использования интерфейса командной строки приходится появление операционной системы Unix и появление первых восьмиразрядных персональных компьютеров с многоплатформенной операционной системой CP/M.

Графический интерфейс

Идея графического интерфейса зародилась в середине 70-х годов, когда в исследовательском центре Xerox Palo Alto Research Center (PARC) была разработана концепция визуального интерфейса. Предпосылкой графического интерфейса явилось уменьшение времени реакции компьютера на команду, увеличение объема оперативной памяти, а также развитие технической базы компьютеров. Аппаратным основанием концепции, конечно же, явилось появление алфавитно-цифровых дисплеев на компьютерах, причем на этих дисплеях уже имелись такие эффекты, как "мерцание" символов, инверсия цвета (смена начертания белых символов на черном фоне обратным, т.е. черных символов на белом фоне), подчеркивание символов. Эти эффекты распространились не на весь экран, а только на один или более символов. Следующим шагом явилось создание цветного дисплея, позволяющего выводить, вместе с этими эффектами, символы в 16 цветах на фоне с палитрой (т.е. цветовым набором) из 8 цветов. После появления графических дисплеев, с возможностью вывода любых графических изображений в виде множества точек на экране различного цвета, фантазии в использовании экрана вообще не стало границ! Первая система с графическим интерфейсом 8010 Star Information System группы PARC, таким образом, появилась за четыре месяца до выхода в свет первого компьютера фирмы IBM в 1981 г. Первоначально визуальный интерфейс использовался только в программах. Постепенно он стал переходить и на операционные системы, используемые сначала на компьютерах Atari и Apple Macintosh, а затем и на IBM-совместных компьютерах.

С более раннего времени и под влиянием также и этих концепций проходил процесс по унификации в использовании клавиатуры и мыши прикладными программами. Слияние этих двух тенденций и привело к созданию того пользовательского интерфейса, с помощью которого, при минимальных затратах времени и средств на переучивание персонала, можно работать с любым программным продуктом. Описание этого интерфейса, общего для всех приложений и операционных систем, и посвящена данная часть.

Графический интерфейс пользователя за время своего развития прошел две стадии. Об эволюции графического интерфейса с 1974 г. по настоящее время будет рассказано ниже.

Простой графический интерфейс

На первом этапе графический интерфейс очень походил на технологию командной строки. Отличия от технологии командной строки заключались в следующем:

1. При отображении символов допускалось выделение части символов цветом, инверсным изображением, подчеркиванием и мерцанием. Благодаря этому повысилась выразительность изображения.

2. В зависимости от конкретной реализации графического интерфейса курсор может представляться не только мерцающим прямоугольником, но и некоторой областью, охватывающей несколько символов и лаже часть экрана. Эта выделенная область отличается от других, невыделенных частей (обычно цветом).

3. Нажатие клавиши Enter не всегда приводит к выполнению команды и переходу к следующей строке. Реакция на нажатие любой клавиши во многом зависит от того, в какой части экрана находится курсор.

4. Кроме клавиши Enter, на клавиатуре все чаше стали использоваться "серые" клавиши управления курсором.

5. Уже в этой редакции графического интерфейса стали использоваться манипуляторы (типа мыши, трекбола и т.п.). Они позволяли быстро выделять нужную часть экрана и перемещать курсор.

Подводя итоги, можно привести следующие отличительные особенности этого интерфейса:

1) выделение областей экрана;

2) переопределение клавиш клавиатуры в зависимости от контекста;

3) использование манипуляторов и серых клавиш клавиатуры для управления курсором;

4) широкое использование цветных мониторов.

Появление этого типа интерфейса совпадает с широким распространением операционной системы MS DOS. Именно она внедрила этот интерфейс в массы, благодаря чему 80-е годы прошли под знаком совершенствования этого типа интерфейса, улучшения характеристик отображения символов и других параметров монитора.

Типичным примером использования этого вида интерфейса является файловая оболочка Norton Commander (о файловых оболочках смотри ниже) и текстовый редактор Multi-Edit. А текстовые редакторы Лексикон, ChiWriter и текстовый процессор Microsoft Word for Dos являются примером, как этот интерфейс превзошел сам себя.

WIMP-интерфейс

Вторым этапом в развитии графического интерфейса стал "чистый" интерфейс WIMP. Этот подвид интерфейса характеризуется следующими особенностями:

1. Вся работа с программами, файлами и документами происходит в окнах – определенных очерченных рамкой частях экрана.

2. Все программы, файлы, документы, устройства и другие объекты представляются в виде значков – иконок. При открытии иконки превращаются в окна.

3. Все действия с объектами осуществляются с помощью меню. Хотя меню появилось на первом этапе становления графического интерфейса, оно не имело в нем главенствующего значения, а служило лишь дополнением к командной строке. В чистом WTMP – интерфейсе меню становится основным элементом управления.

4. Широкое использование манипуляторов для указания на объекты. Манипулятор перестает быть просто игрушкой – дополнением к клавиатуре, а становится основным элементом управления. С помощью манипулятора УКАЗЫВАЮТ на любую область экрана, окна или иконки, ВЫДЕЛЯЮТ ее, а уже потом через меню или с использованием других технологий осуществляют управление ими.

Следует отметить, что WIMP требует для своей реализации цветной растровый дисплей с высоким разрешением и манипулятор. Также программы, ориентированные на этот вид интерфейса, предъявляют повышенные требования к производительности

компьютера, объему его памяти, пропускной способности шины и т.п. Однако этот вид интерфейса наиболее прост в усвоении и интуитивно понятен. Поэтому сейчас WTMP-интерфейс стал стандартом де-факто.

Ярким примером программ с графическим интерфейсом является операционная система Microsoft Windows.

Речевая технология

С середины 90-х гг. XX в., после появления недорогих звуковых карт и широкого распространения технологий распознавания речи, появилась так называемая "речевая технология" – SILK-интерфейс. При этой технологии команды подаются голосом путем произнесения специальных зарезервированных слов – команд, например:

- "Проснись" – включение голосового интерфейса;
- "Отдыхай" – выключение речевого интерфейса;
- "Открыть" – переход в режим вызова той или иной программы. Имя программы называется в следующем слове;
- "Буду диктовать" – переход из режима команд в режим набора текста голосом;
- "Режим команд" – возврат в режим подачи команд голосом и некоторые другие.

Слова должны выговариваться четко, в одном темпе. Между словами обязательна пауза. Из-за неразвитости алгоритма распознавания речи такие системы требует индивидуальной предварительной настройки на каждого конкретного пользователя.

"Речевая" технология является простейшей реализацией SILK-интерфейса.

Биометрическая технология

Биометрическая технология возникла в конце 90-х годов XX в. и на момент написания книги еще разрабатывается. Для управления компьютером используется выражение лица человека, направление его взгляда, размер зрачка и другие признаки. Для идентификации пользователя используется рисунок радужной оболочки его глаз, отпечатки пальцев и другая уникальная информация. Изображения считываются с цифровой видеокамеры, а затем с помощью специальных программ распознавания образов из этого изображения выделяются команды. Эта технология, по-видимому, займет свое место в программных продуктах и приложениях, где важно точно идентифицировать пользователя компьютера.

Семантический интерфейс

Семантический интерфейс возник в конце 70-х годов XX в., с развитием искусственного интеллекта. Его трудно назвать самостоятельным видом интерфейса – он включает в себя и интерфейс командной строки, и графический, и речевой, и мимический интерфейс. Основная его отличительная черта – это отсутствие команд при общении с компьютером. Запрос формируется на естественном языке, в виде связанного текста и образов. По своей сути это трудно называть интерфейсом – это уже моделирование "общения" человека с компьютером.

С середины 90-х годов XX в. автор уже не встречал публикаций, относящихся к семантическому интерфейсу. Похоже, что в связи с важным военным значением этих разработок (например, для автономного ведения современного боя машинами-роботами, для "семантической" криптографии) эти направления были засекречены. Информация, что эти исследования продолжаются, иногда появляется в периодической печати (обычно в разделах компьютерных новостей).

Система ввода-вывода

В состав любой операционной системы входят программные модули, обеспечивающие управление устройствами ввода-вывода ЭВМ. Эти программные модули называют драйверами устройств, а совокупность драйверов ввода-вывода образует систему ввода-вывода, входящую в состав операционной системы.

Драйвер устройства (Device driver) – программа, обеспечивающая взаимодействие операционной системы с физическим устройством.

Система ввода-вывода (Input-Output System) – часть операционной системы, обеспечивающая управление внешними устройствами, подключенными к ЭВМ.

Основной задачей системы ввода-вывода является обеспечение непрерывной организации (планирования, управления) и двусторонней передачи данных между основной памятью и внешними устройствами с целью достижения максимального перекрытия во времени работы этой аппаратуры и процессора.

Состав систем ввода-вывода и, следовательно, перечень драйверов устройств в различных операционных системах не совпадают, что объясняется имеющимися отличиями в аппаратуре ввода-вывода, а также множеством методов, используемых для управления этой аппаратурой. Вместе с тем в большинстве операционных систем существует некоторое ядро системы ввода-вывода, получившее название базовой системы ввода-вывода.

Базовая система ввода-вывода (BIOS – Basic Input Output System) – часть программного обеспечения ЭВМ, поддерживающая управление адаптерами внешних устройств и представляющая стандартный интерфейс для обеспечения переносимости операционных систем между ЭВМ с одинаковым процессором. Базовая система ввода-вывода, как правило, разрабатывается изготовителем ЭВМ, хранится в постоянном запоминающем устройстве и рассматривается как часть ЭВМ.

При построении систем ввода-вывода аппаратура ввода-вывода рассматривается как совокупность аппаратурных процессоров, которые способны работать параллельно и независимо друг от друга, а также относительно центрального процессора. На таких процессорах развиваются так называемые внешние процессы.

Внешние процессы, используя аппаратуру ввода-вывода, могут взаимодействовать как между собой, так и с внутренними процессами, которые развиваются на центральном процессоре. Важным фактом является то, что скорости развития внешних и внутренних процессов существенно различаются, причем эти различия могут достигать нескольких порядков.

Система управления вводом-выводом представляет собой одни или несколько системных процессов (т.е. процессов, принадлежащих операционной системе), обеспечивающих информационное и управляющее взаимодействие внутренних и внешних процессов. Через эту систему происходит инициализация, управление развитием и уничтожение внешних процессов.

С точки зрения внутренних (программных) процессов-пользователей система управления вводом-выводом представляет собой программный интерфейс с необходимыми для этих процессов внешними устройствами. В составе этого интерфейса пользователь имеет возможность выражать запросы на выполнение действий в отношении внешних устройств. При этом различают три типа действий: операции чтения и записи данных, операции управления устройством, операции по проверке состояния устройств. При построении систем управления вводом-выводом руководствуются стремлением сделать большинство ее компонентов "невидимыми" для пользователей, что достигается созданием развитых драйверов внешних устройств с понятным интерфейсом и доступными из любой системы программирования.

Для сглаживания эффекта несоответствия скоростей между внутренними и внешними процессами в системах управления вводом-выводом применяют три основных метода: синхронизация по прерываниям ввода-вывода, буферизация ввода-вывода, блокирование данных.

Для синхронизации параллельной работы могут применяться различные методы, среди которых наиболее совершенными являются средства, основанные на использовании системы прерывания. Канал ввода-вывода через систему прерываний прерывает работу центрального процессора всякий раз при завершении операции ввода-вывода или при возникновении ошибки. Такие сигналы прерывания являются по своему смыслу синхронизирующими, так как они используются для оповещения определенного внутреннего процесса о событии, которое произошло при работе канала ввода-вывода или внешнего устройства.

Одной из главных функций ОС является управление всеми устройствами ввода-вывода компьютера. ОС должна передавать устройствам команды, перехватывать прерывания и обрабатывать ошибки: она также должна обеспечивать интерфейс между устройствами и остальной частью системы. В целях развития интерфейс должен быть одинаковым для всех типов устройств (независимость от устройств).

Физическая организация устройств ввода-вывода

Устройства ввода-вывода делятся на два типа: блок-ориентированные и байт-ориентированные устройства. Блок-ориентированные устройства хранят информацию в блоках фиксированного размера, каждый из которых имеет свой собственный адрес. Самое распространенное блок-ориентированное устройство – диск. Байт-ориентированные устройства не адресуемы и не позволяют производить операцию поиска, они генерируют или потребляют последовательность байтов. Примерами являются терминалы, строчные принтеры, сетевые адаптеры. Однако некоторые внешние устройства не относятся ни к одному классу, например часы, которые, с одной стороны, не адресуемы, а с другой – не порождают потока бантов. Это устройство только выдает сигнал прерывания в некоторые моменты времени.

Внешнее устройство обычно состоит из механического и электронного компонента. Электронный компонент называется контроллером устройства или адаптером. Механический компонент представляет собственно устройство. Некоторые контроллеры могут управлять несколькими устройствами. Если интерфейс между контроллером и устройством стандартизован, то независимые производители могут выпускать совместимые как контроллеры, так и устройства.

Операционная система обычно имеет дело не с устройством, а с контроллером. Контроллер, как правило, выполняет простые функции, например, преобразует поток бит в блоки, состоящие из байт, и осуществляет контроль и исправление ошибок. Каждый контроллер имеет несколько регистров, которые используются для взаимодействия с центральным процессором. В некоторых компьютерах эти регистры являются частью физического адресного пространства. В таких компьютерах нет специальных операций ввода-вывода. В других компьютерах адреса регистров ввода-вывода, называемых часто портами, образуют собственное адресное пространство за счет введения специальных операций ввода-вывода (например команд IN и OUT в процессорах i86).

ОС выполняет ввод-вывод, записывая команды в регистры контроллера. Например, контроллер гибкого диска ЮМ РС принимает 15 команд, таких как READ. WRITE. SEEK, FORMAT и т.д. Когда команда принята, процессор оставляет контроллер и занимается другой работой. При завершении команды контроллер организует прерывание для того, чтобы передать управление процессором операционной системе, которая

должна проверить результаты операции. Процессор ползает результаты и статус устройства, читая информацию из регистров контроллера.

Организация программного обеспечения ввода-вывода

Уровни организации программного обеспечения ввода-вывода

Основная идея организации программного обеспечения ввода-вывода состоит в разбиении его на несколько уровнен, причем нижние уровни обеспечивают экранирование особенностей аппаратуры от верхних, а те. в свою очередь, обеспечивают удобный интерфейс для пользователей.

Ключевым принципом является независимость от устройств. Вид программы не должен зависеть от того, читает ли она данные с гибкого или жесткого диска.

Очень близкой к идее независимости от устройств является идея единообразного именования, т.е. для именования устройств должны быть приняты единые правила.

Другим важным вопросом для программного обеспечения ввода-вывода является обработка ошибок. Вообще говоря, ошибки следует обрабатывать как можно ближе к аппаратуре. Если контроллер обнаруживает ошибку чтения, то он должен попытаться ее скорректировать. Если же это ему не удается, то исправлением ошибок должен заняться драйвер устройства. Многие ошибки могут исчезать при повторных попытках выполнения операций ввода-вывода, например ошибки, вызванные наличием пылинок на головках чтения или на диске. И только если нижний уровень не может справиться с ошибкой, он сообщает об ошибке верхнему уровню.

Еще один ключевой вопрос – это использование блокирующих (синхронных) и неблокирующих (асинхронных) передач. Большинство операций физического ввода-вывода выполняется асинхронно – процессор начинает передачу и переходит на другую работу, пока не наступает прерывание. Пользовательские программы намного легче писать, если операции ввода-вывода блокирующие – после команды READ программа автоматически приостанавливается до тех пор, пока данные не попадут в буфер программы. ОС выполняет операции ввода-вывода асинхронно, но представляет их для пользовательских программ в синхронной форме.

Последняя проблема состоит в том, что одни устройства являются разделяемыми, а другие – выделенными. Диски – это разделяемые устройства, так как одновременный доступ нескольких пользователей к диску* не представляет собой проблему. Принтеры – это выделенные устройства, потому что нельзя смешивать строчки, печатаемые различными пользователями. Наличие выделенных устройств создает для операционной системы некоторые проблемы.

Для решения поставленных проблем целесообразно разделить программное обеспечение ввода-вывода на четыре слоя (рис. 3):

1) обработка прерываний:
2) драйверы устройств;
3) независимый от устройств слой операционной системы;
4) пользовательский слой программного обеспечения.

Рис. 3. Многоуровневая организация программного обеспечения системы ввода-вывода

Обработка прерываний

Прерывания должны быть скрыты как можно глубже в недрах операционной системы, чтобы как можно меньшая часть ОС имела с ними дело. Наилучший способ состоит в разрешении процессу, инициировавшему операцию ввода-вывода, блокировать себя до завершения операции и наступления прерывания. Процесс может блокировать себя, используя, например, вызов DOWN для семафора, или вызов WAIT для переменной условия, или вызов RECEIVE для ожидания сообщения. При наступлении прерывания процедура обработки прерывания выполняет разблокирование процесса, инициировавшего операцию ввода-вывода, используя вызовы UP. SIGNAL или посылая процессу сообщение. В любом случае эффект от прерывания будет состоять в том, что ранее заблокированный процесс теперь продолжит свое выполнение.

Драйверы устройств

Весь зависимый от устройства код помешается в драйвер устройства. Каждый драйвер управляет устройствами одного типа или, может быть, одного класса.

В операционной системе только драйвер устройства знает о конкретных особенностях какого-либо устройства. Например, только драйвер диска имеет дело с дорожками, секторами, цилиндрами, временем установления головки и другими факторами, обеспечивающими правильную работу диска.

Драйвер устройства принимает запрос от устройств программного слоя и решает, как его выполнить. Типичным запросом является чтение и блоков данных. Если

драйвер был свободен во время поступления запроса, то он начинает выполнять запрос немедленно. Если же он был занят обслуживанием другого запроса, то вновь поступивший запрос присоединяется к очереди уже имеющихся запросов, и он будет выполнен, когда наступит его очередь.

Первый шаг в реализации запроса ввода-вывода, например для диска, состоит в преобразовании его из абстрактной формы в конкретную. Для дискового драйвера это означает преобразование номеров блоков в номера цилиндров, головок, секторов: проверку, работает ли мотор, находится ли головка над нужным цилиндром. Короче говоря, он должен решить, какие операции контроллера нужно выполнить и в какой последовательности.

После передачи команды контроллеру драйвер должен решить, блокировать ли себя до окончания заданной операции или нет. Если операция занимает значительное время, как при печати некоторого блока данных, то драйвер блокируется до тех пор, пока операция не завершится, и обработчик прерывания не разблокирует его. Если команда ввода-вывода выполняется быстро (например, прокрутка экрана), то драйвер ожидает ее завершения без блокирования.

Независимый от устройств слой операционной системы

Большая часть программного обеспечения ввода-вывода является независимой от устройств. Точная граница между драйверами и независимыми от устройств программами определяется системой, так как некоторые функции, которые могли бы быть реализованы независимым способом, в действительности выполнены в виде драйверов для повышения эффективности или по другим причинам.

Типичными функциями для независимого от устройств слоя являются:

- обеспечение общего интерфейса к драйверам устройств;
- именование устройств;
- защита устройств:
- обеспечение независимого размера блока;
- буферизация;
- распределение памяти на блок-ориентированных устройствах;
- распределение и освобождение выделенных устройств;
- уведомление об ошибках.

Остановимся на некоторых функциях данного перечня. Верхним слоям программного обеспечения неудобно работать с блоками разной величины, поэтому данный слой обеспечивает единый размер блока, например, за счет объединения нескольких различных блоков в единый логический блок. В связи с этим верхние уровни имеют дело с абстрактными устройствами, которые используют единый размер логического блока независимо от размера физического сектора.

При создании файла или заполнении его новыми данными необходимо выделить ему новые блоки. Для этого ОС должна вести список или битовую карту свободных блоков диска. На основании информации о наличии свободного места на диске может быть разработан алгоритм поиска свободного блока, независимый от устройства и реализуемый программным слоем, находящимся выше слоя драйверов.

Параллельные процессы и критические участки

Рабочая смесь, созданная механизмами высшего и внутреннего планирования, представляет собой совокупность параллельных процессов.

Процессы называются *параллельными,* если они существуют в системе одновременно.

Параллельные процессы называются *независимыми,* если они работают без целенаправленной передачи сигналов информации друг другу.

Параллельные процессы называются *связанными,* если осуществляется направленный обмен сигналами (информацией) между ними.

Связанные параллельные процессы бывают *синхронными,* когда обеспечиваются согласование скоростей их развития в системе, и *асинхронными,* если скорости протекания процессов не регулируются в системе.

В механизмах синхронизации нуждаются все виды параллельных процессов, функционирующих в мультипрограммное вычислительной системе.

Не связанные между собой процессы также нуждаются в синхронизации своей работы. Это объясняется тем, что они используют во время функционирования одни и те же физические и логические внешние устройства, которые в каждый конкретный момент времени могут обслуживать только один процесс (критические ресурсы).

Ресурс системы называется *критическим,* если он допускает в каждый момент времени обслуживание только одного процесса.

Общим принципом, положенным в основу всех механизмов синхронизации процессов, является принцип взаимоисключения.

Пришит взаимоисключения: каждый процесс, обращающий к разделяемым (критическим) ресурсам, должен исключить возможность для всех других процессов одновременного с ним использования этого ресурса.

Использование принципа взаимоисключения требует встраивания в программы процессов механизмов синхронизации, обеспечивающих выполнение следующих условий:

• при обращении нескольких процессов к одному разделяемому ресурсу только одному из них разрешено воспользоваться этим ресурсом;

• в каждый момент времени только один процесс должен владеть критическим ресурсом.

Все механизмы синхронизации, реализующие принцип взаимоисключения, основаны на применении концепции критического участка программы.

Критическим участком, критической областью программы процесса называется тот отрезок программного кода процесса, на котором этот процесс образуется к критическому ресурсу.

Количество критических участков в процессе зависит только от того, к ресурсам какого вида он обращается при своем функционировании.

Когда некоторый процесс находится на своем критическом участке, другие процессы могут продолжать выполнение, но без входа в их критические участки (занятым критическим ресурсам), когда процесс выходит из критического участка, то должно быть разрешено использование освобожденного критического ресурса. Обеспечение взаимоисключения является основной проблемой параллельного программирования.

Примитивы взаимоисключения

Общим подходом к построению механизмов синхронизации с использованием концепции критических участков является применение примитивов взаимоисключения.

Примитивами взаимоисключения называется программная конструкция, обеспечивающая реализацию взаимоисключений для параллельных процессов. В обобщенном виде можно указать два примитива взаимоисключений:

1) примитив *вход_взаимоисключеиия* – с его помощью фиксируется захват критического ресурса данным процессом и устанавливается запрет на использование его другими процессами;

2) примитив *выход_взаимоиск1ючеиия* – с его помощью процесс сообщает об освобождении им критического ресурса.

Простейший алгоритм синхронизации с применением примитивов взаимоисключения состоит в следующем. Главный процесс запускает в работу два параллельных процесса Пх и П.2, после чего он может закончить свою работу. Каждый из параллельных процессов в своем теле имеет области работы с критическим ресурсом. Эти области обязательно обрамляются примитивами *вход_взаимоиас1ючения* и *выход_взаимоисключения.*

В рассматриваемом случае двух процессов эти примитивы работают следующим образом. Когда процесс П1 выполняет примитив *вход_взаимоисключения* и если при этом процесс П2 находится вне своего критического участка, то П1 входит в свой критический участок, выполняет работу с критическим ресурсом, а затем выполняет примитив вы-ход_взаимоисключения, показывая тем самым, что он вышел из своего критического участка. Если П1 выполняет *вход_взаимоисключения,* в то время как П2 находится на своем критическом участке, то процесс *П1* переводится в состояние ожидания, пока процесс Щ не выполнит *выход_взаимоискчючеш)я*. Только после этого процесс П1 может выйти из состояния ожидания и войти в свой критический участок.

Если процессы П1 и П2 выполняют *вход_взаимоисключения* одновременно, то одному из них операционная система разрешает продолжить работу, а другой переводит в состояние ожидания.

Программная реализация примитивов взаимоисключения

Программная реализация примитивов взаимоисключения осуществляется с соблюдением следующих ограничений:

1) задача должна быть решена чисто программным способом на машине, не имеющей специальных команд взаимоисключения;

2) не должно быть никаких предположений об относительных скоростях выполнения асинхронных параллельных процессов;

3) процессы, находящиеся вне своих критических участков, не могут препятствовать другим процессам входить в их собственные критические участки;

4) не должно быть бесконечного откладывания момента входа процессов в их критические участки.

Контрольные вопросы

1. Дать определение и характеристику основных режимов работы, дисциплин и режимов обслуживания заявок в вычислительных системах.
2. Дать определение и характеристику классов программных средств.
3. Изложить классификацию ОС.
4. Охарактеризовать основные принципы построения ОС.
5. Перечислить виды интерфейсов ОС. Охарактеризовать пакетную технологию как интерфейс. Дать описание интерфейса командной строки.
6. Дать описание графических интерфейсов. В каких ОС они применяются?
7. Охарактеризовать речевую технологию как интерфейс.
8. Охарактеризовать биометрическую технологию как интерфейс.
9. Охарактеризовать семантический интерфейс.
10. Что такое данные, источник данных, организация данных?
11. Перечислите методы организации данных. В чем их различия?
12. Опишите способы организации файлов.
13. Как можно хранить файлы на носителе?
14. Перечислите основные операции над файлами.
15. Перечислите и опишите уровни многоуровневой модели файловой системы.

16. Каковы основные компоненты архитектуры современных файловых систем?
17. Дайте определения системе ввода-вывода.
18. Что такое драйвер ввода-вывода?
19. Перечислите и охарактеризуйте типы устройств ввода-вывода.
20. На какие слои (уровни) разбито программное обеспечение ввода-вывода, каково их назначение?

2. Защита информации в современных ОС. Угрозы безопасности ОС. Требования к защите ОС. Разграничение доступа в ОС. Идентификация и аутентификация пользователей ОС. Аудит в ОС.

Информационная безопасность – сравнительно молодая, быстро развивающаяся область информационных технологий. Под *информационной безопасностью* будем понимать защищённость информации и поддерживающей инфраструктуры от случайных или преднамеренных воздействии естественного или искусственного характера, которые могут нанести неприемлемый ущерб субъектам информационных отношений, в том числе владельцам и пользователям информации и поддерживающей инфраструктуры.

Защита информации – это комплекс мероприятий, направленных на обеспечение информационной безопасности.

С методологической точки зрения правильный подход к проблемам информационной безопасности начинается с выявления *субъектов информационных отношении* и интересов этих субъектов, связанных с использованием информационных систем (ИС). Угрозы информационной безопасности – это оборотная сторона использования информационных технологий.

Информационная безопасность – многогранная область деятельности, в которой успех может принести только систематический, комплексный подход. Для решения данной проблемы рассматриваются меры *законодательного, административного, процедурного и программно-технического уровня.*

Спектр интересов субъектов, связанных с использованием информационных систем, можно разделить на следующие категории: обеспечение *доступности, целостности* и *конфиденциальности* информационных ресурсов и поддерживающей инфраструктуры.

Предмет защиты информации

Успех практически любой деятельности в немалой степени зависит от умения распоряжаться такой ценностью, как информация. В законе РФ "Об информации, информатизации и защите информации" определено:

- *" информационные ресурсы* являются объектами собственности граждан, организаций, общественных объединений, государства";
- *"информация* – сведения о лицах, предметах, событиях, явлениях и процессах (независимо от формы их представления), отраженные на материальных носителях, используемые в целях получения знаний и практических решений".

Информация имеет ряд особенностей:

- *не материальна;*
- *хранится и передается с помощью материальных носителей;*
- *любой материальный объект содержит информацию о самом себе либо о другом объекте. Информации присущи следующие свойства:*

Ценность информации определяется степенью ее полезности для владельца. Законом РФ "Об информации, информатизации и защите информации" гарантируется право собственника информации на ее использование и защиту от доступа к ней других лиц (организаций). Если доступ к информации ограничен, то такая информация называется *конфиденциальной.* Конфиденциальная информация может содержать *государственную* или *коммерческую* тайну.

Достоверность информации определяется достаточной для владельца точностью отражать объекты и процессы окружающего мира в определенных временных и пространственных рамках. Информация, искаженно представляющая действительность, может нанести владельцу значительный материальный и моральный ущерб. Если информация искажена умышленно, то ее называют *дезинформацией.*

Своевременность информации, т.е. соответствие ценности и достоверности определенному временному периоду, может быть выражена формулой

$$C(t) = C_0 e^{-2,3t/\tau},$$

где C_0 – ценность информации в момент ее возникновения; t – время от момента возникновения информации до момента определения ее стоимости; τ – время от момента возникновения информации до момента ее устаревания.

Предметом защиты является информация, хранящаяся, обрабатываемая и передаваемая в компьютерных (информационных) системах. Особенностями данного вида информации являются:

- *двоичное представление информации внутри системы, независимо от физической сущности носителей исходной информации;*
- *высокая степень автоматизации обработки и передачи информации;*
- *концентрация большого количества информации в КС.*

Объект защиты информации

Объектом зашиты информации является компьютерная (информационная) система или автоматизированная система обработки информации (АСОИ).

Информационная система – это организационно-упорядоченная совокупность информационных ресурсов, технических средств, технологии и персонала, реализующих информационные процессы в традиционном или автоматизированном режиме для удовлетворения информационных потребностей пользователей.

Информационная безопасность АСОИ – состояние рассматриваемой автоматизированной системы, при котором она, с одной стороны, способна противостоять дестабилизирующему воздействию внешних и внутренних информационных угроз, а с другой – ее наличие и функционирование не создает информационных угроз для элементов самой системы и внешней среды.

Информационная безопасность достигается проведением соответствующего уровня политики информационной безопасности.

Под *политикой информационной безопасности* понимают совокупность норм, правил и практических рекомендаций, регламентирующих работу средств зашиты АСОИ от заданного множества угроз безопасности.

Система защиты информации – совокупность правовых норм, организационных мер и мероприятий, технических, программных и криптографических средств и методов, обеспечивающих защищенность информации в системе в соответствии с принятой политикой безопасности.

Основные положения безопасности информационных систем

Что касается подходов к реализации защитных мероприятий по обеспечению безопасности информационных систем, то сложилась трехэтапная (трехстадийная) разработка таких мер. Первая стадия – *выработка требовании* – включает:

- *определение состава средств информационной системы;*
- *анализ уязвимых элементов ИС;*
- *оценка угроз (выявление проблем, возникающих при наличия уязвимых мест);*
- *анализ риска (прогноз возможных последствий, вызывающих эти проблемы).*

Вторая стадия – *определение способов зашиты* – включает ответы на следующие вопросы: Какие угрозы должны быть устранены и в какой мере? Какие ресурсы системы должны быть защищаемы и в какой степени? С помощью каких средств должна быть реализована защита?

Какова должна быть полная стоимость реализации защиты и затраты на эксплуатацию с учетом потенциальных угроз? Третья стадия – *определение функций* процедур и средств безопасности, реализуемых в виде некоторых механизмов зашиты.*

Основные принципы обеспечения информационной безопасности в АС

Для зашиты АС на основании руководящих документов Гостехкомиссии могут быть сформулированы следующие положения.

1. Информационная безопасность АС основывается на положениях требованиях существующих законов, стандартов и нормативно-методических документов.
2. Информационная безопасность АС обеспечивается комплексом программно-технических средств и поддерживающих их организационных мер.
3. Информационная безопасность АС должна обеспечиваться на всех технологических этапах обработки информации и во всех режимах функционирования, в том числе при проведении ремонтных и регламентных работ.
4. Программно-технические средства защиты не должны существенно ухудшать основные функциональные характеристики АС (надежность, быстродействие, возможность изменения конфигурации АС).
5. Неотъемлемой частью работ по ИБ является опенка эффективности средств защиты, осуществляемая по методике, учитывающей всю совокупность технических характеристик оцениваемого объекта, включая технические решения и практическую реализацию средств защиты.
6. Зашита АС должна предусматривать контроль эффективности средств зашиты. Этот контроль может быть периодическим либо инициироваться по мере необходимости пользователем АС или контролирующим органом.

Рассмотренные подходы могут быть реализованы при обеспечении следующих основных принципов: *Принцип системности.* Системный подход к защите информационных систем предполагает необходимость учета всех взаимосвязанных, взаимодействующих и изменяющихся во времени элементов, условий и факторов:

- *при всех видах информационного проявления и деятельности;*
- *во всех структурных элементах;*
- *при всех режимах функционирования;*
- *на всех этапах жизненного цикла;*
- *с учетом взаимодействия объекта защиты с внешней средой.*

Система защиты должна строиться не только с учетом всех известных каналов проникновения, но и с учетом возможности появления принципиально новых путей реализации угроз безопасности.

Принцип комплексности. В распоряжении специалистов по компьютерной безопасности имеется широкий спектр мер, методов и средств защиты компьютерных систем (современные СВТ, ОС, инструментальные и прикладные программные средства, обладающие теми или иными встроенными элементами защиты). Комплексное их использование предполагает согласование разнородных средств при построении целостной системы зашиты, перекрывающей все существенные каналы реализации угроз и не содержащей слабых мест на стыках отдельных ее компонентов.

Пришит непрерывности зашиты. Зашита информации – это не разовое мероприятие и даже не конкретная совокупность уже проведенных мероприятий и установленных средств защиты, а непрерывный целенаправленный процесс, предполагающий принятие соответствующих мер на всех этапах жизненного цикла АС. Разработка системы зашиты должна вестись параллельно с разработкой самой защищаемой системы. Это позволит учесть требования безопасности при проектировании архитектуры и, в конечном счете, позволит создать более эффективные (как по затратам ресурсов, так и по стойкости) защищенные системы. Большинству физических и технических средств зашиты для эффективного выполнения своих функций необходима постоянная организационная поддержка (своевременная смена и обеспечение правильного хранения и применения имен, паролей, ключей шифрования, переопределение полномочий и т.п.). Перерывы в работе средств защиты могут быть использованы злоумышленниками для анализа применяемых методов и средств защиты, внедрения специальных программных и аппаратных "закладок" и других средств преодоления системы защиты после восстановления ее функционирования.

Разумная достаточность. Создать абсолютно непреодолимую систему защиты принципиально невозможно, при достаточных времени и средствах можно преодолеть любую защиту. Поэтому имеет смысл вести речь только о некотором приемлемом уровне безопасности. Высокоэффективная система защиты стоит дорого, использует при работе существенную часть мощности и ресурсов ИС и может создавать ощутимые дополнительные неудобства пользователям. Важно правильно выбрать тот достаточный уровень зашиты, при котором затраты, риск и размер возможного ущерба были бы приемлемыми.

Гибкость системы защиты. Часто приходится создавать систему защиты в условиях большой неопределенности. Поэтому принятые меры и установленные средства защиты, особенно в начальный период их эксплуатации, могут обеспечивать как чрезмерный, так и недостаточный уровень зашиты. Для обеспечения возможности варьирования уровнем защищенности средства зашиты должны обладать определенной гибкостью. Особенно важно это свойство в тех случаях, когда средства зашиты необходимо устанавливать на работающую систем}', нарушая процесс ее нормального функционирования.

Открытость алгоритмов и механизмов защиты. Суть принципа открытости алгоритмов и механизмов зашиты состоит в том, что зашита не должна обеспечиваться только за счет секретности структурной организации и алгоритмов функционирования ее подсистем. Знание алгоритмов работы системы зашиты не должно давать возможности ее преодоления. Но это вовсе не означает, что информация конкретной системы зашиты должна быть общедоступна – необходимо обеспечивать защиту от угрозы раскрытия параметров системы.

Принцип простоты применения средств защиты. Механизмы зашиты должны быть интуитивно понятны и просты в использовании, применение средств защиты не должно быть связано со знанием специальных языков или с выполнением действии, требующих значительных дополнительных трудозатрат при обычной работе законных

пользователей, а также не должно требовать от пользователя выполнения рутинных непонятных ему операции (ввод нескольких паролей и имен и т.д.).

Анализ угроз информационной безопасности

Угроза – это потенциальная возможность определенным образом нарушить информационную безопасность.

Угроза – это потенциально возможно событие, действие (воздействие), процесс или явление, которое может привести к нанесению ущерба чьим-либо интересам.

Угрозой информационной безопасности АС называется возможность реализации воздействия на информацию, обрабатываемую АС, приводящего к искажению, уничтожению, копированию, блокированию доступа к информации, а также возможность воздействия на компоненты АС. приводящего к утрате, уничтожению или сбою функционирования носителя информации, средства взаимодействия с носителем или средства его управления.

Попытка реализации угрозы называется *атакой,* а тот, кто предпринимает такую попытку, – *злоумышленником.* Потенциальные злоумышленнике называются *источниками угрозы.*

Чаще всего угроза является следствием наличия *уязвимых* мест в защите информационных систем (таких как возможность доступа посторонних лиц к критически важному оборудованию или ошибки в программном обеспечении).

Промежуток времени от момента, когда появляется возможность использовать слабое место, и до момента, когда пробел ликвидируется, называется *окном опасности,* ассоциированным с данным уязвимым местом. Пока существует окно опасности, возможны успешные атаки на ИС.

Если речь идет об ошибках в ПО, то окно опасности "открывается" с появлением средств использования ошибки и ликвидируется при наложении заплат, ее исправляющих.

Для большинства уязвимых мест окно опасности существует сравнительно долго (несколько дней, иногда – недель), поскольку за это время должны произойти следующие события:

– должно стать известно о средствах использования пробела в защите;

– должны быть выпущены соответствующие заплаты:

– заплаты должны быть установлены в защищаемой ИС.

Новые уязвимые места и средства их использования появляются постоянно; это значит, во-первых, что почти всегда существуют окна опасности, и, во-вторых, что отслеживание таких окон должно производиться постоянно, а выпуск и наложение заплат – как можно более оперативно.

Некоторые угрозы нельзя считать следствием каких-то ошибок или просчетов, они существуют в силу самой природы современных ИС. Например, угроза отключения электричества или выхода его параметров за допустимые границы существует в силу зависимости аппаратного обеспечения ИС от качественного электропитания.

Рассмотрение наиболее распространенных угроз, которым подвержены современные информационные системы, дает представление о возможных угрозах, а также об уязвимых местах, которые эти угрозы обычно эксплуатируют, необходимо для того, чтобы выбирать наиболее экономичные средства обеспечения безопасности.

Угрозы, как и все в ИБ. зависят от интересов субъектов информационных отношений (и от того, какой ущерб является для них неприемлемым). Задание возможных угроз информационной безопасности проводится с целью определения полного перечня требований к разрабатываемой системе защиты. Перечень угроз, оценки вероятностей их реализации, а также модель нарушителя служат основой для анализа риска реа-

лизации угроз и формулирования требований к системе зашиты АС. Кроме выявления возможных угроз, должен быть проведен их анализ на основе классификационных признаков. Каждый из признаков классификации отражает одно из обобщенных требований к системе защиты. При этом угрозы, соответствующие каждому признаку классификации, позволяют детализировать отражаемое этим признаком требование.

Угрозы можно классифицировать но нескольким критериям:

1) *по аспекту информационной безопасности* (доступность, целостность, конфиденциальность), против которого угрозы направлены в первую очередь;

2) *по компонентам информационных систем,* на которые угрозы нацелены (данные, программы, аппаратура, поддерживающая инфраструктура):

3) *по способу осуществления* (случайные /преднамеренные действия природного/техногенного характера):

4) *по расположению источника угроз* (внутри/вне рассматриваемой ИС).

Необходимость классификации угроз ИБ АС обусловлена тем, что архитектура современных средств автоматизированной обработки информации, организационное, структурное и функциональное построение информационно-вычислительных систем и сетей, технологии и условия автоматизированной обработки информации такие, что накапливаемая, хранимая и

обрабатываемая информация подвержена случайным влияниям чрезвычайно большого числа факторов, в силу чего становится невозможным формализовать задачу описания полного множества угроз. Как следствие, для защищаемой системы определяют не полный перечень угроз, а перечень классов угроз.

Классификация всех возможных *угроз информационной безопасности* АС может быть проведена по ряду базовых признаков.

1. *По природе возникновения.*

1) *Естественные угрозы* – угрозы, вызванные воздействиями на АС и ее компоненты объективных физических процессов или стихийных природных явлений, независящих от человека.

2) *Искусственные угрозы* – угрозы информационной безопасности АС. вызванные деятельностью человека.

2. По степени преднамеренности проявления.

1) *Угрозы случайного действия и/или угрозы, вызванные ошибками или халатностью персонала.* Например:

- проявление ошибок программно-аппаратных средств АС;
- некомпетентное использование, настройка или неправомерное отключение средств зашиты персоналом службы безопасности;
- неумышленные действия, приводящие к частичному или полному отказу системы или разрушению аппаратных, программных, информационных ресурсов системы (неумышленная порча оборудования, удаление, искажение файлов с важной информацией или программ, в том числе системных и т.п.);
- неправомерное включение оборудования или изменение режимов работы устройств и программ;
- неумышленная порча носителей информации;
- пересылка данных по ошибочному адресу абонента (устройства);
- ввод ошибочных данных;
- неумышленное повреждение каналов связи.

2) *Угрозы преднамеренного действия* (например, угрозы действий злоумышленника для хищения информации).

3. ***По непосредственному источнику угроз.***

1) *Угрозы, непосредственным источником которых является природная среда* (стихийные бедствия, магнитные бури, радиоактивное излучение и т.п.).

2) *Угрозы, источником которых является человек:*

- внедрение агентов в число персонала системы (в том числе, возможно, и в административную группу, отвечающую за безопасность);
- вербовка (шлем подкупа, шантажа и т.п.) персонала или отдельных пользователей, имеющих определенные полномочия;
- угроза несанкционированного копирования секретных данных пользователем АС;
- разглашение, передача или утрата атрибутов разграничения доступа (паролей, ключей шифрования, идентификационных карточек, пропусков и т.п.).

3) *Угрозы, непосредственным источником которых являются санкционированные программно-аппаратные средства:*

- запуск технологических программ, способных при некомпетентном пользовании вызывать потерю работоспособности системы (зависания) или зацикливания) или необратимые изменения в системе (форматирование или реструктуризацию носителей информации, удаление данных и т.п.);
- возникновение отказа в работе операционной системы.

4) *Угрозы, непосредственным источником которых являются несанкционированные программно-аппаратные средства:*

- нелегальное внедрение и использование неучтенных программ (игровых, обучающих, технологических и других программ, не являющихся необходимыми для выполнения нарушителем своих служебных обязанностей) с последующим необоснованным расходованием ресурсов (загрузка процессора, захват оперативной памяти и памяти на внешних носителях);
- заражение компьютера вирусами с деструктивными функциями.

4. ***По положению источника угроз.***

1) *Угрозы, источник которых расположен вне контролируемой зоны территории {помещения), на которой находится АС:*

- перехват побочных электромагнитных, акустических и других излучений устройств и линий связи, а также наводок активных излучений на вспомогательные технические средства, непосредственно не участвующие в обработке информации (телефонные линии, сети питания, отопления и т.п.):
- перехват данных, передаваемых по каналам связи, и их анализ с целью выяснения протоколов обмена, правил вхождения в связь и авторизации пользователя и последующих попыток их имитации для проникновения в систему;
- дистанционная фото- и видеосъемка.

2) *Угрозы, источник которых расположен в пределах контролируемой зоны территории (помещения), на которой находится АС:*

- хищение производственных отходов (распечаток, записей, списанных носителей информации и т.п.);
- отключение или вывод из строя подсистем обеспечения функционирования вычислительных систем (электропитания, охлаждения и вентиляции, линий связи и т.д.);
- применение подслушивающих устройств.

3) *Угрозы, источник которых имеет доступ к периферийным устройствам АС* (терминалам).

4) *Угрозы, источник которых расположен в АС:*

- проектирование архитектуры системы и технологии обработки данных, разработка прикладных программ, которые представляют опасность для работоспособности системы и безопасности информации;

- некорректное использование ресурсов АС.

5. ***По степени зависимости от активности АС.***

1) *Угрозы, которые могут проявляться независимо от активности АС:*

- вскрытие шифров криптозащиты информации;

- хищение носителей информации (магнитных дисков, лент, микросхем памяти, запоминающих устройств и компьютерных систем).

2) *Угрозы, которые могут проявляться только в процессе автоматизированной обработки данных* (например, угрозы выполнения и распространения программных вирусов).

6. По степени воздействия на АС.

1) *Пассивные угрозы,* которые при реализации ничего не меняют в структуре и содержании АС (угроза копирования секретных данных).

2) *Активные угрозы,* которые при воздействии вносят изменения в структуру и содержание АС:

- внедрение аппаратных спецвложений, программных "закладок" и "вирусов" ("троянских коней" и "жучков"), т.е. таких участков программ, которые не нужны для выполнения заявленных функций, но позволяют преодолеть систему зашиты, скрытно и незаконно осуществить доступ к системным ресурсам с целью регистрации и передачи критической информации или дезорганизации функционирования системы:

- действия по дезорганизации функционирования системы (изменение режимов работы устройств или программ, забастовка, саботаж персонала, постановка мощных активных радиопомех на частотах работы устройств системы и т.п.);

- угроза умышленной модификации информации.

7. ***По пианам доступа пользователей или программ к ресурсам АС.***

1) *Угрозы, которые могут проявляться на этапе доступа к ресурсам АС* (например, угрозы несанкционированного доступа в АС).

2) *Угрозы, которые могут проявляться после разрешения доступа к ресурсам АС* (например, угрозы несанкционированного или некорректного использования ресурсов АС).

8. ***По способу доступа к ресурсам АС.***

1) *Угрозы, направленные на использование прямого стандартного пути доступа к ресурсам АС:*

- незаконное получение паролей и других реквизитов разграничения доступа (агентурным путем, используя халатность пользователей, подбором, имитацией интерфейса системы и т.д.) с последующей маскировкой под зарегистрированного пользователя ("маскарад");

- несанкционированное использование терминалов пользователей, имеющих уникальные физические характеристики, такие как номер рабочей станции в сети, физический адрес, адрес в системе связи, аппаратный блок кодирования и т.п.

2) *Угрозы, направленные на использование скрытого нестандартного пути доступа к ресурсам АС:*

- вход в систему в обход средств защиты (загрузка посторонней операционной системы со сменных магнитных носителей и т.п.);

- угроза несанкционированного доступа к ресурсам АС путем использования недокументированных возможностей ОС.

9. ***По текущему месту расположения информации, хранимой и обрабатываемой в АС.***

1) *Угрозы доступа к информации на внешних запоминающих устройствах* (например, угроза несанкционированного копирования секретной информации с жесткого диска).

2) *Угрозы доступа к информации в оперативной памяти:*

– чтение остаточной информации из оперативной памяти;

– чтение информации из областей оперативной памяти, используемых операционной системой (в том числе подсистемой зашиты) или другими пользователями, в асинхронном режиме, используя недостатки мультизадачных АС и систем программирования;

– угроза доступа к системной области оперативной памяти со сторон прикладных программ.

3) *Угрозы доступа к информации, циркулирующей в линиях связи:*

– незаконное подключение к линиям связи с целью работы во время пауз в действиях законного пользователя от его имени с вводом ложных сообщений или модификацией передаваемых сообщений;

– незаконное подключение к линиям связи с целью прямой подмены законного пользователя путем его физического отключения после входа в систему и успешной аутентификации с последующим вводом дезинформации и навязыванием ложных сообщений;

– перехват всего потока данных с целью дальнейшего анализа не в реальном масштабе времени.

4) *Угрозы доступа к информации, отображаемой на терминале или печатаемой на принтере* (например, угроза записи отображаемой информации на скрытую видеокамеру). Вне зависимости от конкретных видов угроз или их проблемно-ориентированной классификации АС удовлетворяет потребности эксплуатирующих ее лиц, если обеспечиваются следующие свойства информации систем ее обработки.

В качестве основного критерия будем использовать аспект ИБ. привлекая при необходимости остальные.

Угроза доступности (отказа служб) возникает всякий раз, когда в результате преднамеренных действий, предпринимаемых другим пользователем или злоумышленником, блокируется доступ к некоторому ресурсу вычислительной системы. Реально блокирование может быть постоянным – запрашиваемый ресурс никогда не будет получен, или оно может вызывать только задержку запрашиваемого ресурса, достаточно долгую для того, чтобы он стал бесполезным. В этих случаях говорят, что ресурс исчерпан.

Доступность информации – свойство системы (среды, средств и технологии обработки), в которой циркулирует информация, характеризующаяся способностью обеспечивать своевременный беспрепятственный доступ субъектов к интересующей их информации и готовность соответствующих автоматизированных служб к обслуживанию поступающих от субъектов запросов всегда, когда возникает в этом необходимость.

Самыми частыми и самыми опасными (с точки зрения размера ущерба) являются *непреднамеренные ошибки* штатных пользователей, операторов, системных администраторов и других лиц, обслуживающих ИС. Эти ошибки и являются собственно угрозами (неправильно введенные данные или ошибка в программе), иногда они создают уязвимые места, которые используют злоумышленники (по данным до 65 % потерь – от непреднамеренных ошибок).

Другие *угрозы доступности* классифицируем по *компонентам ИС,* на которые нацелены угрозы:

- отказ пользователей;
- внутренний отказ информационной системы;
- отказ поддерживающей инфраструктуры.

Обычно, применительно к *пользователям,* рассматриваются следующие угрозы: нежелание работать с информационной системой; невозможность работать с системой в силу отсутствия соответствующей подготовки; невозможность работать с системой в силу отсутствия технической поддержки (неполнота документации, недостаток справочной информации и т.п.).

Основными *источниками внутренних отказов* являются: отступление от установленных правил эксплуатации: выход системы из штатного режима эксплуатации в силу случайных или преднамеренных действий пользователей или обслуживающего персонала (превышение расчетного числа запросов, чрезмерный объем обрабатываемой информации и т.п.); ошибки при (пере) конфигурировании системы; отказы программного и аппаратного обеспечения; разрушение данных; разрушение или повреждение аппаратуры.

По *отношению к поддерживающей инфраструктуре* рекомендуется рассматривать следующие угрозы: нарушение работы (случайное или умышленное) систем связи, электропитания, водо- и/или теплоснабжения, кондиционирования: разрушение или повреждение помещений; невозможность или нежелание обслуживающего персонала или пользователей выполнять свои обязанности.

Приведем некоторые примеры угроз и программных атак на доступность.

В качестве средства вывода системы из штатного режима эксплуатации может использоваться *агрессивное потребление ресурсов* (обычно – полосы пропускания сетей, вычислительных возможностей процессоров или ОЗУ). По расположению источника угрозы такое *потребление* подразделяется на *локальное* и *удаленное.* При просчетах в конфигурации системы локальная программа способна практически монополизировать процессор и/или физическую память, сведя скорость выполнения других программ к нулю.

Удаленное потребление ресурсов в последнее время проявляется в особенно опасной форме – как скоординированные распределенные атаки, когда на сервер с множества разных адресов с максимальной скоростью направляются вполне легальные запросы на соединение и/или обслуживание.

Угроза нарушения целостности включает в себя любое умышленное изменение информации, хранящейся в вычислительной системе или передаваемой из одной системы в другую, в том числе н несанкционированное изменение информации при случайных ошибках программного или аппаратного обеспечения. Санкционированными изменениями являются те, которые сделаны уполномоченными лицами с обоснованной целью (например, периодическая запланированная коррекция некоторой базы данных).

Целостность информации – существование информации в неискаженном виде (неизменном по отношению к некоторому фиксированному ее состоянию). Обычно субъектов интересует обеспечение более широкого свойства – достоверности информации, которое складывается из адекватности (полноты и точности) отображения состояния предметной области и непосредственно целостности информации, т.е. ее неискаженности.

Существует различие между *статической* и *динамической целостностью.* С целью нарушения статической целостности злоумышленник может: ввести неверные данные, изменить данные.

Угрозой целостности является не только фальсификация или изменение данных, но и отказ от совершенных действий. Если нет средств обеспечить *"неотказуемость",* компьютерные данные не могут рассматриваться в качестве доказательства.

Потенциально уязвимы с точки зрения нарушения *целостности* не только *данные,* но и *программы.* Внедрение рассмотренного выше вредоносного ПО – пример подобного нарушения.

Угрозами динамической целостности являются нарушение атомарности транзакций, переупорядочение, кража, дублирование данных или внесение дополнительных сообщений (сетевых пакетов и т.п.). Соответствующие действия в сетевой среде называются активным прослушиванием.

Угроза нарушения конфиденциальности заключается в том, что информация становится известной тому, кто не располагает полномочиями доступа к ней. В связи с данной угрозой используется термин "утечка".

Конфиденциа1ьность информации – субъективно определяемая (приписываемая) характеристика (свойство) информации, указывающая на необходимость введения ограничений на круг субъектов, имеющих доспи к данной информации, и обеспечиваемая способностью системы (среды) сохранять указанную информацию в тайне от субъектов, не имеющих полномочии доступа к ней. Объективные предпосылки подобного ограничения доступности информации для одних субъектов заключены в необходимости защиты их законных интересов от других субъектов информационных отношений.

Конфиденциальную информацию можно разделить на предметную и служебную. Служебная информация (например, пароли пользователей) не относится к определенной предметной области, в информационной системе она играет техническую роль, но ее раскрытие особенно опасно, поскольку оно чревато получением несанкционированного доступа ко всей информации, в том числе предметной.

Многим субъектам приходится выступать в качестве пользователей не одной, а целого ряда систем (информационных сервисов). Если для доступа к таким системам используются многоразовые пароли или иная конфиденциальная информация, то наверняка эти данные будут храниться не только в голове, но и в записной книжке или на листках бумаги, которые пользователь часто оставляет на рабочем столе, а то и попросту теряет. II дело здесь не в неорганизованности людей, а в изначальной непригодности парольной схемы. Невозможно помнить много разных паролей: рекомендации по их регулярной смене только усугубляют положение, заставляя применять несложные схемы чередования или стараться свести дело к двум-трем легко запоминаемым паролям.

Описанный класс уязвимых мест можно назвать размещением конфиденциальных данных в среде, где им не обеспечена необходимая зашита. Угроза же состоит в том, что кто-то не откажется угнать секреты, которые сами просятся в руки. Помимо паролей, хранящихся в записных книжках пользователей, в этот класс попадает передача конфиденциальных данных в открытом виде (в разговоре, в письме, по сети), которая делает возможным *перехват данных.* Для атаки могут использоваться разные технические средства (подслушивание или прослушивание разговоров, пассивное прослушивание сети и т.п.), но идея одна – получить доступ к данным в тот момент, когда они наименее защищены.

Перехват данных – очень серьезная угроза, и если конфиденциальность действительно является критичной, а данные передаются по многим каналам, их зашита может оказаться весьма сложной и дорогостоящей. Технические средства перехвата хорошо проработаны, доступны, просты в эксплуатации, а установить их. например на кабельную сеть, может кто угодно, так что эту угрозу нужно принимать во внимание по отношению не только к внешним, но и к внутренним коммуникациям.

Опасной нетехнической угрозой конфиденциальности являются *методы морально-психологического воздействия,* такие как *маскарад* – выполнение действий под видом лица, обладающего полномочиями для доступа к данным

К неприятным угрозам, от которых трудно защищаться, можно отнести *злоупотребление полномочиями.* На многих типах систем привилегированный пользователь (например системный администратор) способен прочитать любой (незашифрованный) файл, получить доступ к почте любого пользователя и т.д. Другой пример – нанесение ущерба при сервисном обслуживании. Обычно сервисный инженер получает неограниченный доступ к оборудованию и имеет возможность действовать в обход программных защитных механизмов.

Таковы основные угрозы, которые наносят наибольший ущерб субъектам информационных отношений.

На современном этапе развития информационных технологий подсистемы или функции защиты являются неотъемлемой частью комплекса по обработке информации. Информация не представляется "в чистом виде", на пути к ней имеется хотя бы какая-нибудь система защиты, и поэтому чтобы угрожать, атакующая сторона должна преодолеть эту систему. Однако не существует абсолютно стойкой системы защиты, вопрос лишь во времени и средствах, требующихся на ее преодоление. Исходя из данных условий, примем следующую модель: *защита информационной системы считается преодоленной, если в ходе ее исследования определены все уязвимости системы.* Поскольку преодоление защиты также представляет собой угрозу, для защищенных систем будем рассматривать ее четвертый вид – *угрозу раскрытия параметров АС,* включающей в себя систему защиты. С точки зрения практики любое проводимое мероприятие предваряется этапом разведки, в ходе которого определяются основные параметры системы, ее характеристики, в результате чего уточняется поставленная задача и выбираются оптимальные технические средства.

Угрозу раскрытия можно рассматривать как опосредованную. Последствия ее реализации не причиняют какой-либо ущерб обрабатываемой информации, но дают возможность реализоваться первичным или непосредственным угрозам, перечисленным выше. Введение данного вида угроз позволяет описывать с научно-мето до логической точки зрения отличия защищенных информационных систем от открытых. Для последних угроза разведки параметров системы считается реализованной.

Методы обеспечения информационной безопасности

При рассмотрении вопросов зашиты АС целесообразно использовать четырехуровневую градацию доступа к хранимой, обрабатываемой и защищаемой АС информации, которая поможет систематизировать как возможные угрозы, так и меры по их нейтрализации и парированию, т.е. может систематизировать и обобщить весь спектр методов обеспечения зашиты, относящихся к информационной безопасности:

1) уровень носителей информации;
2) уровень средств взаимодействия с носителем;
3) уровень представления информации;
4) уровень содержания информации.

Данные уровни были введены исходя из того, что, во-первых, информация для удобства манипулирования чаше всего фиксируется на некотором материальном носителе, которым может быть бумага, дискета, CD-диск и т.п. Во-вторых, если способ представления информации таков, что она не может быть непосредственно воспринята человеком, возникает необходимость в преобразованиях информации в доступный для человека способ представления. Например, для чтения информации с дискеты необходим компьютер, оборудованный дисководом соответствующего типа. В-третьих, информация может быть охарактеризована способом своего представления. Язык жестов, язык символов и другие – все это способы представления информации. В-четвертых, человеку должен быть доступен смысл представленной информации (семантика).

Защита носителей информации должна обеспечивать парирование всех возможных угроз, направленных как на сами носители, так и на зафиксированную на них информацию, представленную в виде изменения состояний отдельных участков, блоков, полей носителя. Применительно к АС защита носителей информации в первую очередь подразумевает защиту машинных носителей. Вместе с тем, необходимо учитывать, что носителями информации являются также каналы связи, документальные материалы, получаемые в ходе эксплуатации АС, и т.п. Защита средств взаимодействия с носителем охватывает спектр методов защиты программно-аппаратных средств, входящих в состав АС, таких как средства вычислительной техники, операционная система, прикладные программы. В основном зашита на данном уровне рассматривается как защита от несанкционированного доступа, обеспечивающая разграничение доступа пользователей к ресурсам системы. Защита представления информации, т.е. некоторой последовательности символов, обеспечивается средствами криптографической зашиты. Защита содержания информации обеспечивается семантической зашитой данных.

Классификация злоумышленников

Возможности осуществления вредительских воздействий зависят от статуса злоумышленника по отношению к ИВС. Злоумышленником может быть:

• *разработчик ЛВС* (владеет наиболее полной информацией о программных и аппаратных средствах ИВС и имеет возможность внедрения "закладок" на этапах создания и модернизации систем, но не получает доступа на эксплуатируемые объекты ИВС):

• *сотрудник из числа обслуживающего персонала* (работники службы безопасности информации, системные и прикладные программисты, инженерно- технический персонал):

• *пользователь* (имеет общее представление о структуре ИВС и механизмах ее зашиты, но может осуществлять сбор информации методами традиционного шпионажа и попытками несанкционированного доступа);

• *постороннее лицо* (дистанционные методы шпионажа и диверсионная деятельность).

Основные направления и методы реализации угроз информационной безопасности

К *основным направлениям* реализации злоумышленником информационных угроз относятся:

• непосредственное обращение к объектам доступа;

• создание программных и технических средств, выполняющих обращение к объектам доступа в обход средств защиты;

• модификация средств зашиты, позволяющая реализовать угрозы ИБ:

• внедрение в технические средства АС" программных или технических механизмов, нарушающих предполагаемую структуру и функции АС.

Программно-технический уровень информационной безопасности

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

В связи с этим именно компьютер (особенно находящийся в составе сети) следует в первую очередь рассматривать в качестве объекта защиты, а конечного пользователя – в качестве ее наиболее вероятного потенциального нарушителя. Как следствие, под сомнение ставится обоснованность концепции реализованной системы защиты в современных универсальных ОС. Эта система защиты заключается в построении распределенной схемы администрирования механизмов зашиты, элементами которой, помимо администратора, выступают пользователи, имеющие возможность назначать и изменять права доступа к создаваемым ими файловым объектам.

На практике сегодня существует два подхода к обеспечению компьютерной безопасности:

1) использование только встроенных в ОС и приложения средств защиты;

2) применение, наряду со встроенными, дополнительных механизмов защиты. Этот подход заключается в использовании так называемых технических средств добавочной зашиты – программных, либо программно-аппаратных комплексов, устанавливаемых на защищаемые объекты.

Существующая статистика ошибок, обнаруженных в ОС, а также сведения о недостаточной эффективности встроенных в ОС и приложения механизмов зашиты, заставляет специалистов сомневаться в достижении гарантированной защиты от НС Д, при использовании встроенных механизмов, и все большее внимание уделять средствам добавочной зашиты информации.

Таким образом, важнейшим условием защищенности компьютерной информации является квалификация администраторов безопасности и сотрудников эксплуатирующих служб, которая, по крайней мере, не должна уступать квалификации злоумышленников, в противном случае не помогут никакие средства зашиты.

Центральным для программно-технического уровня является понятие *сервиса безопасности*

Далее рассмотрим следующие *основные* и *вспомогательные* сервисы:

• *идентификация* и *аутентификация.*
• *управление доступом:*
• *протоколирование* и *аудит:*
• *шифрование:*
• *контроль целостности:*
• *экранирование:*
• *анализ защищенности:*
• обеспечение *отказоустойчивости:*
• обеспечение *безопасного восстановления:*
• *туннелирование:*
• *управление.*

Совокупность перечисленных выше сервисов безопасности называют полным набором. Считается, что его, в принципе, достаточно для построения надежной зашиты на программно-техническом уровне, правда, при соблюдении целого ряда дополнительных условий (отсутствие уязвимых мест, безопасное администрирование и т.д.).

Для проведения классификации сервисов безопасности и определения их места в обшей архитектуре меры безопасности можно разделить на следующие виды:

- *превентивные,* препятствующие нарушениям ИБ;
- меры *обнаружения нарушений:*
- *локализующие,* сужающие зону воздействия нарушений;
- меры по *выявлению нарушителя:*
- меры *восстановления* режима безопасности.

Большинство сервисов безопасности попадает в число превентивных. Аудит и контроль целостности способны помочь в обнаружении нарушений: активный аудит, кроме того, позволяет запрограммировать реакцию на нарушение с целью локализации и/или прослеживания. Направленность сервисов отказоустойчивости и безопасного восстановления очевидна. Управление играет инфраструктурную роль, обслуживая все аспекты ИС.

Идентификацию и *аутентификацию* можно считать основой программно-технических средств безопасности, поскольку остальные сервисы рассчитаны на обслуживание именованных субъектов. Идентификация и аутентификация – это первая линия обороны, "проходная" информационного пространства организации.

Идентификация позволяет субъекту (пользователю, процессу, действующему от имени определенного пользователя, или иному аппаратно-программному компоненту) назвать себя (сообщить свое имя). Посредством аутентификации вторая сторона убеждается, что субъект действительно тот, за кого он себя выдает. В качестве синонима слова "аутентификация" иногда используют словосочетание "проверка подлинности".

Аутентификация бывает односторонней (обычно клиент доказывает свою подлинность серверу) и *двусторонней* (взаимной). Пример односторонней аутентификации – процедура входа пользователя в систему.

В сетевой среде, когда стороны идентификации/аутентификации территориально разнесены, у рассматриваемого сервиса есть два основных аспекта:

- что служит *аутентификатором* (т.е. используется для подтверждения подлинности субъекта):

- как организован (и защищен) обмен данными идентификации/аутентификации.

Субъект может подтвердить свою подлинность, предъявив, по крайней мере, одну из следующих сущностей:

- нечто, что он знает (пароль, личный идентификационный номер, криптографический ключ и т.п.);

- нечто, чем он владеет (личную карточку или иное устройство аналогичного назначения);

- нечто, что есть часть его самого (голос, отпечатки пальцев и т.п., т.е. свои биометрические характеристики).

В открытой сетевой среде между сторонами идентификации/аутентификации не существует доверенного маршрута; это значит, что в общем случае данные, переданные субъектом, могут не совпадать с данными, полученными и использованными для проверки подлинности. Необходимо обеспечить защиту от пассивного и активного прослушивания сети, т.е. от *перехвата, изменения* шили *воспроизведения* данных. Передача паролей в открытом виде, очевидно, неудовлетворительна; не спасает положение и шифрование паролей, так как оно не защищает от воспроизведения. Нужны более сложные протоколы аутентификации.

Надежная идентификация и аутентификация затруднена не только из-за сетевых угроз, но и по целому ряду причин. Во-первых, почти все аутентификационные сущности можно угнать, украсть или подделать. Во-вторых, имеется противоречие между надежностью аутентификации, с одной стороны, и удобствами пользователя и системного администратора – с другой. В-третьих, чем надежнее средство защиты, тем оно дороже.

Современные средства идентификации/аутентификации должны поддерживать концепцию единого входа в сеть. *Единый вход в сеть* – это, в первую очередь, требование удобства для пользователей. Если в корпоративной сети много информационных сервисов, допускающих независимое обращение, то многократная идентификация/аутентификация становится слишком обременительной. К сожалению, пока нельзя

сказать, что единый вход в сеть стал нормой, доминирующие решения пока не сформировались.

Парольная аутентификация – проста и уже давно встроена в операционные системы и иные сервисы. При правильном использовании пароли могут обеспечить приемлемый для многих организации уровень безопасности. Тем не менее, по совокупности характеристик их следует признать самым слабым средством проверки подлинности. Чтобы пароль был запоминающимся, его зачастую делают простым (имя подруги, название спортивной команды и т.п.). Иногда пароли с самого начала не хранятся в тайне, так как имеют стандартные значения, указанные в документации, и далеко не всегда после установки системы производится их смена.

Ввод пароля можно подсмотреть. Иногда для подглядывания используются даже оптические приборы. Пароли нередко сообщают коллегам, чтобы те могли, например, подменить на некоторое время владельца пароля. Пароль можно угадать "методом грубой силы", используя, скажем, словарь. Если файл паролей зашифрован, но доступен для чтения, его можно скачать к себе на компьютер и попытаться подобрать пароль, запрограммировав полный перебор (предполагается, что алгоритм шифрования известен).

Тем не менее, следующие меры позволяют значительно повысить надежность парольной защиты:

- *наложение технических ограничений* (пароль должен быть не слишком коротким, он должен содержать буквы, цифры, знаки пунктуации и т.п.);
- *управление сроком действия паролей,* их периодическая смена;
- ограничение доступа к файлу паролей;
- ограничение числа неудачных попыток входа в систему (это затруднит применение "метода грубой силы");
- обучение пользователей;
- использование программных *генераторов паролей* (программа, основываясь на несложных правилах, порождает только благозвучные и. следовательно, запоминающиеся пароли).

Рассмотренные выше *пароли можно назвать многоразовыми;* их раскрытие позволяет злоумышленнику действовать от имени легального пользователя. Гораздо более сильным средством, устойчивым к пассивному прослушиванию сети, являются *одноразовые пароли.*

Другой подход к надежной аутентификации состоит в генерации нового пароля через небольшой промежуток времени (например, каждые 60 секунд), для чего могут использоваться программы или специальные интеллектуальные карты. Серверу аутентификации должен быть известен алгоритм генерации паролей и ассоциированные с ним параметры; кроме того, часы клиента и сервера должны быть синхронизированы.

Биометрическая аутентификация представляет собой совокупность автоматизированных методов идентификации и/или аутентификации людей на основе их физиологических и поведенческих характеристик. К числу физиологических характеристик принадлежат особенности отпечатков пальцев, сетчатки и роговицы глаз, геометрия руки и лица и т.п. К поведенческим характеристикам относятся динамика подписи (ручной), стиль работы с клавиатурой. На стыке физиологии и поведения находятся анализ особенностей голоса и распознавание речи.

В общем виде работа с биометрическими данными организована следующим образом. Сначала создается и поддерживается база данных характеристик потенциальных пользователей. Для этого биометрические характеристики пользователя снимаются, обрабатываются, и результат обработки (называемый биометрическим шаблоном) за-

носится в базу данных (исходные данные, такие как результат сканирования пальца или роговицы, обычно не хранятся).

В дальнейшем для идентификации (и одновременно аутентификации) пользователя процесс снятия и обработки повторяется, после чего производится поиск в базе данных шаблонов. В случае успешного поиска личность пользователя и ее подлинность считаются установленными. Для аутентификации достаточно произвести сравнение с одним биометрическим шаблоном, выбранным на основе предварительно введенных данных.

На наш взгляд, к биометрии следует относиться весьма осторожно. Необходимо учитывать, что она подвержена тем же угрозам, что и другие методы аутентификации. Во-первых, биометрический шаблон сравнивается не с результатом первоначальной обработки характеристик пользователя, а с тем, что пришло к месту сравнения. А, как известно, за время пути... много чего может произойти. Во-вторых, биометрические методы не более надежны, чем база данных шаблонов. В-третьих, следует учитывать разницу между применением биометрии на контролируемой территории, под бдительным оком охраны, и в "полевых" условиях, когда, например к устройству сканирования роговицы могут поднести муляж и т.п. В-четвертых, биометрические данные человека меняются, так что база шаблонов нуждается в сопровождении, что создает определенные проблемы и для пользователей, и для администраторов.

Но главная опасность состоит в том, что любая "пробоина" для биометрии оказывается фатальной. Пароли, при всей их ненадежности, в крайнем случае можно сменить. Утерянную аутентификационную карту можно аннулировать и завести новую. Палец же, глаз или голос сменить нельзя. Если биометрические данные окажутся скомпрометированы, придется как минимум производить существенную модернизацию всей системы.

Протоколирование и аудит

Под *протоколированием* понимается сбор и накопление информации о событиях, происходящих в информационной системе. У каждого сервиса свой набор возможных событий, но в любом случае их можно разделить на внешние (вызванные действиями других сервисов), внутренние (вызванные действиями самого сервиса) и клиентские (вызванные действиями пользователей и администраторов).

Аудит – это анализ накопленной информации, проводимый оперативно, в реальном времени или периодически. Оперативный аудит с автоматическим реагированием на выявленные нештатные ситуации называется активным.

Реализация протоколирования и аудита решает следующие задачи:

1) обеспечение подотчетности пользователей и администраторов:

2) обеспечение возможности реконструкции последовательности событий;

3) обнаружение попыток нарушений информационной безопасности;

4) предоставление информации для выявления и анализа проблем.

Слишком обширное или подробное протоколирование не только снижает производительность сервисов, но и затрудняет аудит, т.е. не увеличивает, а уменьшает информационную безопасность.

Разумный подход к упомянутым вопросам применительно к операционным системам предлагается в "Оранжевой книге", где выделены следующие события: вход в систему (успешный или нет); выход из системы; обращение к удаленной системе; операции с файлами (открыть, закрыть, переименовать, удалить); смена привилегий или иных атрибутов безопасности (режима доступа, уровня благонадежности пользователя и т.п.).

При протоколировании события рекомендуется записывать, по крайней мере, следующую информацию: дата и время события; уникальный идентификатор пользователя – инициатора действия; тип события; результат действия (успех или неудача); источник запроса (например, имя терминала): имена затронутых объектов (например, открываемых или удаляемых файлов); описание изменений, внесенных в базы данных зашиты (например, новая метка безопасности объекта).

Характерная особенность протоколирования и аудита – зависимость от других средств безопасности. Идентификация и аутентификация служат отправной точкой подотчетности пользователей, логическое управление доступом защищает конфиденциальность и целостность регистрационной информации. Возможно, для зашиты привлекаются и криптографические методы.

Возвращаясь к целям протоколирования и аудита, отметим, что обеспечение подотчетности важно, в первую очередь, как сдерживающее средство. Если пользователи и администраторы знают, что все их действия фиксируются, они, возможно, воздержатся от незаконных операций. Очевидно, если есть основания подозревать какого-либо пользователя в нечестности, можно регистрировать все его действия, вплоть до каждого нажатия клавиши. При этом обеспечивается не только возможность расследования случаев нарушения режима безопасности, но и откат некорректных изменений (если в протоколе присутствуют данные до и после модификации). Тем самым защищается целостность информации.

Реконструкция последовательности событий позволяет выявить слабости в защите сервисов, найти виновника вторжения, оценить масштабы причиненного ущерба и вернуться к нормальной работе.

Выявление и анализ проблем могут помочь улучшить такой параметр безопасности, как доступность. Обнаружив узкие места, можно переконфигурировать или перенастроить систему, снова измерить производительность и т.д.

Непросто осуществить организацию согласованного протоколирования и аудита в распределенной разнородной системе. Во-первых, некоторые компоненты, важные для безопасности (например, маршрутизаторы), могут не обладать своими ресурсами протоколирования: в таком случае их нужно экранировать другими сервисами, которые возьмут протоколирование на себя. Во-вторых, необходимо увязывать между собой события в разных сервисах.

Активный аудит

Под подозрительной активностью понимается поведение пользователя или компонента информационной системы, являющееся злоумышленным (в соответствии с заранее определенной политикой безопасности) или нетипичным (согласно принятым критериям).

Задача активного аудита – оперативно выявлять подозрительную активность и предоставлять средства для автоматического реагирования на нее.

Активность, не соответствующую политике безопасности, целесообразно разделить на атаки, направленные на незаконное получение полномочий, и на действия, выполняемые в рамках имеющихся полномочий, но нарушающие политику безопасности.

Атаки нарушают любую осмысленную политику безопасности. Иными словами, активность атакующего является разрушительной независимо от политики. Следовательно, для описания и выявления атак можно применять универсальные методы, инвариантные относительно политики безопасности, такие как сигнатуры и их обнаружение во входном потоке событий с помощью аппарата экспертных систем.

Сигнатура атаки – это совокупность условий, при выполнении которых атака считается имеющей место, что вызывает заранее определенную реакцию. Простейший пример сигнатуры – "зафиксированы три последовательные неудачные попытки входа

в систему с одного терминала", пример ассоциированной реакции – блокирование терминала до прояснения ситуации.

Действия, выполняемые в рамках имеющихся полномочий, но нарушающие политику безопасности, мы будем называть злоупотреблением полномочиями. Злоупотребления полномочиями возможны из-за неадекватности средств разграничения доступа выбранной политике безопасности. Простейшим примером злоупотреблении является неэтичное поведение суперпользователя, просматривающего личные файлы других пользователей. Анализируя регистрационную информацию, можно обнаружить подобные события и сообщить о них администратору безопасности, хотя для этого необходимы соответствующие средства выражения политики безопасности.

Выделение злоупотреблений полномочиями в отдельную группу неправомерных действий, выявляемых средствами активного аудита, не является общепринятым, но подобный подход имеет право на существование.

Нетипичное поведение выявляется статистическими методами. В простейшем случае применяют систему порогов, превышение которых является подозрительным. В более развитых системах производится сопоставление долговременных характеристик работы (называемых долгосрочным профилем) с кратко срочными профилями. (Здесь можно усмотреть аналогию биометрической аутентификации по поведенческим характеристикам.)

Применительно к средствам активного аудита различают ошибки первого и второго рода: пропуск атак и ложные тревоги. Нежелательность ошибок первого рода очевидна: ошибки второго рода – отвлекают администратора безопасности от действительно важных дел, косвенно способствуя пропуску атак.

Достоинства сигнатурного метода – высокая производительность, малое число ошибок второго рода, обоснованность решений. Основной недостаток – неумение обнаруживать неизвестные атаки и вариации известных атак.

Основные достоинства статистического подхода – универсальность и обоснованность решений, потенциальная способность обнаруживать неизвестные атаки, т.е. минимизация числа ошибок первого рода. Минусы заключаются в относительно высокой доле ошибок второго рода, плохой работе в случае, когда неправомерное поведение является типичным, когда типичное поведение плавно меняется от легального к неправомерному, а также в случаях, когда типичного поведения нет (как показывает статистика, примерно 5...10 %).

Средства активного аудита могут располагаться на всех линиях обороны информационной системы. На границе контролируемой зоны они могут обнаруживать подо зрительную активность в точках подключения к внешним сетям (не только попытки нелегального проникновения, но и действия по "прощупыванию" сервисов безопасности). В корпоративной сети, в рамках информационных сервисов и сервисов безопасности, активный аудит в состоянии обнаружить и пресечь подозрительную активность внешних и внутренних пользователей, выявить проблемы в работе сервисов, вызванные как нарушениями безопасности, так и аппаратно-программными ошибками. Важно отметить, что активный аудит, в принципе, способен обеспечить защиту от атак на доступность.

Активный аудит развивается более десяти лет, и первые результаты казались весьма многообещающими. Довольно быстро удалось реализовать распознавание простых типовых атак, однако затем было выявлено множество проблем, связанных с обнаружением заранее неизвестных атак, атак распределенных, растянутых во времени и т.п. (Оперативное пополнение базы сигнатур атак таким решением, конечно, не является.) Тем не менее, и на нынешней стадии развития активный аудит полезен как один из рубежей (вернее, как набор прослоек) эшелонированной обороны.

Функциональные компоненты н архитектура

В составе средств активного аудита можно выделить следующие функциональные компоненты:

- компоненты генерации регистрационной информации. Они находятся на стыке между средствами активного аудита и контролируемыми объектами;
- компоненты хранения сгенерированной регистрационной информации;
- компоненты извлечения регистрационной информации (сенсоры).

Обычно различают сетевые и хостовые сенсоры, имея в виду под первыми выделенные компьютеры, сетевые карты которых установлены в режим прослушивания, а под вторыми – программы, читающие регистрационные журналы операционной системы. На наш взгляд, с развитием коммутационных технологий это различие постепенно стирается, так как сетевые сенсоры приходится устанавливать в активном сетевом оборудовании и, по сути, они становятся частью сетевой ОС;

- компоненты просмотра регистрационной информации. Могут помочь при принятии решения о реагировании на подозрительную активность;
- компоненты анализа информации, поступившей от сенсоров. В соответствии с данным выше определением средств активного аудита, выделяют пороговый анализатор, анализатор нарушений политики безопасности, экспертную систему, выявляющую сигнатуры атак, а также статистический анализатор, обнаруживающий нетипичное поведение;
- компоненты хранения информации, участвующей в анализе. Такое хранение необходимо, например, для выявления атак, протяженных во времени;
- компоненты принятия решений и реагирования ("решатели"). "Решатель" может получать информацию не только от локальных, но и от внешних анализаторов, проводя так называемый корреляционный анализ распределенных событий;
- компоненты хранения информации о контролируемых объектах. Здесь могут храниться пассивные данные и методы, необходимые, например, для извлечения из объекта регистрационной информации или для реагирования;
- компоненты, играющие роль организующей оболочки для менеджеров активного аудита, называемые мониторами и объединяющие анализаторы, "решатели", хранилище описаний объектов и интерфейсные компоненты. В число последних входят компоненты интерфейса с другими мониторами как равноправными, так и входящими в иерархию. Такие интерфейсы необходимы, например, для выявления распределенных, широкомасштабных атак;
- компоненты интерфейса с администратором безопасности.

Средства активного аудита строятся в архитектуре менеджер/агент. Основными агентскими компонентами являются сенсоры. Анализ, принятие решений – функции менеджеров. Очевидно, между менеджерами и агентами должны быть сформированы доверенные каналы.

Подчеркнем важность интерфейсных компонентов. Они полезны как с внутренней для средств активного аудита точки зрения (обеспечивают расширяемость, подключение компонентов различных производителей), так и с внешней точки зрения. Между менеджерами (между компонентами анализа и "решателями") могут существовать горизонтальные связи, необходимые для анализа распределенной активности. Возможно также формирование иерархий средств активного аудита с вынесением на верхние уровни информации о наиболее масштабной и опасной активности.

Обратим также внимание на архитектурную общность средств активного аудита и управления, являющуюся следствием общности выполняемых функций. Продуманные интерфейсные компоненты могут существенно облегчить совместную работу этих средств.[i]

Контрольные вопросы

1. Охарактеризуйте информацию и ее свойства.

2. Что является предметом и объектом зашиты информации?

3. Чем определяется ценность информации? Приведите классификацию конфиденциальной информации.

4. Охарактеризуйте свойства достоверности и своевременности информации.

5. Дайте определения информационной безопасности АСОИ и политики информационной безопасности.

---

[i] Составлено на основе материалов: Безбогов, А. А. Безопасность операционных систем : учебное пособие / А. А. Безбогов, А. В. Яковлев, Ю. Ф. Мартемьянов. – Москва: Машиностроение-1.