

Операционная система

**Тема 3. Назначение и
функции операционной
системы
часть 2**

Управление файлами и внешними устройствами

- Операционная система **виртуализирует** отдельный набор данных, хранящихся на внешнем накопителе, в виде **файла** – простой неструктурированной последовательности байтов, имеющих символическое имя.
- Для удобства работы файлы группируются в **каталоги**, а те в каталоги более высокого уровня.
- С помощью ОС пользователь или программист может выполнять над файлами и каталогами различные действия, например: создание, копирование, удаление, запись, чтение и т.д.

Управление файлами и внешними устройствами

- Файловая система тесно взаимодействует с **подсистемой ввода-вывода**, которая по запросам файловой системы осуществляет передачу данных между оперативной памятью и дисками.
- Подсистема ввода-вывода выполняет роль **интерфейса** ко всем устройствам подключенным к компьютеру.
- Управление устройствами осуществляется с помощью специальных программ называемых **драйверами** (для каждого устройства существует своя программа управления).
- Унифицированный интерфейс представляет обмен с любым внешним устройством как **обмен с файлом**, имеющим имя и являющийся неструктурированной последовательностью байтов.

Защита данных и администрирование

- Безопасность данных обеспечивается средствами **отказоустойчивости ОС**.
- Основные задачи:
 - защиту от сбоев и отказов аппаратуры;
 - защиту от ошибок программного обеспечения;
 - защиту от несанкционированного доступа (защита данных от ошибочного или злонамеренного поведения пользователей).

Основные задачи администрирования ОС. Административные меры защиты.

- Организация эффективной и надежной защиты операционной системы невозможна с помощью одних только программно-аппаратных средств. Эти средства обязательно должны, дополняться административными мерами защиты. Без постоянной квалифицированной поддержки со стороны администратора даже самая надежная программно-аппаратная защита оборачивается фикцией.

Основные административные меры защиты.

- 1. Постоянный *контроль корректности* функционирования операционной системы, особенно ее подсистемы защиты. Такой контроль наиболее удобно организовать, если ОС поддерживает *регистрацию событий*. В этом случае ОС автоматически регистрирует в специальном журнале наиболее важные события, произошедшие в процессе функционирования системы.

Основные административные меры защиты

- 2. Организация и поддержание адекватной **политики безопасности**.

Политика безопасности должна постоянно корректироваться, оперативно реагируя на изменения в конфигурации ОС, установку, удаление и изменение конфигурации прикладных программных продуктов и расширений операционной системы, попытки злоумышленников преодолеть защиту операционной системы и т.д.

3. **Инструктирование пользователей** ОС о необходимости соблюдения мер безопасности при работе с ОС и контроль за соблюдением этих мер.

- 4. Регулярное создание и обновление **резервных копий** программ и данных ОС. Постоянный контроль изменений в конфигурационных данных и политике безопасности операционной системы.

Основные принципы администрирования ОС.

- Непрерывность;
- Комплексность;
- Актуальность;
- Адекватность;
- Непротиворечивость. (разграничение доступа, настроек процессов);
- Формальный подход. Применение методик (инструкций, положений, приказов, РД и прочих рекомендательных документов) и четких концептуальных принципов при постановке задач администрирования и их реализации;
- Подконтрольность

Задачи и принципы управления безопасностью.

- Отдельные средства ИБ не обеспечивают эффективного функционирования и требуют объединения в единую и централизованно управляемую и постоянно действующую систему информационной безопасности.
- Система ИБ обычно должна решать следующие **задачи**:
 - ввод в систему списка имен пользователей и терминалов, допущенных к информации ИС;
 - подготовку и ввод в систему, запись паролей пользователей на носители;
 - ввод в систему назначенных полномочий пользователей и терминалов;
 - раздачу пользователям носителей с паролями и значений паролей, запоминаемых и вводимых пользователями вручную с клавиатуры;
 - сбор сигналов несовпадения паролей и нарушения полномочий пользователей;

Задачи и принципы управления безопасностью

- • установление времени, места и причины НСД;
- анализ ситуации, принятие адекватных мер и восстановление нормального функционирования ИС
- контроль конфигурации системы;
- сбор сигналов вскрытия аппаратуры и контроль ввода (вывода) аппаратуры в (из) ремонт (а) и на (из) профилактику(и);
- контроль журнала регистрации доступа к информации ИС и периодический вызов справок из него;
- взаимодействие со службой функционального контроля ИС;
- контроль функционирования системы защиты;
- подготовку ключей, контроль и обеспечение функционирования средств шифрования информации;
- контроль стирания и уничтожения остатков секретной информации на машинных и бумажных носителях;
- регистрацию, учет и разграничение доступа к носителям информации и ПО;
- ведение статистики и прогнозирование НСД.

Интерфейс прикладного программирования

- Возможности ОС доступны прикладному программисту в виде *набора функций*, называющегося **интерфейсом прикладного программирования** (Application Programming Interface - **API**). От обычного пользователя эти функции скрыты за оболочкой пользовательского интерфейса.
- API *определяет функциональность*, которую предоставляет программа (модуль, библиотека), при этом API позволяет *абстрагироваться* от того, как именно эта функциональность реализована.
- Для программистов все *особенности конкретной ОС* представлены *особенностями ее интерфейса прикладного программирования* (API). Поэтому ОС с различной внутренней организацией, но с *одинаковым набором функций API* представляются им **одной и той же** ОС, что значительно *упрощает стандартизацию* ОС и обеспечивает **переносимость** приложений между различными ОС, соответствующими определенному стандарту на API.

Интерфейс прикладного программирования

- Программные компоненты взаимодействуют друг с другом посредством API. При этом обычно компоненты образуют *иерархию* - высокоуровневые компоненты используют API низкоуровневых компонентов, а те, в свою очередь, используют API ещё более низкоуровневых компонентов.
- Практически все операционные системы (UNIX, Windows, Mac OS, и т. д.) имеют API, с помощью которого программисты могут создавать приложения для этой операционной системы. Главный API операционных систем — это множество о системных вызовов

Интерфейс прикладного программирования

- Приложения выполняют обращения к функциям API с помощью **системных вызовов**. Способ, которым приложения получают услуги ОС, очень похож на вызов подпрограмм.
- Информация, необходимая ОС для выполнения вызываемой функции, заранее помещается в определенное место памяти, в регистры процессора и/или стек. Затем управление передается ОС, которая выполняет необходимую функцию и возвращает результаты через память, регистры или стек. Каждая выполняемая функция, кроме значений результатов, возвращает код ошибки, по которому можно судить о корректности выполненной операции.

Наиболее известные API

Операционных систем

[Amiga ROM Kernel](#)
[Cocoa](#)

Linux Kernel API
OS/2 API

[POSIX](#)
[Windows API](#)

Графических интерфейсов

[Direct3D](#) (часть [DirectX](#))
[DirectDraw](#) (часть [DirectX](#))
[GDI](#)
GDI+
[GTK](#)

[Motif](#)
[OpenGL](#)
[OpenVG](#)
[Qt](#)
[SDL](#)

[Tk](#)
[WxWidgets](#)
[X11](#)
[Zune](#)

Звуковых интерфейсов

DirectMusic (часть [DirectX](#))

[DirectSound](#) (часть [DirectX](#))

[OpenAL](#)

Аутентификационных систем

[BioAPI](#)

[PAM](#)

Пользовательский интерфейс

- Современные ОС поддерживают развитые функции пользовательского интерфейса для интерактивной работы за терминалами двух типов: *алфавитно-цифровыми* и *графическими*.
- При работе за *алфавитно-цифровым* терминалом пользователь имеет в своем распоряжении *систему команд*, мощность которой отражает функциональные возможности данной ОС. Обычно командный язык ОС позволяет запускать и останавливать приложения, выполнять различные операции над файлами и каталогами, получать информацию о состоянии операционной системы, администрировать систему. Команды могут вводиться не только в интерактивном режиме с терминала, но и считываться из так называемого *командного файла*, содержащего некоторую последовательность команд.
- Программный модуль операционной системы, отвечающий за чтение отдельных команд или же последовательности команд из командного файла, обычно называется *командным интерпретатором*.

Пользовательский интерфейс

- Ввод команд может быть упрощен, если ОС поддерживает *графический* пользовательский интерфейс. Графический пользовательский интерфейс собой *окно*, в котором содержатся *различные элементы управления*. Взаимодействие пользователя с программой при помощи мыши и при помощи клавиатуры.
- Имеется возможность использовать кнопки и разделы меню, расположенные *внутри* самого *приложения*.
- В графическом интерфейсе пользователя распространены типичные элементы управления. Они позволяют *стандартизировать* процесс взаимодействия с различными программами в разных ОС.