

2 курс

ПЛАН – КОНСПЕКТ
проведения лекционного занятия по дисциплине
«Информатика»

**Раздел 3. «Программное обеспечение вычислительной
техники.»**

**Тема № 3.2: «Операционные системы и оболочки.
Стандартные программы.»**

Подготовил: преподаватель
В.Н. Борисов

Рязань 2024

**Лекционное занятие
по Теме № 3.2. «Операционные системы и оболочки. Стандартные программы.»**

Цель занятия: изучить со студентами основные сведения об операционных системах и оболочках, стандартных программах.

Вид занятия: классно-групповое, комбинированное (по проверке знаний, умений по пройденному материалу, по изучению и первичному закреплению нового материала).

Метод проведения занятия: доведение теоретических сведений.

Время проведения: 2 ч (90 мин.)

Основные вопросы:

1. Назначение и виды операционных систем, структура операционных систем.
2. Назначение и виды систем программирования, структура систем программирования.
3. Назначение и виды сервисных программ, структура сервисных программ.
4. Назначение и виды программ технического обслуживания, структура программ технического обслуживания.
5. Операции с файлами и папками в ОС семейства Windows (в том числе создание папок и ярлыков).

Литература:

1. [2 учебник раздела «Основной учебной литературы» рабочей программы изучения дисциплины]: Гаврилов, М. В. Информатика и информационные технологии : учебник для среднего профессионального образования / М. В. Гаврилов, В. А. Климов. — 5-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 355 с. — (Профессиональное образование). — ISBN 978-5-534-15930-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/viewer/informatika-i-informacionnyie-tehnologii-510331#page/1>, глава 4.

Примерный расчет времени:

1. Вступительная часть – 20 мин.
2. Основная часть – 60 мин.
3. Заключительная часть – 10 мин.

Вступительная часть:

Занятие начать с объявления темы занятия, основных рассматриваемых вопросов, времени изучения темы (нового материала), закрепления на практике полученных знаний, перечисления литературы.

Основная часть (теоретическая):

Первый вопрос: Назначение и виды операционных систем. Структура операционных систем.

Введение: Виды программ для ЭВМ (персональных компьютеров).

Все программы, которые выполняются на компьютере, можно разделить на две части – прикладные и системные. Компьютеры существуют в основном для того, чтобы выполнять прикладные программы, однако понятно, нас в первую очередь будут интересовать не прикладное, а именно системное программирование.

Все системные программы можно, тоже разделить на два класса. В один класс входят программы, предназначенные для управления оборудованием ЭВМ (и, для обеспечения эффективной эксплуатации этого оборудования), а также программы, управляющие на компьютере выполнением других программ. Кроме того, обычно сюда же включают и служебные программы для управления обрабатываемыми данными (файловую систему). Программы этого класса входят в большой комплекс системных программ, который называется операционной системой ЭВМ.

В другой класс входят системные программы, предназначенные для автоматизации процесса разработки, модификации и эксплуатации программ. Программы этого класса входят в состав системы программирования. Система программирования состоит только из таких системных программ, которые помогают писать новые программы. Система программирования является комплексом, в состав которого входят языковые, программные и информационные компоненты.

Назначение, виды, структура операционных систем.

Операционная система (ОС) – это комплекс взаимосвязанных системных программ для организации взаимодействия пользователя с компьютером и выполнения всех других программ.

Вычислительная система – взаимосвязанная совокупность аппаратных средств вычислительной техники и программного обеспечения, предназначенная для обработки информации) и удобства работы с ней.

ОС относятся к составу системного программного обеспечения и являются основной его частью.

Наиболее популярные операционные системы:

- MS DOS
- Microsoft Windows
- Mac OS

- OS/2
- UNIX
- Linux.

Основные функции ОС:

- *управление устройствами компьютера (ресурсами)*
- согласованная работа всех аппаратных средств ПК: стандартизованный доступ к периферийным устройствам, управление оперативной памятью и др.
- *управление процессами*
- выполнение программ и их взаимодействие с устройствами компьютера.
- *управление доступом к данным на энергонезависимых носителях*
(таких как жесткий диск, компакт-диск и т.д.), как правило, с помощью файловой системы.
- *ведение файловой структуры*
- создание, изменение, удаление, хранение файлов на носителях
- *пользовательский интерфейс*
- диалог с пользователем.

Дополнительные функции:

- параллельное или псевдопараллельное выполнение задач (многозадачность).
- взаимодействие между процессами: обмен данными, взаимная синхронизация.
- защита самой системы, а также пользовательских данных и программ от злонамеренных действий пользователей или приложений.
- разграничение прав доступа и многопользовательский режим работы (аутентификация, авторизация).

Главные цели операционной системы:

1. Эффективное использование всех компьютерных ресурсов.
2. Повышение производительности труда программистов.
3. Простота, гибкость, эффективность и надежность организации вычислительного процесса.
4. Обеспечение независимости прикладных программ от аппаратного обеспечения (АО).

Функцией ОС является распределение процессоров, памяти, устройств и данных между процессами, конкурирующими за эти ресурсы. ОС должна управлять всеми ресурсами вычислительной машины таким образом, чтобы обеспечить максимальную эффективность ее функционирования. Критерием эффективности может быть, например, пропускная способность или реактивность системы.

Таким образом, ОС реализует:

- интерфейс пользователя (команды в MS DOS, UNIX; графический интерфейс в ОС Windows);
- разделение аппаратных ресурсов между пользователями (в многопользовательской и многозадачной ОС);
- работу в локальных и глобальных сетях;
- возможность работы с общими данными в режиме коллективного пользования;
- планирование доступа пользователей к общим ресурсам;
- эффективное выполнение операций ввода-вывода;
- восстановление данных и вычислительного процесса в случае ошибок.

Для реализации управления ресурсами разные ОС используют различные алгоритмы, что, в конечном счете, и определяет их облик в целом, включая характеристики производительности, область применения и даже пользовательский интерфейс. Так, например, алгоритм управления процессором в значительной степени определяет, является ли ОС системой разделения времени, системой пакетной обработки или системой реального времени.

Состав операционной системы.

Современные операционные системы имеют сложную структуру, каждый элемент которой выполняет определенные функции по управлению компьютером.

- **Управление файловой системой.** Процесс работы компьютера сводится к обмену файлами между устройствами. В операционной системе имеются *программные модули, управляющие файловой системой.*
- **Командный процессор.** Специальная программа, которая запрашивает у пользователя команды и выполняет их.
- **Драйверы устройств.** Специальные программы, которые обеспечивают управление работой устройств и согласование информационного обмена с другими устройствами, а также позволяют производить настройку некоторых параметров устройств. Технология «*Plug ad Play*» (подключай и играй) позволяет автоматизировать подключение к компьютеру новых устройств и обеспечивает их конфигурирование.
- **Графический интерфейс.** Используется для упрощения работы пользователя.
- **Сервисные программы или утилиты.** Программы, позволяющие обслуживать диски (проверять, сжимать, дефрагментировать и т.д.), выполнять операции с файлами (архивировать и т.д.), работать в компьютерных сетях и т.д.
- **Справочная система.** Позволяет оперативно получить информацию как о функционировании операционной системы в целом, так и о работе ее отдельных модулей.

Наиболее общим подходом к структуризации **операционной системы** является разделение всех ее модулей на две группы:

- **Ядро** – это модули, выполняющие основные функции ОС.
- **Вспомогательные модули**, выполняющие вспомогательные функции ОС. Одним из определяющих свойств ядра является работа в привилегированном режиме.

Функции ОС:

- Планирование заданий.
- Использование процессора.
- Обеспечение программ средствами коммуникации и синхронизации.
- Управление памятью.
- Управление файловой системой.
- Управление вводом выводом.
- Обеспечение безопасности.

Виды интерфейсов пользователя операционных систем.

По типу пользовательского интерфейса различают :

- **текстовые** (линейные) операционные системы
- **графические** операционные системы
- **речевые** операционные системы

Пользовательским интерфейсом называется набор приемов взаимодействия пользователя с приложением. Пользовательский интерфейс включает общение пользователя с приложением и язык общения.

- **Текстовые или линейные операционные системы реализуют интерфейс командной строки.** Основным устройством управления в них является клавиатура. Команда набирается на клавиатуре и отображается на экране дисплея. Окончанием ввода команды служит нажатие клавиши Enter. Для работы с операционными системами, имеющими текстовый интерфейс, необходимо овладеть командным языком данной среды, т.е. совокупностью команд, структура которых определяется синтаксисом этого языка. Первые настоящие операционные системы имели текстовый интерфейс. В настоящее время он также используется на серверах и компьютерах пользователей.
- **Графические операционные системы реализуют интерфейс, основанный на взаимодействии активных и пассивных графических экранных элементов управления.** Устройствами управления в данном случае являются клавиатура и мышь. Активным элементом управления является указатель мыши — графический объект, перемещение которого на экране синхронизировано с перемещением мыши. Пассивные элементы управления — это графические элементы управления приложений (экранные кнопки, значки, переключатели, флажки, раскрывающиеся списки, строки меню и т.д.). Примером исключительно графических ОС

являются операционные системы семейства Windows. Стартовый экран подобных ОС представляет собой системный объект, называемый *рабочим столом*.

Рабочий стол — это графическая среда, на которой отображаются объекты (файлы и каталоги) и элементы управления.

В графических операционных системах большинство операций можно выполнять многими различными способами, например через строку меню, через панель инструментов, через систему окон и др. Поскольку операции выполняются над объектом, предварительно он должен быть выбран (выделен). Основу графического интерфейса пользователя составляет организованная *система окон* и других графических объектов, при создании которой разработчики стремятся к максимальной стандартизации всех элементов и приемов работы.

Окно — это обрамленная прямоугольная область на экране монитора, в которой отображаются приложения, документ, сообщение. Окно является активным, если с ним в данный момент работает пользователь.

Все операции, выполняемые в графических ОС, происходят либо на Рабочем столе, либо в каком-либо окне.

- **Речевые ОС, в случае SILK-интерфейса** (от англ. speech – речь, image – образ, language – язык, knowledge – знание) – на экране по речевой команде происходит перемещение от одних поисковых образов к другим. Предполагается, что при использовании общественного интерфейса не нужно будет разбираться в меню. Экранные образы однозначно укажут дальнейший путь перемещения от одних поисковых образов к другим по смысловым семантическим связям.

Второй вопрос: Назначение, виды, структура систем программирования.

Система программирования — это система для разработки новых программ на конкретном языке программирования.

Система программирования обычно включает в себя следующие **компоненты**:

1. Компилятор или интерпретатор.
2. Интегрированная среда разработки.
3. Средства создания и редактирования текстов программ.
4. Библиотеки стандартных программ и функций.
5. Отладочные программы, помогающие находить и устранять ошибки.
6. Диалоговая среда.
7. Многооконный режим работы.
8. Мощные графические библиотеки.
9. Утилиты для работы с библиотеками.

10. Ассемблер.
11. Справочная служба.

Компилятор — это особый вид транслятора, который переводит тексты с языка программирования высокого уровня (с того языка, которым пользуется программист при написании текста программы) на машинный язык (в машинный код, который понятен компьютеру).

Интерпретатор — это исполняемый файл, который поэтапно читает программу, а затем обрабатывает, сразу выполняя ее инструкции. Он осуществляет программу поэтапно как часть собственного исполняемого файла.

Интегрированная среда разработки — это набор инструментов для разработки и отладки программ, имеющий общую интерактивную графическую оболочку, поддерживающую выполнение всех основных функций жизненного цикла разработки программы.

Библиотеки стандартных программ и функций состоят из совокупности подпрограмм, составленных на одном из языков программирования и удовлетворяющих определенным единым требованиям к структуре, организации их входов и выходов, описаниям подпрограмм.

Важным компонентом понятия системы программирования являются отладочные программы.

Программный модуль отладки позволяет выполнить основные задачи, связанные с мониторингом процесса выполнения результирующей прикладной программы. Отладка позволяет последовательно и пошагово выполнять итоговые программы, просматривать значения объявленных переменных, устанавливать контрольные точки, трассировку для того, чтобы идентифицировать места и виды ошибок в разработке.

Справочная система, входящая в состав системы программирования, предназначена для предоставления пользователю справочной информации по конкретной системе программирования.

Машинно-ориентированные системы — это системы, в которых язык программирования, наборы операторов и изобразительные средства существенно зависят от особенностей архитектуры компьютера.

Машинно-независимые системы программирования — системы, позволяющие описывать алгоритмы решения задач и информацию, подлежащую обработке. Системы часто используются в широких кругах пользователей и не требуют особых знаний организации функционирования ЭВМ.

Виды языков программирования в машинно-независимых системах:

- процедурно-ориентированные;
- проблемно-ориентированные языки;

- объектно-ориентированное программирование. Процедурно-ориентированные являются основными языками описания алгоритмов, которые обеспечивают математические функции многих современных вычислительных машин.

Они включают в себя такие популярные языки как:

1. **Fortran** — один из старейших языков программирования высокого уровня, который используется для приложений с интенсивными вычислениями. Fortran часто применяется в процессе научного и инженерного вычисления. Он удобен благодаря большой программной базе, возможностью работы с документами и библиотекам с открытым исходным кодом, доступных под свободными лицензиями. Язык может осуществлять интуитивную запись в виде массива, которая упрощает запись быстрых векторизованных вычислений.
2. **Бейсик** является одним из самых простых языков программирования. Он был создан с целью обучения студентов основам решения задач с помощью написания кода, поэтому программа ориентировалась на пользователей, для которых скорость выполнения программ была не очень важна, и которым первоначально необходима возможность использовать компьютер для решения своих задач, не имея специальной подготовки. В России сегодня наиболее популярна разновидность **Turbo-Basic** фирмы Borland.
3. Язык программирования **Си** был создан как язык высокого уровня для разработки операционной системы **UNIX** и стал популярен благодаря своей простоте и эффективности. Си существенно повлиял на развитие индустрии программного обеспечения; его синтаксис стал основой для современных и востребованных языков **C++**, **C#**, **Java**.
4. **Паскаль** — язык высокого уровня общего назначения, который был первоначально разработан Никлаусом Виртом в начале 1970-х годов. Он разрабатывался с целью создания платформы для **обучения** программированию, поэтому Паскаль недостаточно удобен при решении сложных задач и широко используется для математических операций. Разновидность языка **Паскаль АВС** стала полноценной системой для начинающих, которая сегодня используется для обучения студентов и школьников.

Проблемно-ориентированные языки — это формальные языки, предназначенные для описания данных (информации) и алгоритмов их обработки (программ) на вычислительной машине.

Основные проблемно-ориентированные языки:

1. **ЛИСП** — семейство языков программирования, программы и данные в которых представляются системами линейных списков символов. Так как исходный код состоит из списков, программы на ЛИСПе позволяют его изменять как структуру данных и создавать макросистемы, позволяющие программистам формировать новый синтаксис или новые предметно-

ориентированные языки, встроенные в ЛИСП. В настоящее время ЛИСП применяется в экспертных системах, системах аналитических вычислений и т.д.

2. **Prolog** — язык логического программирования, который обеспечивает решение задач, выраженных в терминах объектов и отношений между ними. Для того чтобы инициировать вычисления, выполняется специальный запрос к базе знаний, на которые система логического программирования генерирует ответы «истина» и «ложь».

Объектно-ориентированное программирование основано на методологии представления программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.

Примеры объектно-ориентированных языков:

1. **JavaScript** — язык сценариев, который позволяет создавать интерактивные html-документы, производить вычисления, выполнять проверку допустимости данных без обращения к серверу. Скрипты программы позволяют взаимодействовать с сайтами: заполнять формы обратной связи, оставлять комментарии, просматривать всплывающие подсказки и т.д.
2. **Objective-C** — один из языков программирования, который активно используется для разработки мобильных приложений. Он используется корпорацией Apple и необходим для операционных систем OS X и iOS, их программных интерфейсов.
3. **Python** — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Он универсален, поэтому подходит для решения разнообразных задач и многих платформ, начиная с iOS и Android и заканчивая серверными ОС. Python используется в различных сферах IT, таких как машинное обучение, разработка приложений, web, парсинг и другие.
4. **Perl** — это язык программирования общего назначения, который изначально был разработан для работы с текстовой информацией, но в дальнейшем стал способен решать широкий круг задач, например, системное администрирование, веб-разработка, сетевое программирование, разработка графического интерфейса пользователя и т.д.

Актуальные системы программирования.

1. **Eclipse** — свободная интегрированная среда разработки модульных кроссплатформенных приложений, которая часто используется как платформа для разработки расширений. Eclipse является платформой с особым фундаментом для построения и запуска интегрированных инструментов разработки сквозного программного обеспечения. В силу

бесплатности и высокого качества, Eclipse во многих организациях является корпоративным стандартом для разработки приложений.

2. **IntelliJ IDEA** — мощная универсальная среда программирования, поддерживающая язык **Java**. Она позиционирует себя как умная и удобная система программирования для Java (и других языков) с поддержкой всех последних технологий и фреймворков. У этой среды разработки есть мощные аналитические возможности. Система включает в себя набор инструментов для изменения внутренней структуры программы, который позволяет быстро реорганизовывать исходные тексты. Дизайн среды ориентирован на продуктивность работы программистов и позволяет оптимизировать простые рутинные задачи, чтобы дать возможность специалистам сконцентрироваться на достижение функциональных целей.
3. **Delphi** — среда разработки прикладных программ, предназначенных для запуска в ОС Windows, MacOS, а также в мобильных операционных системах — iOS и Android. Delphi отличается простотой и может использоваться в учебных целях. Программы в Delphi пишутся на языке **Object Pascal**, который является преемником и развитием языка **Turbo Pascal**. Программа предназначена, в первую очередь, для разработки приложений в архитектуре клиент-сервер.
4. Система **Borland C++ Builder** – это мощная среда программирования, которая поддерживает принципы визуального объектно-ориентированного программирования для 32-разрядных операционных систем **Microsoft Windows** и позволяет значительно сократить время на разработку приложений.
5. **Symantec Cafe** — первая интегрированная среда визуальной разработки для создания приложений и интернет-страниц. Symantec Cafe позволяет разрабатывать приложения на языке Java, которые могут затем встраиваться в интернет-страницы для повышения их функциональности.

Третий вопрос: Назначение, виды, структура сервисных программ.

Пользовательский интерфейс (сервисные программы) – это программные надстройки операционной системы (оболочки и среды), предназначенные для упрощения общения пользователя с операционной системой.

Сервисные программы предназначены для выполнения различных вспомогательных операций – проверки исправности оборудования, архивации файлов, борьбы с вирусами, форматирования дисков (подготовки новых дисков к работе путем разметки на них дорожек и секторов).

Программы, обеспечивающие интерфейс, сохраняют форму общения (диалог) пользователя с операционной системой, но изменяют язык общения (обычно язык команд преобразуется в язык меню). Сервисные системы условно можно разделить на интерфейсные системы, оболочки операционных систем и утилиты.

Интерфейсные системы – это мощные сервисные системы, чаще всего графического типа, совершенствующие не только пользовательский, но и программный интерфейс операционных систем, в частности, реализующие некоторые дополнительные процедуры разделения дополнительных ресурсов.

Оболочки операционных систем предоставляют пользователю качественно новый по сравнению с реализуемой операционной системой интерфейс и делают необязательным знание последнего.

Утилиты автоматизируют выполнение отдельных типовых, часто используемых процедур, реализация которых потребовала бы от пользователя разработки специальных программ. Многие утилиты имеют развитый диалоговый интерфейс с пользователем и приближаются по уровню общения к оболочкам.

Утилиты предназначены для расширения возможностей операционной системы и встроенных в систему служебных программ за счет введения новых или усовершенствования уже существующих функций.

Так компьютерные утилиты можно разделить на три группы:

- 1) утилиты сервисного обслуживания компьютера,
- 2) утилиты расширения функциональности,
- 3) информационные утилиты.

Многие утилиты дублируют друг друга. Их перечень очень широк, но по функциональному назначению их разделяют на несколько групп:

- 1) антивирусные программы;
- 2) файловые менеджеры;
- 3) программы резервирования;
- 4) программы-упаковщики (архиваторы);
- 5) программы контроля и диагностики компьютера;
- 6) программы обслуживания дисков;
- 7) программы – КЭШи и программы оптимизации дисков;
- 8) программы обслуживания операционной системы....

Четвёртый вопрос: Назначение, виды, структура программ технического обслуживания.

Под программами технического обслуживания понимается совокупность программно-аппаратных средств для диагностики и обнаружения ошибок в процессе работы компьютера или вычислительной системы в целом.

Они включают в себя:

– средства диагностики и тестового контроля правильности работы ЭВМ и ее отдельных частей, в том числе автоматического поиска ошибок и неисправностей с определенной локализацией их в ЭВМ;

– специальные программы диагностики и контроля вычислительной среды информационной системы в целом, в том числе программно-аппаратный контроль, осуществляющий автоматическую проверку работоспособности системы обработки данных перед началом работы вычислительной системы в очередную производственную смену.

Комплекс программ технического обслуживания

Особенности состава технических средств ЭВМ учитываются комплексом программ технического обслуживания (КПТО). Этот комплекс включает в свой состав наладочные, проверочные и диагностические тест-программы.

Наладочные программы обеспечивают автономную настройку и проверку отдельных устройств ЭВМ. Обычно они функционально независимы от программ ОС. Проверочные тест-программы предназначены для периодически проводимых проверок правильности функционирования устройств, например после включения их в работу. Диагностические программы используются в тех случаях, когда необходимо классифицировать отказ оборудования и локализовать место неисправности. Инициирование работы этих программ осуществляется обычно модулями ОС после фиксации сбоев и отказов аппаратуры контроля.

Проверочные тест – программы занимают особое место в КПТО. Их выполнение непосредственно перед вычислениями позволяет убедиться в исправности технических средств системы, а значит, повысить достоверность результатов обработки данных.

Кроме встроенных средств контроля, в ПО ПЭВМ включаются и автономные средства контроля и диагностики. Количество подобных комплектов программ достаточно велико, и каждый из них позволяет детализировать системную информацию: определение полной конфигурации ПК и характеристик отдельных ее частей (тип процессора, наличие сопроцессора, тип материнской платы, типы используемых дисков, объем оперативной памяти и ее распределение, подключение дополнительной периферии и т.д.).

Пятый вопрос: Операции с файлами и папками в операционных системах (ОС) семейства Windows (в том числе создание папок и ярлыков).

Навигация по файловой системе. Операции с файлами и папками. Создание папок и ярлыков.

К основным операциям с файловой структурой относятся:

- навигация по файловой структуре;
- запуск программ и открытие документов;
- создание папок;
- копирование файлов и папок;

перемещение файлов и папок;
удаление файлов и папок;
переименование файлов и папок;
создание ярлыков.

Все операции с файлами и папками в Windows можно выполнять несколькими различными способами. Каждый выбирает себе те приемы, которые ему кажутся наиболее удобными. Обычно с приобретением опыта работы на компьютере совокупность используемых приемов меняется.

Простейшие приемы работы с файловой структурой предоставляет иерархическая система окон папок, берущая свое начало от известной нам папки **Мой компьютер**. Диски, представленные в окне этой папки, можно открыть, а потом разыскать на них любые нужные папки и файлы. Копирование и перемещение файлов и папок из одной папки в другую можно выполнять путем перетаскивания их значков из окна одной папки в окно другой. Для удаления объектов можно использовать перетаскивание на значок Корзины, а можно пользоваться контекстным меню, которое открывается при щелчке правой кнопкой мыши на объекте. Для создания в папке ярлыка документа или программы можно использовать специальное перетаскивание или команду Создать > Ярлык из контекстного меню.

Работа с файловой системой в окнах папок не вполне удобна, но для этой цели есть и более мощное средство -- программа **Проводник**.

Проводник -- служебная программа, относящаяся к категории диспетчеров файлов. Она предназначена для навигации по файловой структуре компьютера и ее обслуживания. Проводник очень глубоко интегрирован в операционную систему Windows. По сути, мы работаем с ним даже тогда, когда его не видим. Если по щелчку правой кнопкой мыши на каком-либо объекте мы получаем контекстное меню, это результат невидимой работы Проводника. Если при перетаскивании объектов из одного окна в другое происходит их копирование или перемещение, это тоже результат заочной деятельности Проводника. Однако с ним можно работать и «очно». Программа запускается командой Пуск > Программы > Стандартные > Проводник.

Окно программы Проводник очень похоже на окна папок. Основное отличие в том, что окно Проводника имеет не одну рабочую область, а две: левую панель, называемую панелью папок, и правую панель, называемую панелью содержимого.

Цель навигации состоит в обеспечении доступа к нужной папке и ее содержимому. Мы специально не говорим о том, что цель навигации -- это поиск нужных файлов и папок, поскольку для этой операции есть специальные средства.

Навигацию по файловой структуре выполняют на левой панели Проводника, на которой показана структура папок. Папки могут быть развернуты или свернуты, а также раскрыты или закрыты. Если папка имеет вложенные папки, то на левой панели рядом с папкой отображается узел, отмеченный знаком «+». Щелчок на узле разворачивает папку, при этом значок узла меняется на «-». Таким же образом папки и сворачиваются.

Для того чтобы раскрыть папку, надо щелкнуть на ее значке. Содержимое раскрытой папки отображается на правой панели. Одна из папок на левой панели раскрыта всегда. Закрыть папку щелчком на ее значке невозможно -- она закроется автоматически при раскрытии любой другой папки.

Задачи навигации по файловой структуре могут быть следующие:

Запуск программ и открытие документов

Эта операция выполняется двойным щелчком на значке программы или документа на правой панели Проводника. Если нужный объект на правой панели не показан, надо выполнить навигацию на левой панели и найти папку, в которой она находится.

Создание папок

Чтобы создать новую папку, сначала следует на левой панели Проводника раскрыть папку, внутри которой она будет создана. После этого надо перейти на правую панель, щелкнуть правой кнопкой мыши на свободном от значков месте и выбрать в контекстном меню пункт Создать > Папку. На правой панели появится значок папки с названием Новая папка. Название выделено, и в таком состоянии его можно редактировать. После того как папка будет создана, она войдет в состав файловой структуры, отображаемой на левой панели.

Копирование и перемещение файлов и папок

Папку, из которой происходит копирование, называют источником. Папку, в которую происходит копирование, называют приемником. Копирование выполняют методом перетаскивания значка объекта с правой панели Проводника на левую.

Первая задача -- найти и раскрыть папку-источник, чтобы на правой панели был виден копируемый объект. Вторая задача -- найти на левой панели папку-приемник, но раскрывать ее не надо. Далее объект перетаскивают с правой панели на левую и помещают на значок папки-приемника. Эта операция требует аккуратности, поскольку попасть одним значком точно на другой не всегда просто. Для контроля точности попадания надо следить за названием папки-приемника. В тот момент когда наведение выполнено правильно, подпись под значком меняет цвет, и кнопку мыши можно отпускать.

Если и папка-источник, и папка-приемник принадлежат одному диску, то при перетаскивании выполняется перемещение, а если разным -- то копирование. В тех случаях, когда нужно обратное действие, выполняют специальное перетаскивание при нажатой правой кнопке мыши.

Удаление файлов и папок.

На левой панели открывают папку, содержащую удаляемый объект, а на правой панели выделяют нужный объект (или группу объектов).

Удаление можно выполнять несколькими способами. Классический способ -- с помощью команды Файл > Удалить из строки меню (если ни один объект не выделен, эта команда не активируется). Более удобный способ -- использовать командную кнопку на панели инструментов. Еще более удобно воспользоваться контекстным меню. Щелкните правой кнопкой мыши на удаляемом объекте и выберите в контекстном меню пункт Удалить. Однако самый удобный способ удаления выделенного объекта состоит в использовании клавиши DELETE клавиатуры.

Использование манипуляторов, таких, как мышь, -- это важное достоинство графических операционных систем. Однако профессионалами давно отмечено, что наивысшая производительность труда и минимальное утомление при работе достигаются при максимальном использовании клавиатуры. Для команд, представленных в строке меню, часто приводятся клавиатурные комбинации, которыми эти команды можно выполнить. Обращайте на них внимание, запоминайте их и старайтесь постепенно переходить к их использованию. Это один из приемов закрепления навыков профессиональной работы с компьютером.

Создание ярлыков объектов

Ярлыки объектов можно создавать двумя способами: методом специального перетаскивания (вручную) или с помощью специальной программы-мастера (автоматически). С приемом специального перетаскивания уже знакомы. Объект выбирается на правой панели Проводника и перетаскивается при нажатой правой кнопке мыши на значок нужной папки на левой панели. В момент отпускания кнопки на экране появляется меню, в котором надо выбрать пункт Создать ярлык.

Второй способ (с использованием мастера) менее нагляден, но во многих случаях более удобен. Мастерами в системе Windows называют специальные программы, работающие в режиме диалога с пользователем. Диалог строится по принципу «запрос -- ответ». Если на все запросы от программы даны корректные ответы, программа автоматически выполнит черновую работу.

Приемы повышения эффективности в работе с файловой структурой.

Приемы, которые здесь описаны, являются общесистемными. Они относятся не только к Проводнику, но и ко всем окнам папок и большинству окон приложений.

Использование буфера обмена для работы с объектами. Система Windows создает и обслуживает на компьютере невидимую для пользователя область памяти, называемую буфером обмена. Этой областью можно и нужно уметь пользоваться. В любой момент времени в ней можно хранить только один объект.

Принцип работы с буфером обмена очень прост:

1. Открываем папку-источник. Выделяем щелчком нужный объект.
2. Копируем или забираем объект в буфер. В первом случае объект остается в папке источнике и может быть размножен. Во втором случае он удаляется из папки источника, но может некоторое время храниться в буфере. Последняя операция называется также вырезанием объекта.
3. Открываем папку-приемник и помещаем в нее объект из буфера обмена.

Три указанные операции (Копировать, Вырезать и Вставить) можно выполнять разными способами. Классический прием состоит в использовании пункта Правка в строке меню, но более удобно пользоваться командными кнопками панели инструментов:

Копировать;

Вырезать;

Вставить.

Самый же эффективный способ работы с буфером обмена состоит в использовании комбинаций клавиш клавиатуры:

CTRL+C -- копировать в буфер;

CTRL+X -- вырезать в буфер;

CTRL+V -- вставить из буфера.

Эти приемы работают во всех приложениях Windows, и их стоит запомнить. Через буфер обмена можно переносить фрагменты текстов из одного документа в другой, можно переносить иллюстрации, звукозаписи, видеофрагменты, файлы, папки и вообще любые объекты. Буфер обмена -- мощное средство для работы с приложениями и документами в Windows.

В буфере обмена всегда может находиться только один объект. При попытке поместить туда другой объект, предыдущий объект перестает существовать. Поэтому буфер обмена не используют для длительного хранения чего-либо. Поместив объект в буфер, немедленно выполняют вставку из буфера в нужное место.

В общем случае буфер обмена невидим для пользователя, и обычно необходимость просмотра его содержимого не возникает. Однако, если она все-таки возникнет, можно воспользоваться специальной служебной программой Папка обмена, которая входит в состав операционной системы и запускается командой Пуск > Программы > Стандартные > Служебные > Буфер обмена. Если на каком-то конкретном компьютере этой программы нет, это означает, что при установке операционной системы ее компонент не был установлен. Его можно доустановить.

Групповое выделение объектов

Для многих операций (удаление, копирование, перемещение и т.п.) требуется выделить не один объект, а несколько. До сих пор мы использовали для выделения щелчок мыши, но он позволяет выделить только один объект. Для группового выделения при щелчке надо держать нажатой клавишу SHIFT или CTRL

Если при щелчке держать нажатой клавишу CTRL, то выделение нового объекта не снимает выделение с объектов, выделенных ранее. Так можно выделить любую произвольную группу. Выделение при нажатой клавише CTRL действует как переключатель, то есть повторный щелчок на выделенном объекте снимает выделение.

Если выделяемые объекты расположены подряд, то можно воспользоваться клавишей SHIFT. В этом случае при нажатой клавише щелкают на первом выделяемом объекте группы и на последнем. Все промежуточные объекты выделяются автоматически. Для того чтобы использовать этот прием группового выделения, иногда бывает полезно предварительно упорядочить (отсортировать) объекты, представленные в окне.

Основные сведения:

Создание ярлыков

Ярлык - это ссылка на объект. Ярлыки удобны для организации быстрого доступа к файлам. Для создания нового ярлыка практически не требуется дополнительной памяти, в отличие от создания нового значка, что эквивалентно копии файла. При удалении ярлыка ликвидируется только ссылка на файл, а при удалении значка - сам файл.

Создание ярлыка осуществляется с помощью *Мастера создания ярлыков*, либо *копированием*.

Создание папок

Папка – это поименованная область на диске, предназначенная для хранения файлов и вложенных папок. Для создания папки на рабочем столе необходимо выполнить щелчок правой кнопкой мыши (вызвать контекстное меню) и

выбрать *Создать* ◊ *Папку*. Для создания вложенной папки необходимо выполнить команду *Файл* ◊ *Создать* ◊ *Папку*.

Выделение объектов

Для выделения объекта (файла или папки) необходимо выполнить одинарный щелчок левой кнопкой мыши по объекту. Выделить группу объектов можно щелчком левой кнопкой мыши и удерживая клавишу *Shift* (выделить первый объект, затем нажать клавишу *Shift* и выделить последний) или удерживая клавишу *Ctrl* (если объекты расположены не подряд).

Копирование объектов

Копирование объектов (файлов и папок) можно выполнить несколькими способами. Основные способы - перетаскиванием мышью и через буфер обмена.

Для копирования объектов с помощью перетаскивания мышью необходимо выделить объект или группу объектов и удерживая клавишу *Ctrl* перетащить левой кнопкой мыши в новую область (например, в папку).

При перетаскивании объекта с помощью правой кнопки мыши появляется контекстное меню, в котором предлагается выбрать действие, которое необходимо выполнить с объектом.

Для копирования через буфер обмена необходимо выделить объект или группу объектов и выполнить команду *Правка* ◊ *Копировать* или пункт *Копировать* контекстного меню объекта, затем перейти в область куда нужно выполнить вставку копированных объектов и выполнить команду *Правка* ◊ *Вставить* или пункт *Вставить* контекстного меню рабочей области.

Перемещение объектов

Перемещение объектов отличается от копирования тем, что при копировании объект остается и в папке-источнике и в папке-приемнике, а при перемещении объект переносится из папки-источника в папку-приемник.

Перемещение объектов (файлов и папок) можно выполнить несколькими способами. Основные способы - перетаскиванием мышью и через буфер обмена.

Для копирования объектов с помощью перетаскивания мышью необходимо выделить объект или группу объектов и удерживая клавишу *Shift* перетащить левой кнопкой мыши в новую область (например, в папку).

Для перемещения через буфер обмена необходимо выделить объект или группу объектов и выполнить команду *Правка* ◊ *Вырезать* или пункт *Вырезать* контекстного меню объекта, затем перейти в область куда нужно выполнить вставку копированных объектов и выполнить

команду **Правка** \diamond **Вставить** или рабочей области.

пункт **Вставить** контекстного меню

Переименование и удаление

Переименование файлов и папок выполняется через меню **Файл** \diamond **Переименовать**, щелчком мыши по имени выделенного значка или ярлыка или пунктом **Переименовать** в контекстном меню объекта.

Удаление файлов в корзину выполняется нажатием клавиши **Delete** или пунктом **Удалить** в контекстном меню объекта.

Заключительная часть.

1. Закончить изложение материала.
2. Ответить на возникшие вопросы.
3. Принять защиту выполненных ранее практических работ.
4. Подвести итоги занятия.
5. Выдать задание на самоподготовку (домашнее задание).

Задание на самоподготовку (домашнее задание):

1. Детально проработать, законспектировать материал занятия, размещенный в данном план-конспекте, в учебнике, указанном на с.2 текущего документа.
2. Подготовиться к опросу по пройденному материалу.