**Представление педагогического опыта по теме:**

**«Использование современных информационных технологий как средство повышения мотивации к изучению учебных дисциплин»**

***Козлова Светлана Ивановна,***

***преподаватель спец. дисциплин***

***ГБПОУ «ТКММП»***

Образование – это индустрия,

направленная в будущее.

С. П. Капица.

**Условия возникновения и становления опыта**

В соответствии с Концепцией модернизации российского образования основными целями являются: подготовка квалифицированного работника соответствующего уровня и профиля, конкурентоспособного на рынке труда, компетентного, ответственного, свободно владеющего своей профессией и ориентирующегося в смежных областях деятельности, способного к эффективной работе по специальности на уровне мировых стандартов, готового к постоянному профессиональному росту, социальной и профессиональной мобильности. Сегодня остро встает вопрос о подготовке новых кадров, которые смогут поднять страну на новую ступень ее развития.

Исходя из этой цели, ставлю перед собой задачи:

* Помочь учащимся освоить такие приёмы, которые позволят расширять полученные знания самостоятельно, т. е. научить оперативно осуществлять поиск информации, производить её структурирование, находить оптимальный алгоритм обработки;
* Способствовать развитию творческого потенциала учащихся;
* Создать условия для формирования у учащихся адекватной самооценки;
* Способствовать формированию коммуникабельности, умения работать в команде.

Поставленные задачи реализую на всех ступенях образовательного процесса, выстраивая свою деятельность в рамках образовательных программ, в которых определены цели, задачи, содержание обучения, программное, методическое и техническое обеспечение, принципы использования программ и критерии оценки их эффективности. Реализация этих программ невозможна без использования современных информационных технологий.

**Актуальность и перспективность опыта:**

Колоссальный объем информации и новых знаний по изучаемым предметам и невозможность освоить их, привели к следующим противоречиям:

* между желание ученика достичь профессионального роста в жизни и отсутствие практического опыта самостоятельной добычи новых знаний;
* между традиционными формами обучения и применением компьютерных технологий;
* между содержанием базового образования и формированием ключевых компетенций обучающихся;
* между наличием высоких требований, предъявляемых обществом к уровню образованности выпускника колледжа, и падение мотивации учащегося к обучению;

Мой педагогический опыт «Использование современных информационных технологий как средство повышения мотивации к изучению учебных дисциплин» актуален сегодня. Он позволяет найти возможные пути решения возникших противоречий.

Перспективность опыта заключается в реализации следующих задач:

* повышение интереса к предмету и положительная динамика качества знаний обучающихся, применение интерактивных форм работы в процессе организации учебной деятельности;
* внедрение новых информационно - коммуникационных технологий в процесс обучения;
* улучшение качества обучения посредством более полного использования доступной информации;
* интеграция различных видов деятельности (учебной, учебно -исследовательской, организационной) в рамках единой методологии, основанной на применении новых информационных технологий.
* формирование познавательного интереса к учебным дисциплинам, развитие творческих способностей, осознанных мотивов учения;

Использование программных продуктов на занятиях позволяет максимально точно изучить тему по учебным дисциплинам. Это дополнительно активизирует познавательную деятельность и способствует повышению мотивации учащихся к получению новых знаний.

**Теоретическая база опыта**

Проникновение современных информационных технологий в сферу образования позволяет педагогам качественно изменить содержание, методы и организационные формы обучения. Целью этих технологий в образовании является усиление интеллектуальных возможностей учащихся в информационном обществе, а также гуманизация, индивидуализация, интенсификация процесса обучения и повышение качества обучения на всех ступенях образовательной системы. В последние годы термин «информационные технологии» часто выступает синонимом термина «компьютерные технологии», так как все информационные технологии в настоящее время так или иначе связаны с применением компьютера.

Информационные и коммуникативные технологии создают широкие возможности для развития современного образования, прежде всего, в направлении индивидуализации, создают условия для реализации творческого потенциала преподавателя и студента.

Информационные технологии выступают уже не столько инструментами дополняющими систему образования, но императивом установления нового порядка знания и его инстуциональных структур. И для того, чтобы обеспечить потребности обучаемых в получении знаний, преподаватель должен овладеть информационными образовательными технологиями, а также, учитывая их развитие, постоянно совершенствовать свою информационную культуру путём самообразования, но при этом не злоупотреблять использованием данных технологий в своей практике и ко всему подходить творчески.

**Технология опыта**

Для создания условий, способствующих организации деятельностного подхода по преподаваемым дисциплинам, применяю ИКТ, что способствует активизации познавательной самостоятельности и активности учащихся. При преподавании дисциплин с использованием ИКТ ставлю следующие задачи:

* научить учащихся поиску, отбору и анализу необходимой учебной информации;
* регистрировать проблемные моменты в изучаемых темах по предметам, требующие построения модели средствами ИКТ;
* организовать поиск решения различными способами;
* применять наглядность и «живые» картинки для восприятия и познания закономерностей изучаемых явлений;
* организовать внеурочную деятельность, проектную и учебно-исследовательскую работу учащегося средствами ИКТ;
* использовать полученные знания на практике, в реальной жизни.

Применение ИКТ на занятиях открыло доступ к новым источникам информации и позволило отойти от традиционных форм изложения материала.

Характер использования ИКТ на занятиях может быть разный – это обучающий, развивающий, коммуникативный, диагностический, общекультурный. Это зависит от цели и задач, стоящих у меня, как преподавателя, при проектировании занятия с использованием компьютера. Провожу занятия демонстрационного типа, занятия обобщения и получения новых знаний, занятия компьютерного тестирования и др. Занятия проходят в кабинете информатики, используются компьютеры и экран, на который проектируется фрагмент занятия. В качестве программного обеспечения используем готовые продукты, разработанные производителями или разрабатываем свои на занятиях по программному модулю «Эксплуатация и модификация информационных систем». Создав свой продукт, студент получает огромное удовлетворение и положительную самооценку, что активизирует его познавательную деятельность и помогает самореализации личности. Работа проводится под руководством преподавателя, что способствует сотрудничеству между учащимися, взаимопомощи и взаимопониманию, партнерским отношениям «Преподаватель — студент». Технология применения ИКТ на занятиях должна быть не самоцелью, а сопровождать предмет для доступного, наглядного изложения материала, для организации интересной познавательной учебной деятельности.

Использование ИКТ начинаю с анализа темы, методов подачи материала, выделения проблемных моментов и путей их разрешения. При этом продумываю возможность разнообразить тип занятия через применение ИКТ. Разумеется, применение ИКТ должно быть методически обоснованно.

Занятия должны включать организационные и содержательные моменты. Орг.момент состоит в продумывании способов применения компьютера на уроках как средства информации, средства наглядности, поиска информации в Сети, создания проектов, учебно- исследовательской работы студентов, а содержательный этап – это проект занятия: цель занятия, задачи занятия, организация деятельности преподавателя и студента, анализ эффективности использования ИКТ на занятии, как средства активизации учебной, познавательной деятельности учащегося.

В своей работе НИТ (новых информационных технологий) в обучение я использую уже давно. Применяю следующие варианты НИТ.

* мультимедийные сценарии занятий;
* электронные учебники, обучающие системы, системы контроля знаний;
* программы имитационного моделирования, тренажеры;
* обучающие компьютерные программы, информационные системы;
* статистические, динамические и интерактивные модели.

Со студентами мы используем такие обучающие программы, как S2Netest, которая позволяет изучать и моделировать компьютерные сети. Работаем с программным продуктом 1С:Предприятие, применяем CASE средства для автоматизации процесса разработки информационной системы и программного обеспечения. Для проверки и обобщения знаний на занятиях использую программы – тесты. Например, при прохождении раздела «Обзор операционных систем» по дисциплине «Операционные системы», студенты сдавали зачет в виде теста за компьютером, созданного мною в программе «MyTest». Учащиеся выполнили задания, и компьютерная программа сама оценила их работу. Большая рутинная работа, связанная с проверкой тестов и их отработкой, возлагается на компьютер, что освобождает время у педагога.

**Анализ результативности**

Таким образом, творческий подход позволяет педагогу максимально эффективно использовать в своей работе богатый инструментарий, представляемый современными компьютерными технологиями. Занятия с использованием ИКТ повышают учебную мотивацию, а, следовательно, и интерес к предмету.

При активном использовании ИКТ успешнее достигаются общие цели образования, легче формируются компетенции в области коммуникации: умение собирать факты, их сопоставлять, организовывать, выражать свои мысли на бумаге и устно, логически рассуждать, слушать и понимать устную и письменную речь, открывать что-то новое, делать выбор и принимать решения.

Выпускник XXI века будет жить в мире компьютеров, в международном информационном обществе, и умение пользоваться информационными технологиями будет во многом определять его жизненный успех. Поэтому важно уже сейчас превратить информационное пространство  колледжа в среду, которая обогатит ученика, сформирует у него необходимые качества. Сделать это возможно, только активно применяя компьютерные информационные технологии в учебном процессе.

**Трудности и проблемы при использовании данного опыта.**

Трудности при использовании данного опыта появляются при выборе среди множества различного программного обеспечения именно то, которое будет более необходимым в их будущей профессии.

**Адресные рекомендации по использованию опыта**

Данный опыт может быть использован при подготовке и проведении различных учебных дисциплин преподавателями средних специальных учебных заведений.

1. В сфере преподавания различных предметов курса использование информационных и коммуникативных технологий открывает новые дидактические возможности, связанные с визуализацией материала, его «оживлением», возможностью представить наглядно те явления и процессы, которые невозможно продемонстрировать иными способами.
2. В сфере организации самостоятельной творческой работы учащихся информационные и коммуникативные технологии играют серьезную роль как инструмент поддержки инновационных технологий, в том числе и во внеурочной работе.
3. Информационно-коммуникативные технологии играют серьезную роль в изменении системы контроля знаний учащихся. Новые системы контроля знаний на базе информационных и коммуникативных технологий характеризуются оперативностью, регулярностью, создают широкие возможности для дифференциации (создание индивидуальных заданий, отличающихся уровнем сложности, темпом выполнения), обобщения результатов и накопления материалов, позволяющих оценивать личностную динамику ученика. Кроме того, они позволяют совмещать процедуры контроля и тренинга.

**Наглядное приложение:**

**Видео фрагмент занятия.**

**Конспект урока**

**по дисциплине ОП.02. «Операционные системы» специальности 09.02.04 Информационные системы (по отраслям)**

**Дисциплина:** ОП.02.Операционные системы

**Тема занятия: Модульная структура операционной системы** (слайд 1)

Актуализация знаний по разделу «Обзор операционных систем»

Выполнение теста за компьютером, подготовленного в программе MyTest

**Цели занятия:** (слайд 2)

1. Знакомство с модульной структурой операционных систем.
2. Изучение принципов работы ядра и вспомогательных модулей ОС, ядра в привилегированном режиме
3. Рассмотрение вопросов микроядерной архитектуры ОС (концепция, преимущества и недостатки)

**Содержание занятия:** (слайд 3)

* Ядро и вспомогательные модули
* Ядро в привилегированном режиме
* Многослойная структура ОС
* Многослойная структура ядра ОС
* Микроядерная архитектура
* Выводы

 **Ход занятия:**

Студентам раздается «Маркировочная таблица», которая состоит из трех колонок «знаю», «узнал новое», «хочу узнать подробнее». Данная таблица позволяет преподавателю контролировать работу каждого студента на занятии, степень его понимания изучаемой темы и интереса к ней.

**Архитектура операционной системы**

 Любая сложная система должна иметь понятную и рациональную структуру, то есть разделяться на части — модули, имеющие вполне законченное функциональное назначение с четко оговоренными правилами взаимодействия. Функциональная сложность операционной системы неизбежно приводит к сложности ее архитектуры.

Вопрос к студентам: (Перечислите основные функции ОС) (слайд 4)

Под архитектурой ОС понимают структурную организацию ОС на основе различных программных модулей.

 Обычно в состав ОС входят исполняемые и объектные модули стандартных для данной ОС форматов, библиотеки разных типов, модули исходного текста программ, программные модули специального формата (например, загрузчик ОС, драйверы ввода-вывода), конфигурационные файлы, файлы документации, модули справочной системы и т. д. (слайд 5)

Большинство современных операционных систем представляют собой хорошо структурированные модульные системы, способные к развитию, расширению и переносу на новые платформы. Какой-либо единой архитектуры ОС не существует, но существуют универсальные подходы к структурированию ОС.

**Ядро и вспомогательные модули ОС**

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

 ядро — модули, выполняющие основные функции ОС;

 модули, выполняющие вспомогательные функции ОС. (слайд 6)

Модули ядра выполняют такие базовые функции ОС, как управление процессами, памятью, устройствами ввода-вывода и т. п. Ядро составляет сердцевину операционной системы, без него ОС является полностью неработоспособной и не сможет выполнить ни одну из своих функций.

В состав ядра входят функции, решающие внутрисистемные задачи организации вычислительного процесса, такие как переключение контекстов, загрузка/выгрузка станиц, обработка прерываний. Эти функции недоступны для приложений. Другой класс функций ядра служит для поддержки приложений, создавая для них так называемую прикладную программную среду. (слайд 7)Приложения могут обращаться к ядру с запросами — системными вызовами — для выполнения тех или иных действий, например для открытия и чтения файла, вывода графической информации на дисплей, получения системного времени и т. д. Функции ядра, которые могут вызываться приложениями, образуют интерфейс прикладного программирования — API.

Функции, выполняемые модулями ядра, являются наиболее часто используемыми функциями операционной системы, поэтому скорость их выполнения определяет производительность всей системы в целом. Для обеспечения высокой скорости работы ОС все модули ядра или большая их часть постоянно находятся в оперативной памяти, то есть являются резидентными.

Ядро является движущей силой всех вычислительных процессов в компьютерной системе, и крах ядра равносилен краху всей системы. Поэтому разработчики операционной системы уделяют особое внимание надежности кодов ядра, в результате процесс их отладки может растягиваться на многие месяцы.

Обычно ядро оформляется в виде программного модуля некоторого специального формата, отличающегося от формата пользовательских приложений.

 Остальные модули ОС выполняют весьма полезные, но менее обязательные функции. Например, к таким вспомогательным модулям могут быть отнесены программы архивирования данных на магнитной ленте, дефрагментации диска, текстового редактора. Вспомогательные модули ОС оформляются либо в виде приложений, либо в виде библиотек процедур.

Поскольку некоторые компоненты ОС оформлены как обычные приложения, то есть в виде исполняемых модулей стандартного для данной ОС формата, то часто бывает очень сложно провести четкую грань между операционной системой и приложениями (рис. 3.1). (слайд 8)

Решение о том, является ли какая-либо программа частью ОС или! нет, принимает производитель ОС. Среди многих факторов, способных повлиять на это решение, немаловажными являются перспективы того, будет ли программа иметь массовый спрос у потенциальных пользователей данной ОС.

Некоторая программа может существовать определенное время как пользовательское приложение, а потом стать частью ОС, или наоборот. Ярким примером такого изменения статуса программы является Web-браузер компании Microsoft, который сначала поставлялся как отдельное приложение, затем стал частью операционных систем Windows NT 4.0 и Windows 95/98, а сегодня существует большая вероятность того, что по решению суда этот браузер снова превратится в самостоятельное приложение.



Рис. 3.1. Нечеткость границы между ОС и приложениями

Вспомогательные модули ОС обычно подразделяются на следующие группы:

 *утилиты* — программы, решающие отдельные задачи управления и сопровождения компьютерной системы, такие, например, как программы сжатия дисков, архивирования данных на магнитную ленту;

 *системные обрабатывающие программы* — текстовые или графические редакторы, компиляторы, компоновщики, отладчики;

 *программы предоставления пользователю дополнительных услуг* — специальный вариант пользовательского интерфейса, калькулятор и даже игры;

 *библиотеки процедур различного назначения*, упрощающие разработку приложений, например библиотека математических функций, функций ввода-вывода и т. д. (слайд 9) (Охарактеризуйте каждую из этих групп)

Как и обычные приложения, для выполнения своих задач утилиты, обрабатывающие программы и библиотеки ОС, обращаются к функциям ядра посредством системных вызовов (рис. 3.2). (слайд 10)

Разделение операционной системы на ядро и модули-приложения обеспечивает легкую расширяемость ОС. Чтобы добавить новую высокоуровневую функцию, достаточно разработать новое приложение, и при этом не требуется модифицировать ответственные функции, образующие ядро системы. Однако внесение изменений в функции ядра может оказаться гораздо сложнее, и сложность эта зависит от структурной организации самого ядра. В некоторых случаях каждое исправление ядра может потребовать его полной перекомпиляции.



Рис. 3.2. Взаимодействие между ядром и вспомогательными модулями ОС

 **Ядро в привилегированном режиме**

Для надежного управления ходом выполнения приложений операционная система должна иметь по отношению к приложениям определенные привилегии. Иначе некорректно работающее приложение может вмешаться в работу ОС и, например, разрушить часть ее кодов. Операционная система должна обладать исключительными полномочиями также для того, чтобы играть роль арбитра в споре приложений за ресурсы компьютера в мультипрограммном режиме. Ни одно приложение не должно иметь возможности без ведома ОС получать дополнительную область памяти, занимать процессор дольше разрешенного операционной системой периода времени, непосредственно управлять совместно используемыми внешними устройствами.

Обеспечить привилегии операционной системе невозможно без специальных средств аппаратной поддержки. Аппаратура компьютера должна поддерживать как минимум два режима работы — пользовательский режим (user mode) и привилегированный режим, который также называют режимом ядра (kernel mode), или режимом супервизора (supervisor mode). Подразумевается, что операционная система или некоторые ее части работают в привилегированном режиме, а приложения — в пользовательском режиме. (слайд 11)

Так как ядро выполняет все основные функции ОС, то чаще всего именно ядро становится той частью ОС, которая работает в привилегированном режиме (рис. 3.3). Иногда это свойство — работа в привилегированном режиме — служит основным определением понятия «ядро».



Рис. 3.3. Архитектура операционной системы с ядром в привилегированном режиме

Приложения ставятся в подчиненное положение за счет запрета выполнения в пользовательском режиме некоторых критичных команд, связанных с переключением процессора с задачи на задачу, управлением устройствами ввода-вывода, доступом к механизмам распределения и защиты памяти. Выполнение некоторых инструкций в пользовательском режиме запрещается безусловно (очевидно, что к таким инструкциям относится инструкция перехода в привилегированный режим), тогда как другие запрещается выполнять только при определенных условиях.

Аналогичным образом обеспечиваются привилегии ОС при доступе к памяти. Например, выполнение инструкции доступа к памяти для приложения разрешается, если инструкция обращается к области памяти, отведенной данному приложению операционной системой, и запрещается при обращении к областям памяти, занимаемым ОС или другими приложениями.

Повышение устойчивости операционной системы, обеспечиваемое переходом ядра в привилегированный режим, достигается за счет некоторого замедления выполнения системных вызовов. Системный вызов привилегированного ядра инициирует переключение процессора из пользовательского 'режима в- привилегированный, а при возврате к приложению — переключение из привилегированного режима в пользовательский (рис. 3.4).(слайд 12) Во всех типах процессоров из-за дополнительной двукратной задержки переключения переход на процедуру со сменой режима выполняется медленнее, чем вызов процедуры без смены режима.



Рис. 3.4. Смена режимов при выполнении системного вызова к привилегированному ядру

 Архитектура ОС, основанная на привилегированном ядре и приложениях пользовательского режима, стала, по существу, классической. Ее используют многие популярные операционные системы, в том числе многочисленные версии UNIX, VAX VMS, IBM OS/390, OS/2, и с определенными модификациями — Windows NT.

**Многослойная структура ОС**

 Вычислительную систему, работающую под управлением ОС на основе ядра, можно рассматривать как систему, состоящую из трех иерархически расположенных слоев: нижний слой образует аппаратура, промежуточный — ядро, а утилиты, обрабатывающие программы и приложения, составляют верхний слой системы (рис. 3.6). (слайд 13) Слоистую структуру вычислительной системы принято изображать в виде системы концентрических окружностей, иллюстрируя тот факт, что каждый слой может взаимодействовать только со смежными слоями. Действительно, при такой организации ОС приложения не могут непосредственно взаимодействовать с аппаратурой, а только через слой ядра.



Рис. 3.6. Трехслойная схема вычислительной системы

Многослойный подход является универсальным и эффективным способом декомпозиции сложных систем любого типа, в том числе и программных. В соответствии с этим подходом система состоит из иерархии слоев. Каждый слой обслуживает вышележащий слой, выполняя для него некоторый набор функций, которые образуют межслойный интерфейс (рис. 3.7).(слайд 14) На основе функций нижележащего слоя следующий (вверх по иерархии) слой строит свои функции — более сложные и более мощные, которые, в свою очередь, оказываются примитивами для создания еще более мощных функций вышележащего слоя. Строгие правила касаются только взаимодействия между слоями системы, а между модулями внутри слоя связи могут быть произвольными. Отдельный модуль может выполнить свою работу либо самостоятельно, либо обратиться к другому модулю своего слоя, либо обратиться за помощью к нижележащему слою через межслойный интерфейс.

Такая организация системы имеет много достоинств. Она существенно упрощает разработку системы, так как позволяет сначала определить «сверху вниз» функции слоев и межслойные интерфейсы, а затем при детальной реализации постепенно наращивать мощность функций слоев, двигаясь «снизу вверх». Кроме того, при модернизации системы можно изменять модули внутри слоя без необходимости производить какие-либо изменения в остальных слоях, если при этих внутренних изменениях межслойные интерфейсы остаются в силе.



Рис. 3.7. Концепция многослойного взаимодействия

Поскольку ядро представляет собой сложный многофункциональный комплекс, то многослойный подход обычно распространяется и на структуру ядра.

**Ядро может состоять из следующих слоев:** (слайд 15)

 Средства аппаратной поддержки ОС. До сих пор об операционной системе говорилось как о комплексе программ, но, вообще говоря, часть функций ОС может выполняться и аппаратными средствами. Поэтому иногда можно встретить определение операционной системы как совокупности программных и аппаратных средств, что и отражено на рис. 3.8.

Машинно-зависимые компоненты ОС. Этот слой образуют программные модули, в которых отражается специфика аппаратной платформы компьютера.

Базовые механизмы ядра. Этот слой выполняет наиболее примитивные операции ядра, такие как программное переключение контекстов процессов, диспетчеризацию прерываний, перемещение страниц из памяти на диск и обратно и т. п.

 Менеджеры ресурсов. Этот слой состоит из мощных функциональных модулей, реализующих стратегические задачи по управлению основными ресурсами вычислительной системы. Обычно на данном слое работают менеджеры (называемые также диспетчерами) процессов, ввода-вывода, файловой системы и оперативной памяти.

Интерфейс системных вызовов. Этот слой является самым верхним слоем ядра и взаимодействует непосредственно с приложениями и системными утилитами, образуя прикладной программный интерфейс операционной системы. 

Рис. 3.8. Многослойная структура ядра ОС

Приведенное разбиение ядра ОС на слои является достаточно условным. В реальной системе количество слоев и распределение функций между ними может быть и иным. В системах, предназначенных для аппаратных платформ одного типа, например ОС NetWare, слой машинно-зависимых модулей обычно не выделяется, сливаясь со слоем базовых механизмов и, частично, со слоем менеджеров ресурсов. Не всегда оформляются в отдельный слой базовые механизмы — в этом случае менеджеры ресурсов не только планируют использование ресурсов, но и самостоятельно реализуют свои планы.

Возможна и противоположная картина, когда ядро состоит из большего количества слоев. Например, менеджеры ресурсов, составляя определенный слой ядра, в свою очередь, могут обладать многослойной структурой.

Выбор количества слоев ядра является ответственным и сложным делом: увеличение числа слоев ведет к некоторому замедлению работы ядра за счет дополнительных накладных расходов на межслойное взаимодействие, а уменьшение числа слоев ухудшает расширяемость и логичность системы. Обычно операционные системы, прошедшие долгий путь эволюционного развития, например многие версии UNIX, имеют неупорядоченное ядро с небольшим числом четко выделенных слоев, а у сравнительно «молодых» операционных систем, таких как Windows NT, ядро разделено на большее число слоев и их взаимодействие формализовано в гораздо большей степени.

**Микроядерная архитектура** (слайд 16)

Микроядерная архитектура является альтернативой классическому способу построения операционной системы. Под классической архитектурой в данном случае понимается рассмотренная выше структурная организация ОС, в соответствии с которой все основные функции операционной системы, составляющие многослойное ядро, выполняются в привилегированном режиме. При этом некоторые вспомогательные функции ОС оформляются в виде приложений и выполняются в пользовательском режиме наряду с обычными пользовательскими программами (становясь системными утилитами или обрабатывающими программами).

Суть микроядерной архитектуры состоит в следующем. В привилегированном режиме остается работать только очень небольшая часть ОС, называемая микроядром (рис. 3.10). Микроядро защищено от остальных частей ОС и приложений. В состав микроядра обычно входят машинно-зависимые модули, а также модули, выполняющие базовые (но не все!) функции ядра по управлению процессами, обработке прерываний, управлению виртуальной памятью, пересылке сообщений и управлению устройствами ввода-вывода, связанные с загрузкой или чтением регистров устройств. Набор функций микроядра обычно соответствует функциям слоя базовых механизмов обычного ядра. Такие функции операционной системы трудно, если не невозможно, выполнить в пространстве пользователя.



Рис. 3.10. Перенос основного объема функций ядра в пользовательское пространство

Все остальные более высокоуровневые функции ядра оформляются в виде приложений, работающих в пользовательском режиме. Однозначного решения о том, какие из системных функций нужно оставить в привилегированном режиме, а какие перенести в пользовательский, не существует. В общем случае многие менеджеры ресурсов, являющиеся неотъемлемыми частями обычного ядра — файловая система, подсистемы управления виртуальной памятью и процессами, менеджер безопасности и т. п., — становятся «периферийными» модулями, работающими в пользовательском режиме.

Работающие в пользовательском режиме менеджеры ресурсов имеют принципиальные отличия от традиционных утилит и обрабатывающих программ операционной системы, хотя при микроядерной архитектуре все эти программные компоненты также оформлены в виде приложений. Утилиты и обрабатывающие программы вызываются в основном пользователями. Ситуации, когда одному приложению требуется выполнение функции (процедуры) другого приложения, возникают крайне редко. Поэтому в операционных системах с классической архитектурой отсутствует механизм, с помощью которого одно приложение могло бы вызвать функции другого.

 Совсем другая ситуация возникает, когда в форме приложения оформляется часть операционной системы. По определению, основным назначением такого приложения является обслуживание запросов других приложений, например создание процесса, выделение памяти, проверка прав доступа к ресурсу и т. д. Именно поэтому менеджеры ресурсов, вынесенные в пользовательский режим, называются серверами ОС, то есть модулями, основным назначением которых является обслуживание запросов локальных приложений и других модулей ОС. Очевидно, что для реализации микроядерной архитектуры необходимым условием является наличие в операционной системе удобного и эффективного способа вызова процедур одного процесса из другого. Поддержка такого механизма и является одной из главных задач микроядра.

Схематично механизм обращения к функциям ОС, оформленным в виде серверов, выглядит следующим образом (рис. 3.11). (слайд 17) Клиент, которым может быть либо прикладная программа, либо другой компонент ОС, запрашивает выполнение некоторой функции у соответствующего сервера, посылая ему сообщение. Непосредственная передача сообщений между приложениями невозможна, так как их адресные пространства изолированы друг от друга. Микроядро, выполняющееся в привилегированном режиме, имеет доступ к адресным пространствам каждого из этих приложений и поэтому может работать в качестве посредника. Микроядро сначала передает сообщение, содержащее имя и параметры вызываемой процедуры нужному серверу, затем сервер выполняет запрошенную операцию, после чего ядро возвращает результаты клиенту с помощью другого сообщения. Таким образом, работа микроядерной операционной системы соответствует известной модели клиент-сервер, в которой роль транспортных средств выполняет микроядро.



Рис. 3.11. Реализация системного вызова в микроядерной архитектуре

Преимущества и недостатки микроядерной архитектуры (слайд 18)

 Операционные системы, основанные на концепции микроядра, в высокой степени удовлетворяют большинству требований, предъявляемых к современным ОС, обладая переносимостью (т.е. возможность добавлять новые сервисы, или модифицировать, или убирать старые) расширяемостью (т.е. минимум изменений кода микроядра за счет его компактности), надежностью (поскольку каждый сервер выполняется в виде отдельного процесса в своей области памяти и в случае краха может быть перезапущен без остановки остальных серверов. Кроме того, компактное микроядро легче протестировать и получить вследствие этого более качественный код) и создавая хорошие предпосылки для поддержки распределенных приложений за счет использования единообразного интерфейса запросов, генерируемых процессами, и поддержки механизма взаимодействия клиентов и серверов путем обмена сообщениями. За эти достоинства приходится платить снижением производительности (создание сообщения и отправка его через микроядро с последующим получением ответа занимает больше времени, чем непосредственный вызов сервиса), и это является основным недостатком микроядерной архитектуры.

**Выводы** (слайд 19-20)

 Простейшая структуризация ОС состоит в разделении всех компонентов ОС на модули, выполняющие основные функции ОС (ядро), и модули, выполняющие вспомогательные функции ОС. Вспомогательные модули ОС оформляются либо в виде приложений (утилиты и системные обрабатывающие программы), либо в виде библиотек процедур. Вспомогательные модули загружаются в оперативную память только на время выполнения своих функций, то есть являются транзитными. Модули ядра постоянно находятся в оперативной памяти, то есть являются резидентными.

 При наличии аппаратной поддержки режимов с разными уровнями полномочий устойчивость ОС может быть повышена путем выполнения функций ядра в привилегированном режиме, а вспомогательных модулей ОС и приложений — в пользовательском. Это дает возможность защитить коды и данные ОС и приложений от несанкционированного доступа. ОС может выступать в роли арбитра в спорах приложений за ресурсы.

Ядро, являясь структурным элементом ОС, в свою очередь, может быть логически разложено на следующие слои (начиная с самого нижнего):

машинно-зависимые компоненты ОС;

 базовые механизмы ядра;

 менеджеры ресурсов;

 интерфейс системных вызовов.

 В многослойной системе каждый слой обслуживает вышележащий слой, выполняя для него некоторый набор функций, которые образуют межслойный интерфейс. На основе функций нижележащего слоя следующий вверх по иерархии слой строит свои функции — более сложные и более мощные, которые, в свою очередь, оказываются примитивами для создания еще более мощных функций вышележащего слоя. Многослойная организация ОС существенно упрощает разработку и модернизацию системы.

Микроядерная архитектура является альтернативой классическому способу построения операционной системы, в соответствии с которым все основные функции операционной системы, составляющие многослойное ядро, выполняются в привилегированном режиме. В микроядерных ОС в привилегированном режиме остается работать только очень небольшая часть ОС, называемая микроядром. Все остальные высокоуровневые функции ядра оформляются в виде приложений, работающих в пользовательском режиме.

 Микроядерные ОС удовлетворяют большинству требований, предъявляемых к современным ОС, обладая переносимостью, расширяемостью, надежностью и создавая хорошие предпосылки для поддержки распределенных приложений. За эти достоинства приходится платить снижением производительности, что является основным недостатком микроядерной архитектуры.

**Выдача домашнего задания:** (слайд 21)

Попов И.И. и др. «Операционные системы, среды и оболочки», работа с учебным материалом, проработка конспекта:

1. Опишите порядок взаимодействия приложения с ОС, имеющей микроядерную архитектуру.
2. Подготовить сообщение о монолитных ОС.

Приложение 1

**Тест по разделу «Обзор операционных систем»**

1.В каком периоде истории эволюции вычислительных систем появилась первая операционная система

А) первый период (1945г-1955г)

Б) второй период (1955г - начало 60-х)

В) третий период (начало 60-х - 1980г)

2. Может ли компьютер работать без операционной системы?

А) да

Б) нет

В) не знаю

3. ОС - комплекс взаимосвязанных программ действующих как интерфейс между приложениями и пользователями, с одной стороны, а с другой стороны...

А) аппаратурой компьютера

Б) внешними устройствами

В) оперативной памятью

4. BIOS - это ...

А) программа-драйвер

Б) программа – приложение

В) программа, выполняющая тестирование компьютерной системы после включения компьютера

5. Программа, управляющая конкретной моделью внешнего устройства и учитывающая все его особенности называется?

А) приложением

Б) драйвером

В) системной обрабатывающей программой

6. Утилита это…

А) прикладная программа

Б) сервисная программа

В) базовая система ввода-вывода

7. ОС, позволяющие одновременно работать нескольким пользователям называются....

А) многозадачными

Б) многопроцессорными

В) многопользовательскими

8. ОС, позволяющие одновременно выполнять только одну задачу, называются...

А) однозадачными

Б) однопользовательскими

В) однопроцессорными

9. Способ обработки данных, при котором обеспечивается определенное гарантированное время ответа на запрос пользователя называется....

А) пакетный режим

Б) режим разделения времени

В) режим реального времени

10. По типу использования ресурсов ОС разделяются на:

А) сетевые и локальные

Б) сетевые и аппаратные

В) локальные и программные

11. Системные процессы инициализируются....

А) по инициативе пользователя

Б) самой ОС для выполнения своих функций

В) верны оба варианта

12. Функциями ОС по управлению памятью не являются

А) отслеживание свободной и занятой памяти

Б) выделение памяти процессом и освобождение памяти при завершении процессов

В) обеспечивание процессов необходимыми системными ресурсами

Ключ к тесту:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Б | А | А | В | Б | Б | В | А | В | А | Б | В |

Приложение 2

Маркировочная таблица

|  |  |  |
| --- | --- | --- |
| ЗНАЮ | УЗНАЛ НОВОЕ | ХОЧУ УЗНАТЬ ПОДРОБНЕЕ |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |