

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению практических работ
по дисциплине «Безопасность операционных систем»
для студентов направления подготовки
10.03.01 «Информационная безопасность»
Направленность (профиль)
«Организация и технологии защиты информации (по отрасли или в сфере профессиональной
деятельности)»

Ставрополь
2026

СОДЕРЖАНИЕ

Введение	
Практическая работа №1	5
Тема: Безопасность компьютерных сетей Протоколирование и аудит	5
Практическая работа №2	20
Тема: Антивирусная защита Протоколирование и аудит	20
Практическая работа №3	28
Тема: Системы безопасности ОС Linux Системы безопасности ОС Windows	28
Практическая работа №4	39
Тема: Модели безопасности основных операционных систем Программно-технический уровень информационной безопасности	39
Практическая работа №5	50
Тема: Угрозы безопасности информации в информационно-вычислительных системах Основные понятия и положения защиты информации в информационно-вычислительных системах	50
Практическая работа №6	69
Тема: Архитектура ОС Linux Архитектура ОС Windows	69
Практическая работа №7	84
Тема: Прерывания, ресурсы и ядро ОС Управление файлами и вводом-выводом в ОС	84
Практическая работа №8	94
Тема: Управление памятью в ОС Управление процессами в ОС	94
1. Подготовка к занятию	94
Практическая работа №9	99
Тема: Концептуальные основы ОС Принципы построения ОС	99

ВВЕДЕНИЕ

1. Цель и задачи освоения дисциплины

Цель освоения дисциплины «Безопасность операционных систем» является обучение принципам построения защиты информации в операционных системах (ОС) и анализа надежности защиты ОС, а также приобретение набора общекультурных и общепрофессиональных компетенций будущего бакалавра по направлению подготовки 10.03.01 «Информационная безопасность»

Задачами дисциплины являются:

- теоретическая и практическая подготовка бакалавров к настройке параметров безопасности различных типов операционных систем;
- объяснение принципов построения подсистем защиты в ОС различной архитектуры, средств и методов несанкционированного доступа к ресурсам ОС;
- привитие студентам системного подхода к проблеме защиты информации в ОС; механизмов защиты информации и возможностей по их преодолению.

2. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
ПК-1 Способен обеспечивать защиту от несанкционированного доступа сооружений и средств связи сетей электросвязи (за исключением сетей связи специального назначения) в процессе их эксплуатации	ИД-1 ПК-1 Способен проводить мониторинг функционирования СССЭ, защищённости от НСД сооружений и СССЭ	- Применяет методы контроля функционирования СССЭ, их защищенности от НСД - Анализирует средства мониторинга работоспособности и эффективности применяемых программных, программно-аппаратных (в том числе криптографических) и технических средств защиты СССЭ от НСД
	ИД-2 ПК-1 Способен к управлению функционированием СССЭ, защищённостью от НСД сооружений СССЭ	- определяет необходимый состав, особенности размещения СССЭ, а также программных, программно-аппаратных (в том числе криптографических) и технических средств и систем защиты СССЭ от НСД; - организывает и проводит монтаж и настройку СССЭ, а также программных, программно-аппаратных (в том числе криптографических) и технических средств и систем защиты СССЭ от НСД; - устанавливает и настраивает программное обеспечение, необходимое для управления СССЭ и средствами их защиты от НСД

ПК-3 Способен обеспечивать защиту информации в автоматизированных системах в процессе их эксплуатации	ИД-1 ПК-3 Способен администрировать системы защиты информации автоматизированных систем	Устанавливает и настраивает операционные системы, системы управления базами данных, компьютерные сети и программные системы с учетом требований по обеспечению защиты информации
---	---	--

Практическая работа №1

Тема: Безопасность компьютерных сетей Протоколирование и аудит

1. Подготовка к занятию

1. Изучить (повторить) теоретический материал,.
2. Ознакомиться с программой работы.
3. Подготовить отчет о работе.
4. Ответить на контрольные вопросы.

2. Распределение времени занятия:

Всего: 90 мин

Вступительная часть 2 мин

Проверка готовности студентов к занятию 5 мин

Основная часть – 70 мин

Проверка выполнения практического занятия 10 мин

Заключительная часть 3 мин

3. Формируемые компетенции: ПК-1, ПК-3

4. Правила работы в лаборатории

К работе в лаборатории допускаются лица, изучившие правила и меры безопасности, сдавшие зачет по ним и усвоившие порядок выполнения лабораторной работы.

4.1. Требования безопасности перед началом работ

- ЗАПРЕЩАЕТСЯ: переодеваться, пользоваться огнем, курить, принимать пищу в лаборатории.

- Убедиться в целостности электрических розеток и разъемов. В лаборатории необходимо быть в сменной обуви.

- Включение компьютера производить только после получения допуска по выполняемой работе и разрешения преподавателя или лаборанта.

4.2. Требования безопасности во время работы

- выполняя практическое занятие, студенты обязаны использовать только вычислительную технику, периферийное оборудование, соединительные кабели, измерительное оборудование и носители информации, непосредственно относящиеся к данному лабораторному занятию;

- подключение и отключение составляющих вычислительного комплекса производить только при полном снятии напряжения со всех составляющих вычислительного комплекса;

- при обнаружении неисправностей в оборудовании немедленно отключить источники питания и доложить об этом руководителю занятий или лаборанту.

4.3. Требования безопасности по окончании работы

- доложить руководителю занятий или лаборанту о завершении работ;

- привести в порядок и сдать рабочее место лаборанту, и доложить руководству

1. Теоретическая часть: Файловые системы

1.1. Файловые системы Windows

Операционная система Microsoft Windows XP в бета версиях, известная как Microsoft Codename Whistler, является продолжением линейки Windows NT. Это полностью 32 разрядная операционная система с приоритетной многозадачностью. В ее основе лежат базовые принципы:

- совместимость – поддержка файловых систем FAT 16, FAT32 и NTFS, поддержка приложений написанных под DOS, Windows 9x, Windows NT, а также некоторых приложений под OS/2 и POSIX;
- переносимость – реализация поддержки процессоров разных архитектур;
- реализация системы безопасности на уровне пользователей.

Первоначально Microsoft планировала разработку двух независимых операционных систем - Neptun (эта система должна была стать продолжением Windows 9x) и Odyssey (должна была стать продолжением линейки Windows NT). Однако впоследствии планы корпорации изменились и обе разработки были объединены в один проект Windows XP – операционную систему с полностью переработанным интерфейсом, новыми возможностями и более высоким уровнем обеспечения безопасности.

Все операционные системы, как современные, так и давно уже неиспользуемые, имеют одну общую черту – хранение информации в операционных системах осуществляется подсистемой, называемой файловой системой.

Файловая система – это набор спецификаций и соответствующее им программное обеспечение, которое отвечает за создание, удаление, организацию, чтение, запись, модификацию и перемещение файлов информации, а также за управление доступом к файлам и за управление ресурсами, которые используются файлами. Файловая система определяет способ организации данных на диске и принципы хранения данных на физическом носителе. Например, как должны сохраняться данные файла, какая информация (например, имя, дата создания и т.п.) о файле должна храниться и каким образом. Формат хранения данных определяет основные характеристики файловой системы.

Информация на магнитных дисках размещается и передается блоками. Каждый блок называется сектором и располагается на концентрических дорожках поверхности диска. Группа дорожек одного радиуса, расположенных на поверхностях магнитных дисков, образуют цилиндры. Каждый сектор состоит из поля данных и поля служебной информации, ограничивающей и идентифицирующей его. Размер сектора (объем поля данных) устанавливается контроллером или драйвером. Физический адрес сектора на диске определяется с помощью трех «координат»:

- номер цилиндра;
- номер рабочей поверхности диска;
- номер сектора на дорожке.

Обмен информацией между оперативно запоминающим устройством и дисками физически осуществляется только секторами. Диск может быть разбит на несколько разделов, которые могут использоваться как одной операционной системой, так и несколькими. На каждом разделе может быть организована своя файловая система. Для организации хотя бы одной файловой системы должен быть определен, по крайней мере, один раздел. Разделы могут быть двух типов:

- первичный раздел;
- расширенный раздел.

Максимальное число первичных разделов – четыре, но обязательно должен быть хотя бы один. Если первичных разделов больше одного, то один должен быть активным, в нем находится загрузчик операционной системы. На одном диске может быть только один расширенный раздел, который в свою очередь может содержать большое количество подразделов – логических дисков.

Операционная система Windows XP поддерживает работу со следующими файловыми системами:

- FAT (File Allocation Table) – файловая система, разработанная для MS-DOS и являющаяся основной для Windows 3.x и 9x. Windows XP и Windows Server 2003 поддерживают три разновидности FAT: FAT12, FAT16 и FAT32. Первые две обеспечивают совместимость со старыми операционными системами Microsoft. Кроме того, FAT12 используется как формат хранения данных на гибких дисках. FAT 32 – модифицированная версия FAT, используемая в Windows 95 OSR2, Windows 98 и Windows Millennium.

- NTFS (Windows NT file system) – файловая система, разработанная специально для Windows NT и унаследованная Windows 2000, Windows XP, Windows 2003.

- CDFS (Compact Disk File System) – файловая система компакт-дисков.

- UDF (Universal Disk Format) – универсальный формат дисков, используемый современными магнитооптическими накопителями и технологией DVD.

- DFS (Distributed File System) – распределенная файловая система.

Возможность поддержки различных файловых систем в линейке современных операционных систем семейства Windows заложена в архитектуре системы ввода-вывода, которая отвечает за обработку запросов ввода-вывода и выполняет следующие задачи:

- обеспечение работы сверхпроизводительных операций ввода-вывода;
- возможность использования асинхронного ввода-вывода;
- поддержка нескольких файловых систем;
- модульная архитектура, с возможностью добавления новых файловых систем и устройств;
- предоставление расширенных возможностей, например кэширования;
- защита совместно используемых ресурсов.

Список зарегистрированных файловых систем можно посмотреть с помощью утилиты WinObj. У каждой системы есть свои полезные свойства, но возможности защиты и аудита различны. На выбор файловой системы оказывают влияние следующие факторы: цель, для которой предполагается использовать компьютер, аппаратная платформа, количество жестких дисков и их объем, требования к безопасности, используемые в системе приложения.

1. Файловая система FAT

Аббревиатура FAT (File Allocation Table) означает «таблица размещения файлов». Этот термин относится к линейной табличной структуре со сведениями о файлах – именами файлов, их атрибутами и другими данными, определяющими местоположение файлов или их фрагментов в среде FAT. Элемент FAT определяет фактическую область диска, в котором хранится начало физического файла. В файловой системе FAT логическое дисковое пространство любого логического диска состоит из двух областей:

- системная область – создается при форматировании диска и обновляется при манипулировании файловой структурой;
- область данных – содержит файлы и каталоги, подчиненные корневому каталогу, доступна через пользовательский интерфейс.

Системная область состоит из следующих компонентов:

- загрузочной записи;
- зарезервированных секторов;
- таблицы размещения файлов (FAT);
- корневого каталога.

Таблица размещения файлов представляет собой карту (образ) области данных, в которой описывается состояние каждого участка области данных. Область данных разбивается на кластеры. Один или несколько смежных секторов в логическом дисковом адресном пространстве (только в области данных) объединяются в единый дисковый блок – кластер.

Кластер – минимальная адресуемая единица дисковой памяти, выделяемая файлу или некорневому каталогу. Например, в FAT16 размер кластера составляет 32 кбайт. Файл или каталог занимает целое число кластеров. Последний кластер при этом может быть задействован не полностью, что приведет к заметной потере дискового пространства при большом размере

кластера.

В таблице FAT кластеры, принадлежащие одному файлу (некорневому каталогу), связываются в цепочки. Для указания номера кластера в системе управления файлами FAT16 используется 16-битовое слово, следовательно, можно хранить информацию максимум о 65536 кластерах. Так как FAT используется при доступе к диску очень интенсивно, она загружается в оперативную память и находится там максимально долго.

Корневой каталог отличается от обычного каталога тем, что он размещается в фиксированном месте логического диска и имеет фиксированное число элементов. Структура системы файлов является иерархической. Файлам присваиваются первые доступные адреса кластеров в томе. Номер начального кластера файла представляет собой адрес первого кластера, занятого файлом, в таблице размещения файлов. Каждый кластер содержит указатель на следующий кластер, использованный файлом, или индикатор (0xFFFF), указывающий, что данный кластер является последним кластером файла.

Файлы на дисках имеют 4 атрибута, которые могут сбрасываться и устанавливаться пользователем: Archive (архивный), System (системный), Hidden (скрытый) и Read-only (только чтение).

Файловая система FAT первоначально была ориентирована на небольшие диски и простые структуры каталога. Затем ее усовершенствовали для обеспечения работы с большими дисками и мощными персональными компьютерами. Windows XP и Windows Server 2003 поддерживают файловую систему FAT по трем причинам:

- для возможности обновления операционной системы с прежних версий Windows;
- для совместимости с другими операционными системами при многовариантной загрузке;
- как формат гибких дисков.

В название каждой версии FAT входит число, которое указывает разрядность, применяемую для идентификации кластеров на диске. Двенадцатиразрядный идентификатор кластеров в FAT12 ограничивает размер дискового раздела 212 (4096) кластерами. В Windows используются кластеры размером от 512 байт до 8 Кб, так что размер тома FAT12 ограничен 32 Мб. Поэтому Windows использует FAT12 как формат 5,25 и 3,5-дюймовых дискет, способных хранить до 1,44 Мб данных. FAT16 за счет 16-разрядных идентификаторов кластеров может адресовать до 216 (65536) кластеров. В Windows размер кластера FAT16 варьируется от 512 байт до 64 Кб, поэтому размер тома FAT16 ограничен 4 Гб. Размер кластеров, используемых Windows, зависит от размера тома.

Таблица 1

Размеры кластеров в FAT16 по умолчанию (в Windows)

№ п/п	Размер тома (Мб)	Размер кластера
1	0-32	512 байт
2	33-64	1 Кб
3	65-128	2 Кб
4	129-256	4 Кб
5	257-511	8 Кб
6	512-1023	16 Кб
7	1024-2047	32Кб
8	2048-4095	64Кб

Файловая система FAT32 обеспечивает оптимальный доступ к жестким дискам, CD-ROM и сетевым ресурсам, повышая скорость и производительность всех операций ввода/вывода. FAT32 представляет собой усовершенствованную версию FAT, предназначенную для использования на томах, объем которых превышает 2 Гбайта. Размер кластера в FAT32 равен 4 кбайт. FAT32 является полностью независимой 32-разрядной файловой системой и содержит многочисленные усовершенствования и дополнения по сравнению с FAT16. Принципиальное отличие FAT32 заключается в более эффективном использовании дискового пространства за счет использования

кластеров меньшего размера, что приводит к экономии дискового пространства. FAT32 может перемещать корневой каталог и использовать резервную копию FAT вместо стандартной. Расширенная загрузочная запись FAT32 позволяет создавать копии критических структур данных, что повышает устойчивость дисков к нарушениям структуры FAT по сравнению с предыдущими версиями. Корневой каталог представляет собой обычную цепочку кластеров, поэтому может находиться в произвольном месте диска, что снимает ограничение на размер корневого каталога.

FAT32 использует 32-разрядные идентификаторы кластеров, но при этом резервирует старшие 4 бита, так что эффективный размер идентификатора кластера составляет 28 бит. Поскольку максимальный размер кластеров FAT32 равен 32 Кб, теоретически FAT32 может работать с 8-терабайтными томами. Однако реализация FAT32 в Windows XP / Windows 2003 не позволяет создавать тома, превышающие 32 Гб, хотя операционная система может задействовать существующие тома FAT32 любого размера.

Таблица 2

Размеры кластеров в FAT32 по умолчанию (в Windows)

№ п/п	Размер тома (Мб)	Размер кластера
1	От 32 Мб до 8 Гб	4
2	8-16 Гб	8
3	16-32 Гб	16
4	32Гб	32

Файловая система FAT не обеспечивает функций защиты данных и автоматического восстановления. Поэтому она используется, только если альтернативной системой на компьютере является MS-DOS или Windows 95/98, а также для передачи данных на гибких дисках.

2. Файловая система NTFS

Аббревиатура NTFS (New Technology File System) означает новая технология файловой системы. NTFS является наиболее надежной системой специально разработанной для Windows NT и усовершенствованной в более поздних версиях Windows. Она обладает характеристиками защищенности, поддерживая контроль доступа к данным и привилегии владельца, играющие важную роль в обеспечении целостности конфиденциальных данных. Папки и файлы NTFS могут иметь назначенные им права доступа вне зависимости от того, являются они общими или нет. Если файл будет скопирован из раздела или тома NTFS в раздел или на том FAT, все права доступа и другие уникальные атрибуты, присущие NTFS, будут утрачены.

NTFS использует 64-разрядные индексы кластеров, но Windows XP ограничивает размеры томов NTFS до значений, при которых возможна адресация 32-разрядными кластерами, то есть до 128 Тб (с использованием кластеров по 64 Кб).

Таблица 3

Размеры кластеров на томах NTFS

№ п/п	Размер тома (Мб)	Размер кластера
1	512 Мб и менее	512 байт
2	513 - 1024 Мб	1 Кб
3	1025 -2048 Мб	2 Кб
4	более 2048 Мб (2 Гб)	4 Кб

Одно из важнейших свойств NTFS – самовосстановление. При неожиданном сбое системы информация о структуре папок и файлов на томе FAT может быть утеряна. NTFS протоколирует все вносимые изменения, что позволяет избежать разрушения данных о структуре тома (в некоторых случаях данные файлов могут быть утеряны).

Способность самовосстановления и поддержка целостности реализуется за счет использования протокола выполняемых действий и ряда других механизмов. NTFS рассматривает каждую операцию, модифицирующую системные файлы на NTFS-томах, как транзакцию¹ и сохраняет информацию о такой транзакции в протоколе. Начатая транзакция может быть либо полностью завершена (commit), либо откатывается (rollback). В последнем случае NTFS-том возвращается в состояние, предшествующее началу транзакции. Для того чтобы управлять транзакциями, NTFS записывает все операции, входящие в транзакцию, в файл протокола, перед тем как осуществить запись на диск. После того как транзакция завершена, все операции выполняются. Таким образом, под управлением NTFS не может быть незавершенных операций. В случае дисковых сбоев незавершенные операции отменяются.

Под управлением NTFS также выполняются операции, позволяющие определять дефектные кластеры и отводить новые кластеры для файловых операций. Этот механизм называется cluster remapping. NTFS, по сравнению с FAT, поддерживает ряд дополнительных возможностей, основные из них:

- защита файлов и каталогов;
- сжатие файлов;
- поддержка многопоточных файлов;
- отслеживание связей;
- дисковые квоты;
- шифрование;
- точки повторной обработки;
- точки соединения;
- теневые копии.

Защита файлов и папок. Структурой NTFS предусмотрено хранение для каждого файла и каждой папки специального блока безопасности, который содержит следующую информацию:

- идентификатор (имя) пользователя, создавшего файл;
- список контроля доступа, в котором перечислены разрешения доступа к файлу или папке для пользователей и групп;
- системный список контроля доступа, в котором перечислено, какие действия (например, чтение, запись и т.п.) для каких пользователей и групп необходимо фиксировать в журнале аудита.

Это позволяет операционной системе: обеспечивать разграничение доступа к файлам и папкам и фиксировать действия, выполняемые пользователями над объектами. Поскольку на томах FAT подобная информация не хранится, то защита файлов и папок на них не осуществляется.

Сжатие файлов и каталогов. NTFS обеспечивает динамическое сжатие файлов и каталогов. Сжатие является атрибутом файла или каталога, который можно снять или установить. Сжатие возможно только на разделах, размер блока которых не превышает 4096 байтов. Если каталог имеет атрибут сжатый (compressed), все файлы, копируемые в него, тоже получают этот атрибут. Производительность компьютера при использовании сжатых файлов возрастает до 50% в зависимости от типа хранимых данных. Такой результат достигается за счет повышения загрузки процессора в 3-5 раз. Однако на больших (более 4 Гб) разделах и на отказоустойчивых томах производительность заметно снижается. Поэтому рекомендуется использовать функцию сжатия на небольших томах в компьютерах с быстрыми процессорами или в многопроцессорных системах.

¹ Транзакция – совокупность операций над данными, которая, с точки зрения обработки данных, либо выполняется полностью, либо совсем не выполняется

Многопоточные файлы. Например, нужно иметь две версии текста контракта: одну на русском, другую на языке, приемлемом для фирмы. Можно создать несколько разных версий файлов и пересылать их вместе, но удобнее использовать специальную версию текстового процессора, в меню которого можно указать желаемый язык документа, и он будет извлечен из одного общего файла. Для реализации такой функциональности применяются именованные потоки NTFS. При создании нового файла (например, текстовым редактором), данные по умолчанию заносятся в неименованный поток файла.

Однако у того же файла могут быть и именованные потоки, которые записываются следующим образом: файл.txt: первый поток; файл.txt: второй поток; файл.txt: третий поток. В каждый из этих потоков заносится своя информация. Именованные потоки используются только на NTFS. При копировании многопоточного файла на диск, отформатированный под FAT, операционная система предупредит о потере данных; при копировании посредством командной строки скопирован будет только неименованный поток и система не предупредит о потере данных.

Отслеживание связей. Ярлыки играют важную роль организации доступа пользователя к программам и файлам. Однако они имеют и недостатки, одним из которых является нарушение связи между ярлыком и соответствующим ему ресурсом, если ресурс переносится в другое место или переименовывается. В Windows XP / Windows 2003 работает служба отслеживания изменившихся связей (Distributed Link Tracking), позволившая приложениям находить ресурс, соответствующий данному ярлыку, и связи OLE даже в случае, если этот ресурс был переименован или перенесен в другое место дерева папок.

Каждая связь состоит из двух частей – клиента и источника. Например, если документ Word содержит связь OLE с электронной таблицей Excel, сам документ является клиентом связи, а электронная таблица – источником связи. Служба отслеживания восстанавливает разрушенную связь в случаях, если:

- источник связи был переименован;
- источник связи был перемещен с одного тома NTFS 5.0 на другой в пределах одного компьютера;
- источник связи был перемещен с тома NTFS 5.0 одного компьютера на том NTFS 5.0 другого компьютера;
- том NTFS 5.0 с источником связи был физически перемещен с одного компьютера на другой компьютер в пределах одного домена;
- компьютер, на котором находится том NTFS 5.0 с источником связи, был переименован, но остался в том же домене;
- изменилось имя общего ресурса, где находится файл-источник связи, и образовалась любая комбинация описанных выше случаев.

Отслеживаются только источники связей, находящиеся на томах NTFS 5.0. Если источник перемещен в другую файловую систему, то попытки отследить изменившуюся связь будут предприняты, но вероятность успешного результата мала. Если источник опять будет перенесен в NTFS 5.0, связь будет восстановлена. В текущей версии NTFS 5.0 во время работы службы отслеживания связей, тома NTFS 5.0 не могут быть заблокированы. Поэтому для них нельзя выполнить такие операции, как форматирование или запуск утилиты chkdsk /f. Выполнять подобные операции можно только после остановки работы службы отслеживания связей.

Квоты дискового пространства. В случае одновременной работы нескольких пользователей возникают ситуации учета дискового пространства, занятого их файлами. Ситуация разрешается с помощью введения квот на дисковое пространство, доступное для работы каждому пользователю. Администратор может кватировать дисковое пространство по каждому тому и для каждого пользователя. (Нельзя задать квоту для подкаталога или для группы.). Windows учитывает пространство, занимаемое файлами, владельцем которых является контролируемый пользователь: если пользователь владеет файлом, размер последнего добавляется к общей сумме занимаемого пользователем дискового пространства. Поскольку кватирование выполняется по каждому тому, то не имеет значение, находятся ли тома на одном физическом диске или на

различных устройствах.

После установки квот дискового пространства пользователь может хранить на томе ограниченный объем данных, в то время как на этом томе может оставаться свободное пространство. Если пользователь превышает выданную ему квоту, в журнал событий вносится соответствующая запись. Затем, в зависимости от конфигурации системы, пользователь либо сможет записать информацию на том (более мягкий режим), либо ему будет отказано в записи из-за отсутствия свободного пространства. Устанавливать и просматривать квоты на диске можно только в разделе с NTFS 5.0 и при наличии необходимых полномочий (задаваемых с помощью локальных или доменных групповых политик) у пользователя, устанавливающего квоты.

Точки повторной обработки (repase points). Точки позволяют выполнять при открытии папки или файла заранее созданный программный код. Точка повторной обработки – это контролируемый системой атрибут, который может быть ассоциирован с папкой или файлом. Значением атрибута являются задаваемые пользователем данные, максимальный размер которых может достигать 16 Кбайт. Они представляют собой 32-разрядный ярлык (определяемый Microsoft), указывающий, какой фильтр файловой системы должен быть извещен о попытке получения доступа к данной папке или файлу. Фильтр выполняет заранее определенный код, предназначенный для управления процессом доступа. Поскольку размер данных атрибута точки повторной обработки может достигать 16 Кбайт, помимо ярлыка в атрибуте можно сохранить информацию, имеющую значение для соответствующего фильтра. Фильтр файловой системы может полностью изменить способ отображения данных файла. Поэтому фильтры устанавливаются только администраторами системы. Если по каким-либо причинам система не может найти фильтр, соответствующий определенному ярлыку повторной обработки, доступ к папке или файлу не будет предоставлен, однако они могут быть удалены. Точки повторной обработки используются при создании соединений папок NTFS, позволяющих перенаправлять запрос к папке или файлу в другое место файловой системы.

Точки соединения NTFS. Точки соединения NTFS (junction point) представляют собой новое средство, позволяющее отображать целевую папку в пустую, находящуюся в пространстве имен файловой системы NTFS 5.0 локального компьютера. Целевой папкой может служить любой допустимый путь Windows. Точки соединения поддерживаются только в NTFS 5.0 и служат для создания общего пространства имен хранения информации. Точки соединения прозрачны для приложений. Исключение составляет случай, когда информация об определенной точке соединения необходима программе для работы. Прозрачность означает, что приложение или пользователь, осуществляющий доступ к локальной папке NTFS, автоматически перенаправляются к другой папке. Доступ к локальным томам файловой системы, подключенным с помощью точки соединения, может быть получен даже в случае, если томам не присвоены имена.

Шифрование данных. Шифрование обеспечивается дополнительным компонентом операционной системы «Шифрованной файловой системой» (Encrypted File System, EFS), который представляет собой серьезный механизм защиты данных, поскольку зашифрованные данные могут быть доступны только пользователю, имеющему специальный ключ для расшифровки. EFS обеспечивает следующие функции:

- прозрачное шифрование – шифрование/расшифровка происходят прозрачно при чтении или записи файла на диск и не требуют от владельца файла расшифровывать/зашифровывать файл при каждом к нему обращении;
- защита ключей шифрования – в EFS ключи, используемые для шифрования файла, зашифрованы наиболее эффективным открытым ключом сертификата пользователя (стандарт X.509 v3), который хранится вместе со списком зашифрованных уникальных ключей, использованных для шифрования файла; для расшифровки этих ключей владелец файла использует свой закрытый ключ;
- восстановление данных – если закрытый ключ владельца не доступен, агент восстановления откроет файл своим закрытым ключом; в системе может быть несколько агентов

восстановления, каждый со своим открытым ключом, но для восстановления файла, при его шифровании должен существовать и использоваться минимум один открытый ключ восстановления;

- безопасные временные и страничные файлы – в связи с тем, что многие приложения в процессе редактирования документов создают временные файлы, система EFS шифрует временные копии зашифрованного файла; EFS располагается в ядре операционной системы Windows и хранит ключи шифрования в невыгружаемом пуле, что позволяет предотвратить их копирование в страничный файл.

Теневые копии. Служба теневого копирования реализуется только на томах NTFS-формата и позволяет создавать копии томов по расписанию, она создает мгновенные снимки состояния томов, обеспечивая архивацию файлов. Благодаря такой технологии пользователь быстро восстанавливать удаленные файлы или старые версии файлов.

Windows XP содержит утилиту CONVERT.EXE, которая преобразует тома FAT или FAT32 в эквивалентные тома NTFS. Также преобразовать файловую систему в NTFS можно при установке Windows XP, положительно ответив на вопрос о преобразовании в процессе установки.

3. Файловая система CDFS

В Windows XP, Windows 2000, Windows 2003 поддерживается файловая система компакт-дисков CDFS, выполненная по стандарту ISO 9660. Файловая система CD-ROM (CDFS) является относительно простым форматом, который был определен в 1988 году как стандарт форматирования только для чтения для носителей информации на CD-ROM. Поддержка CDFS была введена в Windows NT 4.0. Реализация в Windows включает в себя поддержку длинного имени файла, определяемую Уровнем 2 стандарта Международной организации по стандартизации (ISO) 9660. Файловая система CDFS (только для чтения), обслуживается драйвером \Windows\System32\Drivers\Cdfs.sys, который поддерживает надмножества форматов ISO-9660 и Joliet. Если формат ISO-9660 сравнительно прост и имеет ряд ограничений, то формат Joliet более гибок и поддерживает Unicode-имена произвольной длины. Если на диске присутствуют структуры для обоих форматов (чтобы обеспечить максимальную совместимость), CDFS использует формат Joliet. Из-за своей простоты формат CDFS имеет ряд ограничений:

- длина имени каталога и файла должны быть меньше чем 32 символа;
- глубина дерева каталогов может быть не больше, чем восемь уровней;
- максимальная длина файлов не должна превышать 4 Гб;
- число каталогов не может превышать 65 535.

CDFS считается унаследованным форматом, поскольку индустрия уже приняла в качестве стандарта для носителей, предназначенных только для чтения, универсальный дисковый формат UDF (Universal Disk Format).

4. Файловая система UDF

UDF – универсальный формат дисков представляет собой файловую систему, соответствующую стандарту Международной Организации по Стандартизации (ISO) 13346, предназначенную для доступа к DVD-ROM и CD-ROM. Универсальный дисковый формат (UDF) определяется Ассоциацией по технологии оптического хранения данных (OSTA). Он разработан, чтобы заменить CDFS и добавить поддержку устройств DVD-ROM. UDF включается в спецификации DVD и более гибок, чем формат CDFS. Поддержка формата UDF была введена в Windows 2000 и включала в себя поддержку только чтения дисков. Начиная с Windows XP, осуществляется поддержка чтения и записи данных на диск. В Windows XP/Windows 2003 включена собственная поддержка чтения-записи оптических дисков DVD-RAM и возможность чтения формата UDF 2.01 (Universal Disk Format), включая DVD-диски и DVD-видео. Файловая система UDF имеет ниже перечисленные особенности:

- длина имени файла может быть до 254 символов в ASCII-кодировке или до 127 символов в

Unicode-кодировке;

- имена файлов могут включать буквы как верхнего, так и нижнего регистра;
- файлы могут быть разреженными (sparse); размеры файлов задаются 64-битными значениями;
- максимальная длина пути составляет 1023 символа.

Файловая система UDF в Windows является UDF-совместимой реализацией OSTA – подмножеством формата ISO-13346 с расширениями для поддержки CD-R, DVD-R/RW и т.д. Организации по Стандартизации определила UDF в 1995 году как формат магнитооптических носителей, главным образом DVD-ROM, предназначенный для замены формата ISO-9660. Формат UDF включен в спецификацию DVD и более гибок, чем CDFS. Драйвер UDF (\Windows\System32\Drivers\Udfs.sys) поддерживает UDF версии 1.02 и 1.5 в Windows 2000 и версий 2.0 и 2.01 в Windows XP и Windows Server 2003.

5. Файловая система DFS

DFS (Distributed File System) – распределенная файловая система, которая позволяет объединить серверы и предоставляемые в общее пользование ресурсы в более простое пространство имен. Файловые системы обеспечивают однородный поименованный доступ к набору секторов на дисках, а DFS – однородный поименованный доступ к набору серверов, совместно используемых ресурсов и файлов, организуя их в виде иерархической структуры. В свою очередь новый том DFS может быть иерархично подключен к другим совместно используемым ресурсам Windows. DFS позволяет объединить физические устройства хранения в логические элементы, что делает физическое расположение данных прозрачным как для пользователей, так и для приложений. Преимущества DFS:

- Настраиваемый иерархический вид совместно используемых сетевых ресурсов. Связывая сетевые ресурсы, администраторы могут создавать единый иерархический том, представляемый в виде одного огромного жесткого диска. Пользователи могут создавать собственные тома DFS, которые в свою очередь могут быть включены в другие тома DFS. Такая технология называется DFS-связями.

- Гибкое администрирование тома. Отдельные сетевые ресурсы, входящие в DFS, можно отключить, не воздействуя на оставшуюся часть, что позволяет администраторам управлять физическими компонентами ресурсов, не изменяя логического их представления для пользователей.

- Графические средства администрирования. Каждым корнем DFS можно управлять простым графическим инструментом, позволяющим просматривать тома, изменять их конфигурацию, устанавливать DFS-связи и управлять удаленными корнями DFS.

- Повышенная доступность данных. Несколько сетевых ресурсов, предоставленных в совместное использование только для чтения, можно объединить под одним логическим именем DFS. Если один из ресурсов становится недоступным, автоматически становится доступным альтернативный.

- Балансировка нагрузки. Несколько сетевых ресурсов, предоставленных в совместное использование только для чтения, можно объединить под одним логическим именем DFS, предоставляя тем самым ограниченную балансировку нагрузки между дисками или серверами. При доступе к такому ресурсу пользователь автоматически перенаправляется на один из составляющих том DFS.

- Прозрачность имен. Пользователи перемещаются по пространству имен, не задумываясь о физическом расположении данных. Данные физически могут быть перемещены на любой сервер, но последующее переконфигурирование DFS делает это перемещение незаметным для пользователя, так как он по-прежнему оперирует с существующим для него пространством имен DFS.

- Интеграция с моделью безопасности Windows. Не требуется дополнительной работы по обеспечению безопасности. Любой пользователь, подключенный к тому DFS, имеет доступ к ресурсам, только если обладает необходимыми правами. При этом применяется модель безопасности Windows.

- Интеграция клиента DFS с Windows XP Professional, Windows 2000 Professional и Windows 9x. Клиент DFS встроен в Windows NT Workstation, начиная с 4 версии. Эта дополнительная функциональность не влияет на требования клиента к памяти.

- Интеллектуальное кэширование на клиентской части. В том DFS можно включить сотни тысяч совместно используемых ресурсов. На клиентской стороне не делается предположений о том, к какой части данных пользователю разрешен доступ. Поэтому при первом обращении к каталогу определенная информация кэшируется локально. При повторном обращении к той же информации задействуется часть, находящаяся в кэше, и повторного поиска ссылки не происходит, что позволяет заметно повысить производительность в больших иерархических сетях.

- Взаимодействие с другими сетевыми файловыми системами. Любой том, к которому можно осуществить доступ через редиректор Windows, можно включить в пространство имен DFS. Такой доступ можно осуществлять либо через клиентские редиректоры, либо через шлюзы на сервере.

Примечание:

1.Редиректор (англ. redirector, перенаправляющий) – модуль в прокси-серверах, отвечающий за фильтрацию и обработку адресов (URL²) запросов от клиентов к серверам. Может быть встроенным в прокси-сервер или запускающийся отдельным приложением (скриптом). Задачи, решаемые с помощью редиректора:

- закрытие доступа к определенным адресам по сложным критериям;
- замена одного содержимого на другое (например, баннеров на пустые изображения);
- выдача сообщения о точной причине запрета доступа к странице;
- выдача предупреждения о возможной фишинг-атаке (при наличии фишинг-фильтра);
- анализ статистики обращения к определенным ресурсам (как разрешенным, так и запрещенным)

Часть задач редиректора может быть решена при помощи списков контроля доступа прокси-сервера. Однако использование редиректора позволяет упростить конфигурирование в случае сложной схемы разрешений и запретов (разные наборы сайтов для разных групп пользователей).

2.Прокси-сервер (англ. проху, представитель, уполномоченный) – служба в компьютерных сетях, позволяющая клиентам выполнять косвенные запросы к другим сетевым службам. Сначала клиент подключается к прокси-серверу и запрашивает какой-либо ресурс (например, e-mail), расположенный на другом сервере. Затем прокси-сервер либо подключается к указанному серверу и получает ресурс у него, либо возвращает ресурс из собственного кеша (в случаях, если прокси-сервер имеет свой кеш). В некоторых случаях запрос клиента или ответ сервера может быть изменен прокси-сервером в определенных целях. Также прокси-сервер позволяет защищать клиентский компьютер от некоторых сетевых атак.

Чаще всего прокси-серверы применяются для следующих целей:

- обеспечение доступа с компьютеров локальной сети в Интернет;
- кеширование данных: если часто происходят обращения к одним и тем же внешним ресурсам, то можно держать их копию на прокси-сервере и выдавать по запросу, снижая нагрузку на канал во внешнюю сеть и ускоряя получение клиентом запрошенной информации;
- сжатие данных: прокси-сервер загружает информацию из Интернета и передает информацию конечному пользователю в сжатом виде с целью экономии внешнего трафика;
- защита локальной сети от внешнего доступа: например, можно настроить прокси-сервер так, что локальные компьютеры будут обращаться к внешним ресурсам только через него, а внешние компьютеры не смогут обращаться к локальным вообще (они «видят» только прокси-сервер);
- ограничение доступа из локальной сети к внешней: например, можно запретить доступ к определенным веб-сайтам, ограничить использование Интернета локальным пользователям, устанавливать квоты на трафик или полосу пропускания, фильтровать рекламу и вирусы;

² URL (*Uniform (Universal) Resource Locator*) – единый (универсальный) указатель (определитель местонахождения) ресурсов, который представляет собой стандартизированный способ записи адреса ресурса в сети Интернет.

- анонимизация доступа к различным ресурсам: прокси-сервер может скрывать сведения об источнике запроса или пользователе, в таком случае целевой сервер видит лишь информацию о прокси-сервере и не имеет возможности определить истинный источник запроса.

Многие прокси-серверы используются для нескольких целей одновременно, некоторые – ограничивают работу несколькими портами: 80 (HTTP), 443 (Шифрованное соединение HTTPS), 20,21 (FTP). В отличие от шлюза, прокси-сервер чаще всего не пропускает ICMP-трафик (невозможно проверить доступность машины командами ping и tracert).

3. Фишинг (англ. phishing – выуживание) – вид Интернет-мошенничества, целью которого является получение доступа к конфиденциальным данным пользователей (логинам и паролям). Фишинг достигается путем проведения массовых рассылок электронных писем от имени популярных брендов, (например, от имени социальных сетей, банков и прочих сервисов). В письме содержится прямая ссылка на сайт, оказавшись на котором, пользователь может сообщить мошенникам ценную информацию, позволяющую получить доступ к аккаунтам³ и банковским счетам.

Фишинг – одна из разновидностей социальной инженерии, основанная на незнании пользователями основ сетевой безопасности, например, сервисы не рассылают писем с просьбами сообщить свои учетные данные, пароль и прочее. Для защиты от фишинга производители основных Интернет-браузеров договорились о применении одинаковых способов информирования пользователей о том, что они открыли подозрительный сайт, который может принадлежать мошенникам. Новые версии браузеров уже обладают такой возможностью, которая соответственно именуется «антифишинг».

2. Практическая часть

2.1. Вопросы по разделу

1. Дайте определение термину «файловая система».
2. Определите назначение файловой системы.
3. Объясните, каким образом размещается и передается информация на магнитных дисках.
4. Дайте определение термину «сектор» и поясните из каких элементов он состоит.
5. Дайте определение термину «цилиндр».
6. Как определяется физический адрес сектора?
7. Как осуществляется обмен между ОЗУ и дисками?
8. Определите типы разделов и их использование.
9. С какими файловыми системами поддерживает работу операционная система Windows XP?
10. В каком элементе архитектуры операционной системы Windows XP реализуется возможность поддержки различных файловых систем?
11. Что собой представляет FAT?
12. Что определяет элемент FAT (перечислите области логического дискового пространства и дайте им определение)?
13. Из каких элементов создается системная область диска и как она создается?
14. Дайте определение термину «кластер».
15. Чем отличается корневой каталог от обычного каталога?
16. Перечислите атрибуты файла.
17. Перечислите отличие FAT12 от FAT16.
18. Перечислите отличие FAT16 от FAT32.
19. В каких случаях используется файловая система FAT32?
20. Перечислите отличие NTFS от FAT32.

³ Аккаунт (англ. account) – учетная запись, содержащая сведения, которые пользователь сообщает о себе некоторой компьютерной системе.

21. Определите термин «NTFS».
22. Определите важнейшее свойство файловой системы NTFS. Как это свойство реализуется.
23. Перечислите дополнительные возможности NTFS по сравнению с FAT32.
24. Каким образом реализуется защита файлов в NTFS?
25. Каким образом обеспечивается сжатие файлов и каталогов в NTFS?
26. Объясните значение понятия «многопоточные файлы» в NTFS.
27. Объясните назначение службы отслеживания связей в NTFS.
28. Объясните значение понятия «квоты дискового пространства» в NTFS.
29. Объясните значение понятия «точки повторной обработки» в NTFS.
30. Каким образом реализуется шифрование файлов в NTFS?
31. Перечислите функции системы EFS в NTFS.
32. Объясните значение понятия «теневые копии» в NTFS.
33. Что произойдет, если файл будет скопирован из раздела или тома NTFS в раздел или на том FAT?
34. Определите назначение и ограничения файловой системы CDFS.
35. Определите назначение и ограничения файловой системы UDF.
36. Определите назначение и преимущества файловой системы DFS.
37. Определите термин «редиректор».
38. Определите термин «прокси-сервер».
39. Определите термин «фишинг».
40. Что представляет собой файловая система MS-DOS?
41. Определите термин «аккаунт».
42. Определите термин «дескриптор файла».
43. Определите формат каталоговой записи в системе MS-DOS.
44. Определите формат каталоговой записи в системе Windows 98.
45. Определите формат каталоговой записи с фрагментом длинного имени файла в Windows 98.
46. Каким образом реализуется отличие каталоговых записей, содержащих длинные и короткие имена файла в Windows 98?
47. Определите тип и назначение операционной системы UNIX.
48. Какую файловую структуру имеет операционная система UNIX?
49. Что фактически представляет имя файла в операционной системе UNIX?
50. С чего начинается работа пользователя в операционной системе UNIX?
51. Определите термин «аутентичность пользователя».
52. В соответствии с какой информацией реализуется аутентичность пользователя?
53. Перечислите права пользователя UNIX?
54. К каким группам относятся файлы операционной системы UNIX? Дать определение каждой из групп.
55. Перечислите файловые системы, поддерживаемые операционной системой Linux.
56. Определите отличие файловой системы Ext2fs от файловой системы Ext3fs и ReiserFS.
57. С помощью какого механизма в Linux и UNIX решается проблема защиты целостности файловой системы при некорректном завершении работы?
58. Перечислите особенности файловой системы XFS.
59. Определите назначение виртуальных файловых систем.
60. Перечислите виртуальные файловые системы, их назначение и отличие.

Упражнение 1. Преобразование файловой системы FAT16 или FAT32 в NTFS с помощью Windows XP

Список ограничений и требований:

1. Файловые системы UDF и CDFS применяются только на оптических носителях и не могут

быть преобразованы в NTFS.

2. На гибких дисках применяется только файловая система FAT12.

3. После преобразования файловой системы FAT32 в NTFS возможно ухудшение быстродействия некоторых более ранних программ, которые разрабатывались без учета особенностей Windows NT 4.0 и Windows 2000. Эта проблема не возникает в случае форматирования чистого раздела.

4. Преобразовать файловую систему FAT или FAT32 в NTFS можно с помощью команды Convert.exe. При этом (в отличие от операции форматирования) существующие в разделе файлы сохраняются в неповрежденном виде.

5. Преобразование файловой системы является необратимым процессом, то есть если диск или раздел преобразован в формат NTFS, его нельзя преобразовать обратно в формат FAT16 или FAT32. Чтобы восстановить предыдущую файловую систему, раздел необходимо снова отформатировать с помощью файловой системы FAT16 или FAT32. При этом все существующие в разделе данные, включая программы и личные файлы, будут удалены. Необходимо восстановить резервную копию данных или переустановить операционную систему и программы.

6. Чтобы преобразовать файловую систему на диске или разделе с помощью программы Convert.exe, там должен иметься определенный объем свободного пространства. В противном случае преобразование выполнено не будет.

7. Если помимо Windows XP на компьютере установлены операционные системы Windows других версий следует учитывать:

- только Windows 2000 и Windows XP имеют полный доступ к файлам в разделе NTFS;
- Windows NT 4.0 с пакетом обновления 4 (SP4) или более поздних версий может получать доступ к файлам в разделе NTFS, однако существуют ограничения в отношении файлов, сохраненных с помощью функций, которые были введены в более поздних версиях NTFS;
- Windows Millennium Edition, Windows 98 второго издания, операционные системы Windows более ранних версий, а также MS-DOS не поддерживают разделов с файловой системой NTFS.

8. Несмотря на то, что вероятность повреждения или потери данных в процессе преобразования невысока, рекомендуется создать резервную копию данных, расположенных в подлежащем преобразованию разделе.

Для преобразования файловой системы FAT16 или FAT32 в NTFS следует:

- нажать кнопку «Пуск», выбрать пункт «Все программы», выбрать пункт «Стандартные» и выбрать пункт «Командная строка»;
- в командной строке набрать команду, где «буква_диска» – означает диск, подлежащий преобразованию:

`convertбуква_диска: /fs:ntfs`

например, для преобразования в формат NTFS диска E: служит следующая команда:

`convert e: /fs:ntfs`

- в случае если на подлежащем преобразованию диске установлена операционная система, появится предложение отложить выполнение задания до следующей перезагрузки компьютера, поскольку преобразование невозможно, когда операционная система запущена;

- нажать кнопку «Да»;
- в командной строке появится сообщение: «Тип файловой системы: FAT»;
- ввести метку тома для диска (буква_диска);
- после завершения преобразования в командной строке появится сообщение: «Преобразование завершено»;
- закрыть окно командной строки.

Для получения дополнительных сведений о программе Convert.exe и просмотра списка параметров командной строки следует:

- нажать кнопку «Пуск», выбрать пункт «Все программы», выбрать пункт «Стандартные» и выбрать пункт «Командная строка»;
- в командной строке ввести «help convert»;
- нажать клавишу «ВВОД»;
- появится список параметров командной строки, поддерживаемых программой Convert.exe.

Устранение неполадок:

1.Если при попытке преобразования тома в формат NTFS в командной строке появится сообщение об ошибке: «Не удастся получить монопольный доступ к диску буква_диска, поэтому он не будет преобразован. Выполнить его преобразование автоматически при следующей перезагрузке системы?». Такая ситуация возникает, если преобразуемый диск в данный момент времени используется. Можно в командной строке ввести символ «Y». Диск или раздел будет преобразован в формат NTFS при следующей перезагрузке компьютера.

2.Если при попытке преобразования тома в формат NTFS в командной строке появляется сообщение об ошибке: «Этот том используется другим процессом, и выполнить команду Convert для него невозможно. Чтобы запустить Convert, вначале следует отключить этот том. Все открытые дескрипторы тома будут далее неверны. Хотите отключить том?». Такая ситуация возникает, если в подлежащем преобразованию разделе в данный момент используются некоторые файлы, в том числе файлы, к которым пользователи подключены по сети. Для разрешения этой ситуации следует закрыть все программы, которые используют файлы из данного раздела и ввести в командной строке символ «Y», чтобы начать преобразование.

3.Если при попытке преобразования тома в формат NTFS в командной строке появляется сообщение об ошибке: «Не удастся получить монопольный доступ к диску буква_диска, поэтому он не будет преобразован. Выполнить его преобразование автоматически при следующей перезагрузке системы?». Для разрешения этой ситуации в командной строке следует ввести символ «Y». Раздел или диск будет преобразован в формат NTFS при следующей перезагрузке компьютера.

Список литературы, рекомендуемый к использованию по данной теме:

Основная: 1-4

Дополнительная: 1-3

Интернет-ресурсы: 1

Методическая: 1-3

Программное обеспечение: 1-4

Практическая работа №2

Тема: Антивирусная защита Протоколирование и аудит

1. Подготовка к занятию

1. Изучить (повторить) теоретический материал,.
2. Ознакомиться с программой работы.
3. Подготовить отчет о работе.
4. Ответить на контрольные вопросы.

2. Распределение времени занятия:

Всего: 90 мин

Вступительная часть 2 мин

Проверка готовности студентов к занятию 5 мин

Основная часть – 70 мин

Проверка выполнения практического занятия 10 мин

Заключительная часть 3 мин

3. Формируемые компетенции: ПК-1, ПК-3

4. Правила работы в лаборатории

К работе в лаборатории допускаются лица, изучившие правила и меры безопасности, сдавшие зачет по ним и усвоившие порядок выполнения лабораторной работы.

4.1. Требования безопасности перед началом работ

- ЗАПРЕЩАЕТСЯ: переодеваться, пользоваться огнем, курить, принимать пищу в лаборатории.

- Убедиться в целостности электрических розеток и разъемов. В лаборатории необходимо быть в сменной обуви.

- Включение компьютера производить только после получения допуска по выполняемой работе и разрешения преподавателя или лаборанта.

4.2. Требования безопасности во время работы

- выполняя практическое занятие, студенты обязаны использовать только вычислительную технику, периферийное оборудование, соединительные кабели, измерительное оборудование и носители информации, непосредственно относящиеся к данному лабораторному занятию;

- подключение и отключение составляющих вычислительного комплекса производить только при полном снятии напряжения со всех составляющих вычислительного комплекса;

- при обнаружении неисправностей в оборудовании немедленно отключить источники питания и доложить об этом руководителю занятий или лаборанту.

4.3. Требования безопасности по окончании работы

- доложить руководителю занятий или лаборанту о завершении работ;

- привести в порядок и сдать рабочее место лаборанту, и доложить руководству

Файловая система операционной системы UNIX

Операционная система UNIX представляет собой ядро многопользовательской операционной системы с разделением времени. Она дает пользователям возможность запускать свои программы, управляет периферийными устройствами и обеспечивает работу файловой системы.

Работу операционной системы UNIX можно представить в виде функционирования множества взаимосвязанных процессов. При загрузке системы сначала запускается ядро (процесс 0), которое в свою очередь запускает командный интерпретатор shell (процесс 1). Взаимодействие пользователя с системой UNIX происходит в интерактивном режиме посредством командного языка. Оболочка операционной системы shell интерпретирует вводимые команды, запускает соответствующие программы (процессы), формирует и выводит ответные сообщения.

Важной составной частью UNIX является файловая система, которая является сложной многопользовательской системой, так как в основе этой системы лежала операционная система MULTICS. Файловая система имеет иерархическую структуру, образующую дерево каталогов и файлов. Дерево начинается в корневом каталоге, с добавлением связей, формирующих направленный ациклический граф. Имена файлов могут содержать до 14 символов, включающих в себя любые символы ASCII, кроме косой черты (использовавшейся в качестве разделителя компонентов пути) и символа NUL (использовавшегося для дополнения имен короче 14 символов). Символ NUL обозначается байтом 0.

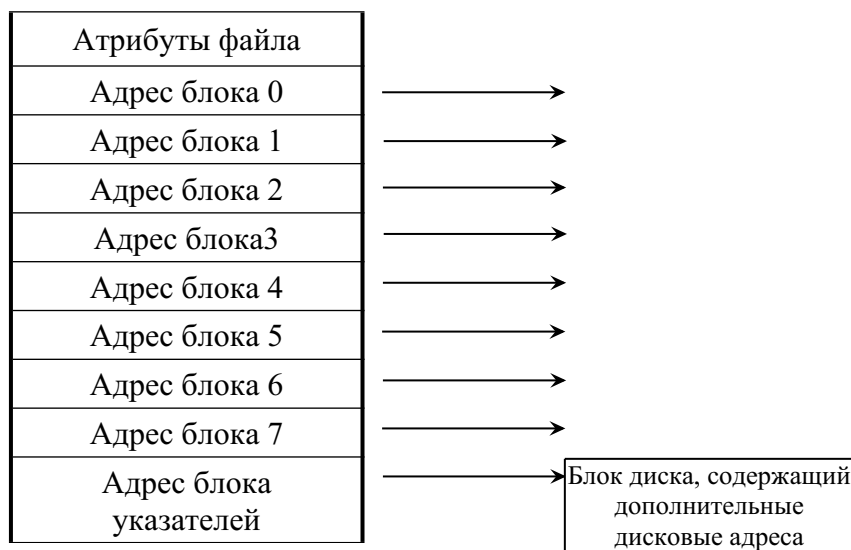


Рис. 5. Пример *i*-того узла

Корневой каталог обозначается символом «/», путь по дереву каталогов состоит из имен каталогов, разделенных символом «/», например: /usr/include/sys. В каждый момент времени с любым пользователем связан текущий каталог, то есть местоположение пользователя в иерархической файловой системе. Каталог UNIX содержит по одной записи для каждого файла этого каталога. Каждая каталоговая запись проста, так как в системе UNIX используется схема *i*-узлов (рис.5.). Каталогная запись состоит всего из двух полей: имени файла (14 байт) и номера *i*-узла для этого файла (рис.6.).

Каждый файл операционной системы UNIX может быть однозначно определен некоторой структурой данных, называемой описателем файла (дескриптором). Он содержит всю информацию о файле: тип файла, режим доступа, идентификатор владельца, размер, адрес файла, даты последнего доступа и последней модификации, дату создания и пр.

Обращение к файлу происходит по имени. Путь к файлу от корневого каталога называется

разницей, что алгоритм начинает работу не с корневого, а с рабочего каталога.

Работа пользователя в системе начинается с того, что активизируется сервер терминального доступа `getty` (программа, организующая диалог работы с пользователем в многопользовательской операционной системе), который запускает программу `login`, запрашивающую у пользователя имя и пароль.

Далее происходит проверка аутентичности пользователя (подлинности регистрации пользователя в системе) в соответствии с той информацией, которая хранится в файле `/etc/passwd`. В этом файле хранятся записи, содержащие:

- регистрационное имя пользователя;
- зашифрованный пароль;
- идентификатор пользователя;
- идентификатор группы;
- информация о минимальном сроке действия пароля;
- общая информация о пользователе;
- начальный каталог пользователя;
- регистрационный `shell` пользователя.

Если пользователь зарегистрирован в системе и ввел правильный пароль, `login` запускает программу, указанную в `/etc/passwd` – регистрационный `shell` пользователя.

Пользователь системы – это объект, обладающий определенными правами, определяющими возможность запуска программ на выполнение, а также владение файлами. Единственный пользователь системы, обладающий неограниченными правами – это суперпользователь или администратор системы.

Система идентифицирует пользователей по идентификатору пользователя (`UID` – `User Identifier`). Каждый пользователь является членом одной или нескольких групп – списка пользователей, имеющих сходные задачи. Каждая группа имеет свой уникальный идентификатор группы (`GID` – `Group Identifier`). Принадлежность группе определяет совокупность прав, которыми обладают члены данной группы.

Права пользователя UNIX – это права на работу с файлами. Файлы имеют двух владельцев: пользователя (`user owner`) и группу (`group owner`). Соответственно, атрибуты защиты файлов определяют права пользователя-владельца файла, права члена группы-владельца (`g`) и права всех остальных (`o`). В каждый момент времени с любым пользователем связан текущий каталог, то есть местоположение пользователя в иерархической файловой системе.

Всякий файл операционной системы UNIX в соответствии с его типом может быть отнесен к одной из следующих групп: обычные файлы, каталоги, специальные файлы и каналы.

Обычный файл представляет собой последовательность байтов. Никаких ограничений на файл системой не накладывается, и никакого смысла не приписывается его содержимому: смысл байтов зависит исключительно от программ, обрабатывающих файл.

Каталог – это файл особого типа, отличающийся от обычного файла наличием структуры и ограничением по записи: осуществить запись в каталог может только ядро операционной системы UNIX. Каталог устанавливает соответствие между файлами (номерах описателей) и их локальными именами.

Специальный файл – это файл, поставленный в соответствие некоторому внешнему устройству и имеющий специальную структуру. Его нельзя использовать для хранения данных как обычный файл или каталог, но над ним можно производить те же операции, что и над любым другим. При этом ввод/вывод информации в этот файл будет соответствовать вводу с внешнего устройства или выводу на него.

Канал – это программное средство, связывающее процессы операционной системы UNIX буфером ввода/вывода. Например, запуск процессов в виде `$процесс_1|процесс_2` означает, что стандартный вывод процесса_1 будет замкнут на стандартный ввод процесса_2. При этом сначала создается канал, а потом на выполнение одновременно запускаются оба процесса, и общее время их выполнения определяется более медленным процессом.

1.5. Файловые системы операционной системы Linux

В Linux долгое время была одна файловая система Ext2fs – вторая расширенная файловая система. Система определяется как расширенная по сравнению с файловой системой операционной системы Minix, послужившей прототипом Linux (до сих пор используемой на отформатированных в этой операционной системе дисках). Вторая – означает, что ранние версии Linux базировались на Extfs с более ограниченными возможностями.

По способу организации хранения данных Extfs напоминает файловую систему Unix. Отличительные особенности:

- дробление дискового раздела на группы блоков;
- наличие нескольких копий суперблока, что повышает надежность хранения данных;
- наличие эффективного механизма кэширования дисковых операций, что обеспечивает их быстрое действие;
- относительно слабая устойчивость при аварийном завершении работы (вследствие мертвого зависания или отказа питания).

Проблема нарушения целостности файловой системы при некорректном завершении работы характерна и для всех операционных систем семейства Unix. Потому для реализации механизма восстановления стали разрабатывать журналируемые файловые системы. Журнал представляет собой log-файл дисковых операций, в котором фиксируются не выполненные, а только предстоящие манипуляции с файлами. Журналирование направлено на обеспечение целостности файловой системы, но не гарантирует сохранности пользовательских данных.

В большинстве журналируемых файловых систем фиксируются будущие операции только над метаданными изменяемых файлов, что достаточно для сохранения целостности файловой системы и предотвращения долговременных проверок, но не предотвращает потери данных в аварийных ситуациях. В некоторых файловых системах журналирование распространяется и на область данных файла, однако повышение надежности снижает быстрое действие системы.

Текущие версии ядра Linux поддерживают в качестве альтернативных четыре журналируемые файловые системы: ReiserFS, Ext3fs и XFS, JFS (результаты импортирования в Linux файловых систем, разработанных первоначально для рабочих станций под операционные системы Irix (SGI) и AIX (IBM), соответственно).

Журналируемая файловая система ReiserFS разработана специально для Linux фирмой Namesys (<http://www.namesys.com>) и поддерживается ее ядром (<http://www.kernel.org>), начиная с первых версий ветви 2.4.x. В ReiserFS осуществляется журналирование только операций над метаданными файлов, что при определенном снижении надежности, обеспечивает высокую производительность. Кроме этого, ReiserFS обладает уникальной (и по умолчанию задействованной) возможностью оптимизации дискового пространства, занимаемого мелкими, менее одного блока, файлами. Они хранятся в своих inode, без выделения блоков в области данных. Вместе с экономией места это способствует росту производительности, так как данные и метаданные (в терминах ReiserFS – stat -data) файла хранятся в непосредственной близости и могут быть считаны одной операцией ввода/вывода.

Хвосты файлов (их конечные части), меньшие по размеру, чем один блок, могут быть подвергнуты упаковке. Этот режим (tailing) включается по умолчанию при создании ReiserFS, обеспечивая около 5% экономии дискового пространства, но несколько снижает быстрое действие. ReiserFS не совместима с Ext2fs на уровне утилит обслуживания файловой системы, но соответствующий инструментальный, разрешающий проблему, объединен в пакет reiserfsprogs и включен в штатный комплект современных дистрибутивов. Распространенные загрузчики Linux иногда не способны загрузить ядро Linux с раздела ReiserFS, поэтому ReiserFS не рекомендуется к употреблению на загрузочном разделе.

Ext3fs – представляет собой журналируемую надстройку над классической Ext2fs, разработанной в компании Red Hat и поддерживаемой ядром Linux, начиная с версии 2.4.16. Она сохраняет со своей прародительницей полную совместимость, в том числе и на уровне утилит

обслуживания (начиная с версии 1.21, пакета e2fsprogs). Переход от Ext2fs к Ext3fs осуществляется добавлением файла журнала без переформатирования раздела и рестарта машины. Ext3fs является системой, в которой возможно журналирование операций не только с метаданными, но и с данными файлов, так как предусмотрено три режима работы: полное журналирование (full data journaling); журналирование с обратной записью (writeback); последовательное журналирование, задеиствуемое по умолчанию (ordered).

Режим полного журналирования распространяется на метаданные и на данные файлов. Все их изменения сначала пишутся в файл журнала и только после этого фиксируются на диске. В случае аварийного отказа журнал можно повторно перечитать, приведя данные и метаданные в непротиворечивое состояние. Данный механизм гарантирует от потерь данных, однако является медленным.

В режиме отложенной записи в файл журнала записываются только изменения метаданных файлов и нет гарантии сохранности данных, однако обеспечивается наибольшее быстродействие. В последовательном режиме также физически журналируются только метаданные файлов, но связанные с ними блоки данных логически группируются в единый модуль (транзакцию, transaction) и записываются перед записью на диск новых метаданных, что способствует сохранности данных. Данный режим при меньших накладных расходах по сравнению с полным журналированием, обеспечивает промежуточный уровень быстродействия.

Файловая 64-разрядная система XFS развивается фирмой SGI для Unix, впервые появилась в версии Irix 5.3, вышедшей в 1994 г. В Linux она была импортирована недавно (<http://oss.sgi.com/projects/xfs>) и штатно поддерживается ядром, начиная с его ветки 2.6.X. XFS представляет собой сбалансированную файловую систему, ориентированную на размещение в больших дисковых разделах для работы с большими файлами. Особенности XFS:

- использование механизма деления единого дискового раздела на несколько равных областей (allocation group), имеющих собственные списки inodes и свободные блоки, для распараллеливания дисковых операций (самостоятельные файловые подсистемы);
- использование логического журналирования только для изменений метаданных, но с частым сбросом их на диск для минимизации возможных потерь при сбоях;
- применение механизма ассигнования дискового пространства при записи файлов не во время журналирования, а при фактическом сбросе их на диск (delayed allocation), что вместе с повышением производительности предотвращает фрагментацию дискового раздела;
- использование списков контроля доступа (ACL, Access Control List) и расширенных атрибутов файлов (extended attributes).

Возможность работы с XFS обеспечивает специальный патч (xfs-2.4.1X-all-i386.bz2), который вместе с соответствующими утилитами поддержки можно получить с сайта SGI (<http://oss.sgi.com/projects/xfs>). Утилиты поддержки для XFS объединены в несколько пакетов.

Файловая система JFS разработана компанией IBM для собственной версии Unix и уже долгое время поддерживается ядром Linux в качестве альтернативной системы, однако по ряду причин широкого распространения она не получила.

На уровне обмена данными Linux поддерживает множество файловых систем, некоторые из них только в режиме чтения (например, NTFS или HPFS).

Все упомянутые выше файловые системы могут располагаться не только на реальных блочных устройствах (дисках и дисковых разделах), но и на виртуальных дисках (RAM-дисках).

1.6. Виртуальные файловые системы

Операционные системы POSIX-совместимые поддерживают еще и несколько типов виртуальных файловых систем, которые располагаются в оперативной памяти и служат для специальных целей. Первой виртуальной файловой системой была система процессов – procfs, которая представляла протекающие в системе процессы в виде файлов каталога /proc, откуда получали информацию о процессах команды типа ps и top.

В каталоге /proc каждому процессу соответствует подкаталог, именем которого является

идентификатор процесса (номер в порядке запуска). Внутри такого подкаталога находится набор файлов, содержимое некоторых из них можно посмотреть. Важная информация содержится в файлах корня `/proc`. Используя команды просмотра, можно посмотреть следующую информацию: о процессоре (`/proc/cpuinfo`); о текущей конфигурации ядра системы (`/proc/config.gz`); о загруженных его модулях (`/proc/modules`), об устройствах, подсоединенных к шине PCI (`/proc/pci`), сведения о сетевой карте и внутреннем модеме, необходимые для правильной их настройки.

Другой виртуальной файловой системой является файловая система устройств – `devfs`, которая по умолчанию задействована во FreeBSD 5-й ветки и во многих современных дистрибутивах Linux. Благодаря `devfs`, файлы устройств стали создаваться автоматически при старте системы (в соответствии с устройствами, поддерживаемыми текущей конфигурацией ядра и определяемыми в ходе загрузки операционной системы). Если в ходе работы не окажется какого-либо устройства, из числа поддерживаемых, то автоматически будет создан файл соответствующий этому устройству. Файловая система `devfs` сделала очень легким подключение устройств типа карт PCMCIA, USB-накопителей, цифровых камер и сканеров. Достаточно воткнуть USB-драйв в соответствующий разъем и в каталоге `/dev` появится соответствующий ему файл.

В настоящее время в Linux файловая система устройств постепенно заменяется системой поддержки динамического именования устройств – `udev`. В отличие от `devfs`, `udev` – пользовательская программа (входящая в одноименный пакет) и при ее использовании необходимость в поддержке `devfs` отпадает. Для своего функционирования `udev` нуждается в виртуальной файловой системе – `sysfs`, но ее поддержка в ядрах серии 3.6.X осуществляется автоматически (а сама эта файловая система монтируется по умолчанию в каталог `/sys`). `Udev` присваивает имена всем устройствам, в том числе и при их подключении.

Любой POSIX-системе имена файлов конкретных устройств более или менее безразличны, так как оперирует она не с ними, а с их идентификаторами. Ранее в качестве таковых выступали старший и младший номера устройства, определяющие их класс и конкретный его экземпляр. В `udev` стали использоваться идентификаторы устройств (серийный номер винчестера, его положение на канале IDE-шины и т.д.), сочетание которых для каждого диска (раздела) уникально. Программа `udev` извлекает эти сведения из файловой системы `sysfs` и, руководствуясь определенными правилами, ставит им в соответствие имена (например, `/dev/hda`).

Во FreeBSD и многих дистрибутивах Linux используется также файловая система в оперативной памяти, именуемая `mfs` в первом случае и `tmpfs` – во втором случае. Системы реализованы по-разному, но с точки зрения пользователя выглядят одинаково, как дисковые разделы, смонтированные в некоторые каталоги. Они заменяют собой блочные устройства там, где требуется быстрая, но не обязательно долговременная запись. Эти системы целесообразно использовать для промежуточных каталогов при архивации, разархивации, пакетной конвертации графических файлов, компиляции программ.

В отличие от стандартных файловых систем для хранения данных `mfs` и `tmpfs` не нуждаются в своем создании. Для их использования достаточно факта поддержки ядром (непосредственно или в виде модуля) и монтирования в какой-либо каталог. Обычно таким каталогом выступает `/tmp`, содержимое которого не должно сохраняться после рестарта системы. В BSD-системах в некоторых случаях целесообразно монтирование `mfs` в каталог `/usr/obj`, предназначенный для промежуточных продуктов компиляции при сборке ядра и базовых компонентов. В Linux одно из штатных мест монтирования `tmpfs` – каталог `/dev/shm`.

Список литературы, рекомендуемый к использованию по данной теме:

Основная: 1-4

Дополнительная: 1-3

Интернет-ресурсы: 1

Методическая: 1-3

Программное обеспечение: 1-4

Практическая работа №3

Тема: Системы безопасности ОС Linux Системы безопасности ОС Windows

1. Подготовка к занятию

1. Изучить (повторить) теоретический материал,.
2. Ознакомиться с программой работы.
3. Подготовить отчет о работе.
4. Ответить на контрольные вопросы.

2. Распределение времени занятия:

Всего: 90 мин

Вступительная часть 2 мин

Проверка готовности студентов к занятию 5 мин

Основная часть – 70 мин

Проверка выполнения практического занятия 10 мин

Заключительная часть 3 мин

3. Формируемые компетенции: ПК-1, ПК-3

4. Правила работы в лаборатории

К работе в лаборатории допускаются лица, изучившие правила и меры безопасности, сдавшие зачет по ним и усвоившие порядок выполнения лабораторной работы.

4.1. Требования безопасности перед началом работ

- ЗАПРЕЩАЕТСЯ: переодеваться, пользоваться огнем, курить, принимать пищу в лаборатории.

- Убедиться в целостности электрических розеток и разъемов. В лаборатории необходимо быть в сменной обуви.

- Включение компьютера производить только после получения допуска по выполняемой работе и разрешения преподавателя или лаборанта.

4.2. Требования безопасности во время работы

- выполняя практическое занятие, студенты обязаны использовать только вычислительную технику, периферийное оборудование, соединительные кабели, измерительное оборудование и носители информации, непосредственно относящиеся к данному лабораторному занятию;

- подключение и отключение составляющих вычислительного комплекса производить только при полном снятии напряжения со всех составляющих вычислительного комплекса;

- при обнаружении неисправностей в оборудовании немедленно отключить источники питания и доложить об этом руководителю занятий или лаборанту.

4.3. Требования безопасности по окончании работы

- доложить руководителю занятий или лаборанту о завершении работ;

- привести в порядок и сдать рабочее место лаборанту, и доложить руководству

1. Теоретические сведения

Реестр - это иерархическая централизованная база данных, используемая в ОС Microsoft Windows для хранения сведений, необходимых для настройки операционной системы для работы с пользователями, программными продуктами и устройствами.

В реестре хранятся данные, которые необходимы для правильного функционирования Windows. К ним относятся профили всех пользователей, сведения об установленном программном обеспечении и типах документов, которые могут быть созданы каждой программой, информация о свойствах папок и значках приложений, а также установленном оборудовании и используемых портах.

Системный реестр заменяет собой большинство текстовых INI-файлов, которые использовались в Windows 3.x, а также файлы конфигурации MS-DOS, такие как Autoexec.bat и Config.sys. Версии реестра для разных версий операционных систем семейства Windows имеют определенные различия.

Куст реестра - это группа разделов, подразделов и параметров реестра с набором вспомогательных файлов, содержащих резервные копии этих данных. Вспомогательные файлы для всех кустов за исключением HKEY_CURRENT_USER хранятся в системах Windows NT 4.0, Windows 2000, Windows XP, Windows Server 2003 и Windows Vista в папке %SystemRoot%\System32\Config. Вспомогательные файлы для куста HKEY_CURRENT_USER хранятся в папке %SystemRoot%\Profiles\Имя_пользователя. Расширения имен файлов в этих папках указывают на тип содержащихся в них данных. Отсутствие расширения также иногда может указывать на тип содержащихся в файле данных.

Куст реестра	Вспомогательные файлы
HKEY_LOCAL_MACHINE\SAM	Sam, Sam.log, Sam.sav
HKEY_LOCAL_MACHINE\Security	Security, Security.log, Security.sav
HKEY_LOCAL_MACHINE\Software	Software, Software.log, Software.sav
HKEY_LOCAL_MACHINE\System	System, System.alt, System.log, System.sav
HKEY_CURRENT_CONFIG	System, System.alt, System.log, System.sav, Ntuser.dat, Ntuser.dat.log
HKEY_USERS\DEFAULT	Default, Default.log, Default.sav

В Windows 98 файлы реестра называются User.dat и System.dat. В Windows Millennium Edition — Classes.dat, User.dat и System.dat.

Примечание. Средства безопасности в Windows NT, Windows 2000, Windows XP, Windows

Server 2003 и Windows Vista позволяют администратору контролировать доступ к разделам реестра.

Следующая таблица содержит перечень и краткое описание стандартных разделов. Максимальная длина имени раздела составляет 255 символов.

Папка/стандартный раздел	Описание
HKEY_CURRENT_USER	Данный раздел является корневым для данных конфигурации пользователя, вошедшего в систему в настоящий момент. Здесь хранятся папки пользователя, цвета экрана и параметры панели управления. Эти сведения сопоставлены с профилем пользователя. Вместо полного имени раздела иногда используется аббревиатура HKCU.
HKEY_USERS	Данный раздел содержит все активные загруженные профили пользователей компьютера. Раздел HKEY_CURRENT_USER является подразделом раздела HKEY_USERS. Вместо полного имени раздела иногда используется аббревиатура HKU.
HKEY_LOCAL_MACHINE	Раздел содержит параметры конфигурации, относящиеся к данному компьютеру (для всех пользователей). Вместо полного имени раздела иногда используется аббревиатура HKLM.
HKEY_CLASSES_ROOT	Является подразделом HKEY_LOCAL_MACHINE\Software. Хранящиеся здесь сведения обеспечивают выполнение необходимой программы при открытии файла с использованием проводника. Вместо полного имени раздела иногда используется аббревиатура HKCR. Начиная с Windows 2000, эти сведения хранятся как в HKEY_LOCAL_MACHINE, так и в HKEY_CURRENT_USER. Раздел HKEY_LOCAL_MACHINE\Software\Classes содержит параметры по умолчанию, которые относятся ко всем пользователям локального компьютера. Параметры, содержащиеся в разделе HKEY_CURRENT_USER\Software\Classes, переопределяют принятые по умолчанию и относятся только к текущему

пользователю. Раздел HKEY_CLASSES_ROOT включает в себя данные из обоих источников. Кроме того, раздел HKEY_CLASSES_ROOT предоставляет эти объединенные данные программам, разработанным для более ранних версий Windows. Изменения настроек текущего пользователя выполняются в разделе HKEY_CURRENT_USER\Software\Classes. Модификация параметров по умолчанию должна производиться в разделе HKEY_LOCAL_MACHINE\Software\Classes. Данные из разделов, добавленных в HKEY_CLASSES_ROOT, будут сохранены системой в разделе HKEY_LOCAL_MACHINE\Software\Classes. Если изменяется параметр в одном из подразделов раздела HKEY_CLASSES_ROOT и такой подраздел уже существует в HKEY_CURRENT_USER\Software\Classes, то для хранения информации будет использован раздел HKEY_CURRENT_USER\Software\Classes, а не HKEY_LOCAL_MACHINE\Software\Classes.

HKEY_CURRENT_CONFIG Данный раздел содержит сведения о профиле оборудования, используемом локальным компьютером при запуске системы.

В	Windows	7,	согласно	сведениям	из
---	---------	----	----------	-----------	----

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\hivelist файлы реестра хранятся в следующих местах:

- 01= Ветка реестра «HKEY_LOCAL_MACHINE\HARDWARE» формируется в зависимости от оборудования (динамически);
- 02= Ветка реестра «HKEY_LOCAL_MACHINE\BCD00000000» формируется из файла «%SystemRoot%\Boot\BCD»
- 03= Ветка реестра «HKEY_LOCAL_MACHINE\SYSTEM» формируется из файла «%SystemRoot%\System32\config\SYSTEM»
- 04= Ветка реестра «HKEY_LOCAL_MACHINE\SOFTWARE» формируется из файла «%SystemRoot%\System32\config\SOFTWARE»
- 05= Ветка реестра «HKEY_LOCAL_MACHINE\SECURITY» формируется из файла «%SystemRoot%\System32\config\SECURITY»
- 06= Ветка реестра «HKEY_LOCAL_MACHINE\SAM» формируется из

файла «%SystemRoot%\System32\config\SAM»

- 07= Ветка реестра «HKEY_USERS\DEFAULT» формируется из файла «%SystemRoot%\System32\config\DEFAULT»
- 08= Ветка реестра «HKEY_USERS\S-1-5-18» формируется из файла «%SystemRoot%\System32\config\systemprofile\NTUSER.DAT» (относится к учетной записи system)^[1]
- 09= Ветка реестра «HKEY_USERS\S-1-5-19» формируется из файла «%SystemRoot%\ServiceProfiles\LocalService\NTUSER.DAT» (относится к учетной записи LocalService)
- 10= Ветка реестра «HKEY_USERS\S-1-5-20» формируется из файла «%SystemRoot%\ServiceProfiles\NetworkService\NTUSER.DAT» (относится к учетной записи NetworkService)
- 11= Ветка реестра «HKEY_USERS\<SID_пользователя>» формируется из файла «%USERPROFILE%\NTUSER.DAT»
- 12= Ветка реестра «HKEY_USERS\<SID_пользователя>_Classes» формируется из файла «%USERPROFILE%\AppData\Local\Microsoft\Windows\UsrClass.dat».

Резервные копии файлов реестра DEFAULT, SAM, SECURITY, SOFTWARE и SYSTEM находятся в папке «%SystemRoot%\System32\config\RegBack». Само резервное копирование производится силами Планировщика задач в 0 ч. 00 мин. каждые 10 дней по заданию «RegIdleBackup», расположенному в иерархии задач по пути «\Microsoft\Windows\Registry».

Следующая таблица содержит список типов данных, определенных и используемых Windows на сегодняшний день.

Имя	Тип	Описание
Двоичный параметр	REG_BINARY	Необработанные двоичные данные. Большинство сведений об аппаратных компонентах хранится в виде двоичных данных и выводится в редакторе реестра в шестнадцатеричном формате.
Параметр DWORD	REG_DWORD	Данные представлены в виде значения, длина которого составляет 4 байта (32-разрядное целое). Этот тип данных используется для хранения параметров драйверов устройств и служб. Значение отображается в окне редактора реестра в двоичном,

		шестнадцатеричном или десятичном формате. Эквивалентами типа DWORD являются DWORD_LITTLE_ENDIAN (самый младший байт хранится в памяти в первом числе) и REG_DWORD_BIG_ENDIAN (самый младший байт хранится в памяти в последнем числе).
Расширяемая строка данных	REG_EXPAND_SZ	Строка данных переменной длины. Этот тип данных включает переменные, обрабатываемые при использовании данных программой или службой.
Многострочный параметр	REG_MULTI_SZ	Многострочный текст. Этот тип, как правило, имеют списки и другие записи в формате, удобном для чтения. Записи разделяются пробелами, запятыми или другими символами.
Строковый параметр	REG_SZ	Текстовая строка фиксированной длины.
Двоичный параметр	REG_RESOURCE_LIST	Последовательность вложенных массивов. Служит для хранения списка ресурсов, которые используются драйвером устройства или управляемым им физическим устройством. Обнаруженные данные система сохраняет в разделе \ResourceMap. В окне редактора реестра эти данные отображаются в виде двоичного параметра в шестнадцатеричном формате.
Двоичный параметр	REG_RESOURCE_REQUIREMENTS_LIST	Последовательность вложенных массивов. Служит для хранения списка драйверов аппаратных ресурсов, которые могут быть использованы определенным драйвером устройства или управляемым им физическим устройством. Часть этого списка система записывает в раздел \ResourceMap. Данные определяются системой. В окне редактора реестра они отображаются в виде двоичного параметра в шестнадцатеричном формате.
Двоичный параметр	REG_FULL_RESOURCE_LIST	Последовательность вложенных массивов. Служит для

параметр	RCE_DESCRIPTOR	хранения списка ресурсов, которые используются физическим устройством. Обнаруженные данные система сохраняет в разделе \HardwareDescription. В окне редактора реестра эти данные отображаются в виде двоичного параметра в шестнадцатеричном формате.
Отсутствует	REG_NONE	Данные, не имеющие определенного типа. Такие данные записываются в реестр системой или приложением. В окне редактора реестра отображаются в виде двоичного параметра в шестнадцатеричном формате.
Ссылка	REG_LINK	Символическая ссылка в формате Юникод.
Параметр QWORD	REG_QWORD	Данные, представленные в виде 64-разрядного целого. Начиная с Windows 2000, такие данные отображаются в окне редактора реестра в виде двоичного параметра.

Редактирование реестра

Для редактирования реестра администратор может воспользоваться редактором реестра (Regedit.exe), групповой или системной политикой, файлами реестра (REG) либо специальным сценарием (например файлом сценария на языке VisualBasic).

Использование редактора реестра

Предупреждение. Неправильное изменение параметров системного реестра с помощью редактора реестра или любым иным способом может привести к серьезным неполадкам. Для их устранения может потребоваться переустановка операционной системы.

Редактор реестра можно использовать для выполнения следующих задач:

- поиск поддерева, раздела, подраздела или параметра;
- добавление подраздела или параметра;
- изменение значения параметра;
- удаление подраздела или параметра;
- переименование подраздела или параметра.

Область переходов редактора реестра отображает набор папок. Каждая папка представляет собой раздел реестра локального компьютера. При просмотре реестра удаленного компьютера будут видны только два стандартных раздела: HKEY_USERS и

HKEY_LOCAL_MACHINE.

Использование групповой политики

Консоль управления Microsoft (MMC) содержит средства администрирования, которые используются для управления сетями, компьютерами, службами и другими системными компонентами. С помощью оснастки "Групповая политика" администратор может определить параметры безопасности для пользователей и компьютеров. Групповую политику можно реализовать на локальном компьютере с помощью локальной оснастки "Групповая политика" (файл Gpedit.msc) или в Active Directory с помощью оснастки "Active Directory - пользователи и компьютеры". Дополнительные сведения об использовании групповой политики см. в справке по соответствующей оснастке, используемой для работы с групповой политикой.

Использование файла реестра (REG)

Для внесения изменений в реестр можно создать файл реестра (с расширением REG) и выполнить его на соответствующем компьютере. Выполнить REG-файл можно вручную или с помощью сценария входа.

Использование сервера сценариев Windows

Сервер сценариев Windows позволяет выполнять сценарии на языке VBScript или JScript непосредственно в операционной системе. В таких сценариях для удаления, чтения и записи разделов и параметров реестра используются методы сервера сценариев Windows.

Использование консольной программы редактирования реестра Windows

Для редактирования системного реестра можно воспользоваться консольной программой редактирования реестра Windows (Reg.exe). Для получения справки по программе Reg.exe введите в командной строке команду **reg /?** и нажмите кнопку **ОК**.

Восстановление реестра

Для восстановления реестра используйте подходящий способ из числа указанных ниже.

Восстановление разделов реестра

Чтобы восстановить экспортированные разделы реестра, необходимо выполнить REG-файл, сохраненный при экспорте подразделов реестра. Кроме того, можно восстановить весь реестр из его резервной копии. Дополнительные сведения о восстановлении всего реестра см. в подразделе "Восстановление всего реестра" этой статьи.

Восстановление всего реестра

Для восстановления всего реестра необходимо восстановить состояние системы из его резервной копии. Дополнительные сведения о восстановлении состояния системы из резервной копии см. в следующей статье базы знаний Майкрософт: [309340](#) Восстановление файлов и папок с помощью программы «Архивация данных» в Windows XP

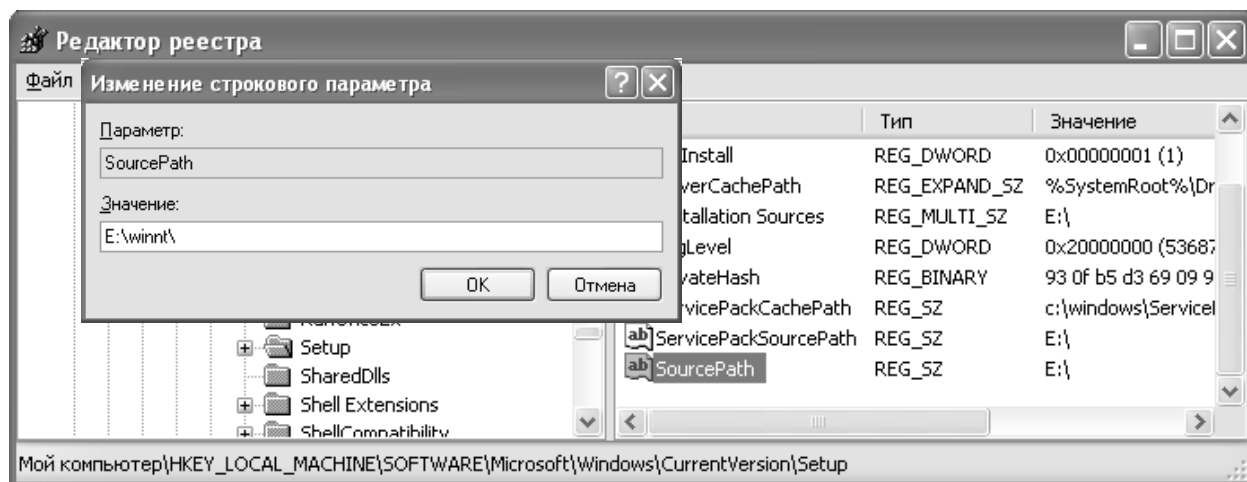
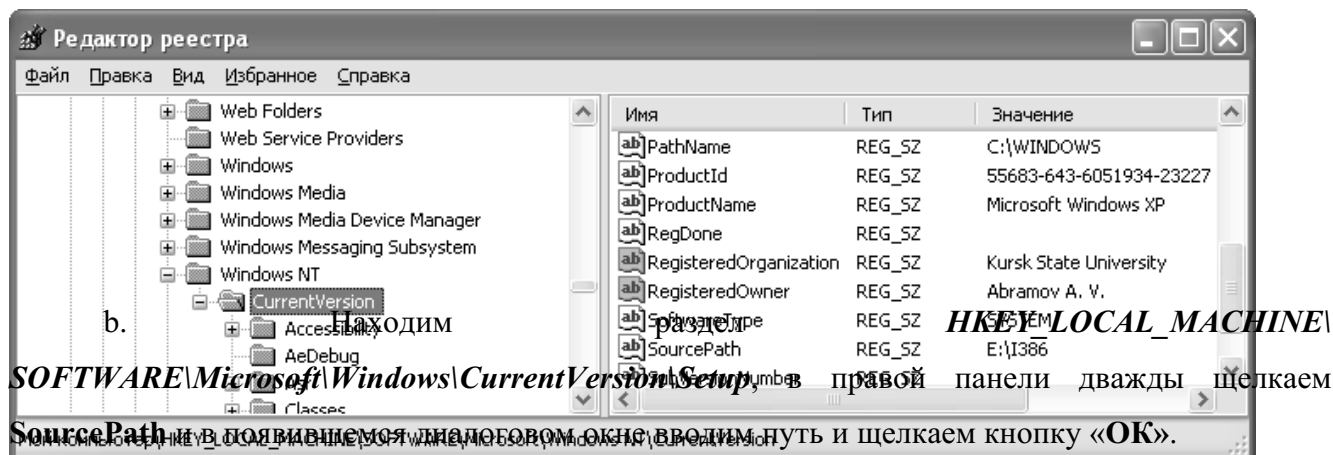
При создании резервной копии состояния системы обновленные копии файлов реестра сохраняются в папке %SystemRoot%\Repair.

2. Практические задания.

1. Выясните с помощью реестра установленную версию Windows и путь к системному каталогу Windows.
2. Выполнить копирование реестра средствами WINDOWS.
3. Изучить команды редактора реестра
4. Изучить структуру реестра
5. Внести изменения в реестр:
 - a. Изменение регистрации владельца и организации.
 - b. Изменить путь, используемый программой установки Windows.
 - c. Удалить список команды **Выполнить...**
 - d. Добавить команду **Открыть с помощью...** типам файлов.
6. По заданию преподавателя добавить программу для запуска во время загрузки Windows.
7. По заданию преподавателя установите выполнение команды или запуск определенной программы при загрузке командной консоли.
8. Показать результат преподавателю.
9. Восстановить сохраненный системный реестр.

Технология выполнения задания:

1. Запускаем **Редактор реестра**: **Пуск \ Выполнить...** и вводим в окне **Открыть** название программы **regedit**. Далее находим раздел **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion** и просматриваем значения переменных Version, VersionNumber и SystemRoot.
 2. Выделяем элемент реестра **Мой компьютер** и выбираем команду **Файл \ Экспорт**. Появится диалоговое окно **Экспорт файла реестра**.
- В этом окне задаём папку для сохранения, имя копии реестра **OldReg.reg**, устанавливаем переключатель **Весь реестр** в группе **Диапазон экспорта** и нажимаем кнопку «**Сохранить**».
5.
 - a. Находим раздел **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion**, в правой панели дважды щелкаем параметр **RegisteredOwner** и в появившемся диалоговом окне вводим имя и щелкаем кнопку «**ОК**». Аналогичные действия выполняем с параметром **RegisteredOrganization**.

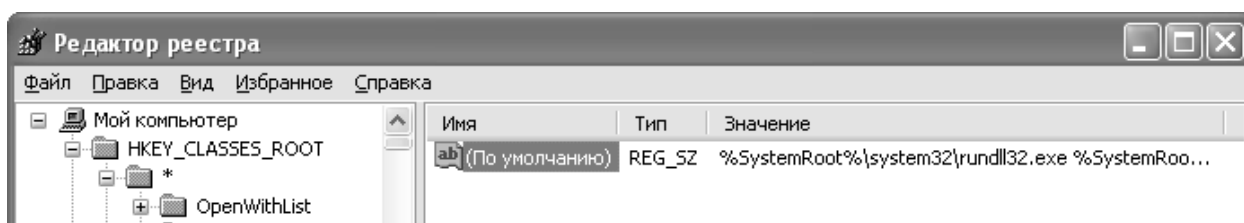


с. Находим раздел

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU, На правой панели выбираем необходимые параметры и удаляем их (**Правка \ Удалить**).

d. Находим раздел **HKEY_CLASSES_ROOT\Unknown\shell\openas \command**, в правой панели дважды щелкаем параметр **По умолчанию**. Скопируем его значение из появившегося диалогового окна в буфер обмена. Закрываем окно и переходим к разделу **HKEY_CLASSES_ROOT***.

В левой панели щелкаем правой кнопкой мыши звездочку (*) и выбираем команду **Создать \ Раздел**, называем новый раздел **Shell**. Далее в **Shell** создаём подраздел **openas**, а в **openas** – **command**. В правой панели щелкаем дважды кнопкой мыши параметр **По умолчанию**. В поле «Значение» появившегося диалогового окна вставляем содержимое буфера обмена.



6. Переходим в раздел

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows

CurrentVersion\Run. В правой панели щелкаем правой кнопкой мыши и выбираем команду ***Создать \ Строковый параметр.*** Вводим имя параметра, затем меняем его значение на «D:\Program Files\Winamp\Winamp.exe».

7. Переходим в раздел

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Command Processor. Выбираем

параметр ***AutoRun*** и в качестве его значения задаём строку

«color e5».

9. Выбираем команду ***Файл \ Импорт.*** В появившемся диалоговом окне вводим имя файла «C:\OldReg.rg» и нажимаем кнопку «Открыть».

Контрольные вопросы:

- 1.Перечислить системные процессы Windows.
- 2.Дать определения драйвера устройства в различных контекстах.
- 3.Пояснить принцип функционирования уровня абстрагирования от оборудования.
- 4.Перечислить основные компоненты исполнительной системы Windows.
- 5.Дать определения подсистемам Windows.
- 6.Перечислить возможные места хранения назначенных политик безопасности
- 7.Отобразить переменные среды в Windows двумя способами: из командной оболочки и окна свойств системы
- 8.Задать переменную среды с различными вариантами динамически формируемых значений.

Список литературы, рекомендуемый к использованию по данной теме:

Основная: 1-4

Дополнительная: 1-3

Интернет-ресурсы: 1

Методическая: 1-3

Программное обеспечение: 1-4

Практическая работа №4

Тема: Модели безопасности основных операционных систем Программно-технический уровень информационной безопасности

1. Подготовка к занятию

1. Изучить (повторить) теоретический материал.
2. Ознакомиться с программой работы.
3. Подготовить отчет о работе.
4. Ответить на контрольные вопросы.

2. Распределение времени занятия:

Всего: 90 мин

Вступительная часть 2 мин

Проверка готовности студентов к занятию 5 мин

Основная часть – 70 мин

Проверка выполнения практического занятия 10 мин

Заключительная часть 3 мин

3. Формируемые компетенции: ПК-1, ПК-3

4. Правила работы в лаборатории

К работе в лаборатории допускаются лица, изучившие правила и меры безопасности, сдавшие зачет по ним и усвоившие порядок выполнения лабораторной работы.

4.1. Требования безопасности перед началом работ

- ЗАПРЕЩАЕТСЯ: переодеваться, пользоваться огнем, курить, принимать пищу в лаборатории.

- Убедиться в целостности электрических розеток и разъемов. В лаборатории необходимо быть в сменной обуви.

- Включение компьютера производить только после получения допуска по выполняемой работе и разрешения преподавателя или лаборанта.

4.2. Требования безопасности во время работы

- выполняя практическое занятие, студенты обязаны использовать только вычислительную технику, периферийное оборудование, соединительные кабели, измерительное оборудование и носители информации, непосредственно относящиеся к данному лабораторному занятию;

- подключение и отключение составляющих вычислительного комплекса производить только при полном снятии напряжения со всех составляющих вычислительного комплекса;

- при обнаружении неисправностей в оборудовании немедленно отключить источники питания и доложить об этом руководителю занятий или лаборанту.

4.3. Требования безопасности по окончании работы

- доложить руководителю занятий или лаборанту о завершении работ;

- привести в порядок и сдать рабочее место лаборанту, и доложить руководству

Теоретические сведения.

Внутренние и внешние команды. Структура команд.

Некоторые команды распознаются и выполняются непосредственно самим командным интерпретатором — такие команды называются внутренними (например, COPY или DIR). Другие команды операционной системы представляют собой отдельные программы, расположенные по умолчанию в том же каталоге, что и Cmd.exe, которые Windows загружает и выполняет аналогично другим программам. Такие команды называются внешними (например, MORE или XCOPY).

Рассмотрим структуру самой командной строки и принцип работы с ней. Для того, чтобы выполнить команду, вы после приглашения командной строки (например, C:\>) вводите имя этой команды (регистр не важен), ее параметры и ключи (если они необходимы) и нажимаете клавишу <Enter>. Например:

```
C:\>COPY C:\myfile.txt A:\ /V
```

Имя команды здесь — COPY, параметры — C:\myfile.txt и A:\, а ключом является /V. Отметим, что в некоторых командах ключи могут начинаться не с символа /, а с символа – (минус), например, -V.

Многие команды Windows имеют большое количество дополнительных параметров и ключей, запомнить которые зачастую бывает трудно. Большинство команд снабжено встроенной справкой, в которой кратко описываются назначение и синтаксис данной команды. Получить доступ к такой справке можно путем ввода команды с ключом /?. Например, если выполнить команду ATTRIB /?, то в окне MS-DOS мы увидим следующий текст:

Отображение и изменение атрибутов файлов.

```
ATTRIB [+R|-R] [+A|-A] [+S|-S] [+H|-H] [[диск:][путь]имя_файла] [/S]
```

+ Установка атрибута.

- Снятие атрибута.

R Атрибут "Только чтение".

A Атрибут "Архивный".

S Атрибут "Системный".

H Атрибут "Скрытый".

/S Обработка файлов во всех вложенных папках указанного пути.

Для некоторых команд текст встроенной справки может быть довольно большим и не уместиться на одном экране. В этом случае помощь можно выводить последовательно по одному экрану с помощью команды MORE и символа конвейеризации |, например:

```
XCOPY /? | MORE
```

В этом случае после заполнения очередного экрана вывод помощи будет прерываться до нажатия любой клавиши. Кроме того, используя символы перенаправления вывода > и >>, можно текст, выводимый на экран, направить в текстовый файл для дальнейшего просмотра. Например, для вывода текста справки к команде XCOPY в текстовый файл xcopy.txt, используется следующая команда:

```
XCOPY /? > XCOPY.TXT
```

Замечание

Вместо имени файла можно указывать обозначения устройств компьютера. В Windows поддерживаются следующие имена устройств: PRN (принтер), LPT1–LPT3 (соответствующие параллельные порты), AUX (устройство, присоединяемое к последовательному порту 1), COM1–COM3 (соответствующие последовательные порты), CON (терминал: при вводе это клавиатура, при выводе - монитор), NUL (пустое устройство, все операции ввода/вывода для него игнорируются).

Перенаправление ввода/вывода и конвейеризация (композиция) команд

Рассмотрим более подробно поддерживаемые в Windows UNIX-подобные концепции переназначения устройств стандартного ввода/вывода и конвейерного выполнения команд. С помощью переназначения устройств ввода/вывода одна программа может направить свой

вывод на вход другой или перехватить вывод другой программы, используя его в качестве своих входных данных. Таким образом, имеется возможность передавать информацию от процесса к процессу при минимальных программных издержках. Практически это означает, что для программ, которые используют стандартные входные и выходные устройства, операционная система позволяет:

выводить сообщения программ не на экран (стандартный выходной поток), а в файл или на принтер (перенаправление вывода);

читать входные данные не с клавиатуры (стандартный входной поток), а из заранее подготовленного файла (перенаправление ввода);

передавать сообщения, выводимые одной программой, в качестве входных данных для другой программы (конвейеризация или композиция команд).

Из командной строки эти возможности реализуются следующим образом. Для того, чтобы перенаправить текстовые сообщения, выводимые какой-либо командой, в текстовый файл, нужно использовать конструкцию

команда > имя_файла

Если при этом заданный для вывода файл уже существовал, то он перезаписывается (старое содержимое теряется), если не существовал — создается. Можно также не создавать файл заново, а дописывать информацию, выводимую командой, в конец существующего файла. Для этого команда перенаправления вывода должна быть задана так:

команда >> имя_файла

С помощью символа < можно прочесть входные данные для заданной команды не с клавиатуры, а из определенного (заранее подготовленного) файла:

команда < имя_файла

Приведем несколько примеров перенаправления ввода/вывода.

Вывод встроенной справки для команды COPY в файл copy.txt:

COPY /? > copy.txt

Добавление текста справки для команды XCOPY в файл copy.txt:

XCOPY /? >> copy.txt

Ввод новой даты из файла date.txt (DATE — это команда для просмотра и изменения системной даты):

DATE < date.txt

Если при выполнении определенной команды возникает ошибка, то сообщение об этом по умолчанию выводится на экран. В случае необходимости сообщения об ошибках (стандартный поток ошибок) можно перенаправить в текстовый файл с помощью конструкции

команда 2> имя_файла

В этом случае стандартный вывод будет производиться на экран. Также имеется возможность информационные сообщения и сообщения об ошибках выводить в один и тот же файл.

Делается это следующим образом:

команда > имя_файла 2>&1

Например, в приведенной ниже команде стандартный выходной поток и стандартный поток ошибок перенаправляются в файл copy.txt:

XCOPY A:\1.txt C: > copy.txt 2>&1

Наконец, с помощью конструкции

команда1 | команда2

можно использовать сообщения, выводимые первой командой, в качестве входных данных для второй команды (конвейер команд).

Используя механизмы перенаправления ввода/вывода и конвейеризации, можно из командной строки посылать информацию на различные устройства и автоматизировать ответы на запросы, выдаваемые командами или программами, использующими стандартный ввод. Для решения таких задач подходит команда

ECHO [сообщение]

которая выводит сообщение на экран. Рассмотрим примеры использования этой команды.

Посылка символа прогона на принтер:

ECHO ^L > PRN

Удаление всех файлов в текущем каталоге без предупреждения (автоматический положительный ответ на запрос об удалении):

ECHO y | DEL *.*

Соединение по телефону из командной строки (модем связан с портом COM2):

ECHO ATDT 1(123)555-1234 > COM2

Команды MORE и SORT

Одной из наиболее часто используемых команд, для работы с которой применяется перенаправление ввода/вывода и конвейеризация, является MORE. Эта команда считывает стандартный ввод из конвейера или перенаправленного файла и выводит информацию частями, размер каждой из которых не больше размера экрана. Используется MORE обычно для просмотра длинных файлов. Возможны три варианта синтаксиса этой команды:

MORE [диск:][путь]имя_файла

MORE < [диск:][путь]имя_файла

имя_команды | MORE

Параметр [диск:][путь]имя_файла определяет расположение и имя файла с просматриваемыми на экране данными. Параметр имя_команды задает команду, вывод которой отображается на экране (например, DIR или команда TYPE, используемая для вывода содержимого текстового файла на экран). Приведем два примера.

Для поэкранного просмотра помощи команды DIR используется команда:

XCOPY /? | MORE

Для поэкранного просмотра текстового файла news.txt возможны следующие варианты команд:

MORE news.txt

MORE < news.txt

TYPE news.txt | MORE

Другой распространенной командой, использующей перенаправление ввода/вывода и конвейеризацию, является SORT. Эта команда работает как фильтр: она считывает символы в заданном столбце, упорядочивает их в возрастающем или убывающем порядке и выводит отсортированную информацию в файл, на экран или другое устройство. Возможны два варианта синтаксиса этой команды:

SORT [/R] [/+n] [[диск1:][путь1]файл1] [> [диск2:][путь2]файл2]

или

[команда] SORT [/R] [/+n] [> [диск2:][путь2]файл2]

В первом случае параметр [диск1:][путь1]файл1 определяет имя файла, который нужно отсортировать. Во втором случае будут отсортированы выходные данные указанной команды. Если параметры файл1 или команда не заданы, то SORT будет считывать данные с устройства стандартного ввода.

Параметр [диск2:][путь2]файл2 задает файл, в который будет направляется сортированный вывод; если этот параметр не задан, то вывод будет направлен на устройство стандартного вывода.

По умолчанию сортировка выполняется в порядке возрастания. Ключ /R позволяет изменить порядок сортировки на обратный (от Z к A и затем от 9 до 0). Например, для поэкранного просмотра отсортированного в обратном порядке файла price.txt, нужно задать следующую команду:

SORT /R < price.txt |MORE

Ключ /+n задает сортировку в файле по символам n-го столбца. Например, /+10 означает, что сортировка должна осуществляться, начиная с 10-й позиции в каждой строке. По умолчанию файл сортируется по первому столбцу.

Примеры команд для работы с файловой системой

Рассмотрим некоторые наиболее часто используемые команды для работы с файловой системой. Отметим сначала несколько особенностей определения путей к файлам в Windows.

Пути к объектам файловой системы

Напомним, что файловая система логически имеет древовидную структуру и имена файлов задаются в формате [диск:][путь\]имя_файла, то есть обязательным параметром является только имя файла. При этом, если путь начинается с символа "\", то маршрут вычисляется от корневого каталога, иначе — от текущего каталога. Например, имя C:123.txt задает файл 123.txt в текущем каталоге на диске C:, имя C:\123.txt — файл 123.txt в корневом каталоге на диске C:, имя ABC\123.txt — файл 123.txt в подкаталоге ABC текущего каталога.

Существуют особые обозначения для текущего каталога и родительского каталогов. Текущий каталог обозначается символом . (точка), его родительский каталог — символами .. (две точки).

Например, если текущим каталогом является C:\WINDOWS, то путь к файлу autoexec.bat в корневом каталоге диска C: может быть записан в виде ..\autoexec.bat.

В именах файлов (но не дисков или каталогов) можно применять так называемые групповые символы или шаблоны: ? (вопросительный знак) и * (звездочка). Символ * в имени файла означает произвольное количество любых допустимых символов, символ ? — один произвольный символ или его отсутствие. Скажем, под шаблон text??1.txt подходят, например, имена text121.txt и text11.txt, под шаблон text*.txt — имена text.txt, textab12.txt, а под шаблон text.* — все файлы с именем text и произвольным расширением.

Для того, чтобы использовать длинные имена файлов при работе с командной строкой, их нужно заключать в двойные кавычки. Например, чтобы запустить файл с именем 'Мое приложение.exe' из каталога 'Мои документы', нужно в командной строке набрать "C:\Мои документы\Мое приложение.exe" и нажать клавишу <Enter>.

Перейдем теперь непосредственно к командам для работы с файловой системой.

Команда CD

Текущий каталог можно изменить с помощью команды

CD [диск:][путь\]

Путь к требуемому каталогу указывается с учетом приведенных выше замечаний. Например, команда CD \ выполняет переход в корневой каталог текущего диска. Если запустить команду CD без параметров, то на экран будут выведены имена текущего диска и каталога.

Команда COPY

Одной из наиболее часто повторяющихся задач при работе на компьютере является копирование и перемещение файлов из одного места в другое. Для копирования одного или нескольких файлов используется команда COPY.

Синтаксис этой команды:

COPY [/A/B] источник [/A/B] [+ источник [/A/B] [+ ...]]
[результат [/A/B]] [/V]/[Y/-Y]

Краткое описание параметров и ключей команды COPY приведено в [табл. 2.1](#).

Таблица 2.1. Параметры и ключи команды COPY

Параметр	Описание
источник	Имя копируемого файла или файлов
/A	Файл является текстовым файлом ASCII, то есть конец файла обозначается символом с кодом ASCII 26 (<Ctrl>+<Z>)
/B	Файл является двоичным. Этот ключ указывает на то, что интерпретатор команд должен при копировании считывать из источника число байт, заданное размером в каталоге копируемого файла
результат	Каталог для размещения результата копирования и/или имя создаваемого файла
/V	Проверка правильности копирования путем сравнения файлов после копирования
/Y	Отключение режима запроса подтверждения на замену файлов
/-Y	Включение режима запроса подтверждения на замену файлов

Приведем примеры использования команды COPY.

Копирование файла abc.txt из текущего каталога в каталог D:\PROGRAM под тем же именем:

```
COPY abc.txt D:\PROGRAM
```

Копирование файла abc.txt из текущего каталога в каталог D:\PROGRAM под новым именем def.txt:

```
COPY abc.txt D:\PROGRAM\def.txt
```

Копирование всех файлов с расширением txt с диска A: в каталог 'Мои документы' на диске C:

```
COPY A:\*.txt "C:\Мои документы"
```

Если не задать в команде целевой файл, то команда COPY создаст копию файла-источника с тем же именем, датой и временем создания, что и исходный файл, и поместит новую копию в текущий каталог на текущем диске. Например, для того, чтобы скопировать все файлы из корневого каталога диска A: в текущий каталог, достаточно выполнить такую краткую команду:

```
COPY A:\*.*
```

В качестве источника или результата при копировании можно указывать имена не только файлов, но и устройств компьютера. Например, для того, чтобы распечатать файл abc.txt на принтере, можно воспользоваться командой копирования этого файла на устройство PRN: COPY abc.txt PRN

Другой интересный пример: создадим новый текстовый файл и запишем в него информацию, без использования текстового редактора. Для этого достаточно ввести команду COPY CON my.txt, которая будет копировать то, что вы набираете на клавиатуре, в файл my.txt (если этот файл существовал, то он перезапишется, иначе — создастся). Для завершения ввода необходимо ввести символ конца файла, то есть нажать клавиши <Ctrl>+<Z>.

Команда COPY может также объединять (склеивать) нескольких файлов в один. Для этого необходимо указать единственный результирующий файл и несколько исходных. Это достигается путем использования групповых знаков (? и *) или формата файл1 + файл2 + файл3. Например, для объединения файлов 1.txt и 2.txt в файл 3.txt можно задать следующую команду:

```
COPY 1.txt+2.txt 3.txt
```

Объединение всех файлов с расширением dat из текущего каталога в один файл all.dat может быть произведено так:

```
COPY /B *.dat all.dat
```

Ключ /B здесь используется для предотвращения усечения соединяемых файлов, так как при комбинировании файлов команда COPY по умолчанию считает файлами текстовыми.

Если имя целевого файла совпадает с именем одного из копируемых файлов (кроме первого), то исходное содержимое целевого файла теряется. Если имя целевого файла опущено, то в его качестве используется первый файл из списка. Например, команда COPY 1.txt+2.txt добавит к содержимому файла 1.txt содержимое файла 2.txt. Командой COPY можно воспользоваться и для присвоения какому-либо файлу текущей даты и времени без модификации его содержимого. Для этого нужно ввести команду типа

```
COPY /B 1.txt +,,
```

Здесь запятые указывают на пропуск параметра приемника, что и приводит к требуемому результату.

Команда COPY имеет и свои недостатки. Например, с ее помощью нельзя копировать скрытые и системные файлы, файлы нулевой длины, файлы из подкаталогов. Кроме того, если при копировании группы файлов COPY встретит файл, который в данный момент нельзя скопировать (например, он занят другим приложением), то процесс копирования полностью прервется, и остальные файлы не будут скопированы.

Команда XCOPY

Указанные при описании команды COPY проблемы можно решить с помощью команды XCOPY, которая предоставляет намного больше возможностей при копировании.

Необходимо отметить, правда, что XCOPY может работать только с файлами и каталогами, но не с устройствами.

Синтаксис этой команды:

XCOPY источник [результат] [ключи]

Команда XCOPY имеет множество ключей, мы коснемся лишь некоторых из них.

Ключ /D[:[дата]] позволяет копировать только файлы, измененные не ранее указанной даты.

Если параметр дата не указан, то копирование будет производиться только если источник новее результата. Например, команда

XCOPY "C:\Мои документы*.*" "D:\BACKUP\Мои документы" /D

скопирует в каталог 'D:\BACKUP\Мои документы' только те файлы из каталога 'C:\Мои документы', которые были изменены со времени последнего подобного копирования или которых вообще не было в 'D:\BACKUP\Мои документы'.

Ключ /S позволяет копировать все непустые подкаталоги в каталоге-источнике. С помощью же ключа /E можно копировать вообще все подкаталоги, включая и пустые.

Если указан ключ /C, то копирование будет продолжаться даже в случае возникновения ошибок. Это бывает очень полезным при операциях копирования, производимых над группами файлов, например, при резервном копировании данных.

Ключ /I важен для случая, когда копируются несколько файлов, а файл назначения отсутствует.

При задании этого ключа команда XCOPY считает, что файл назначения должен быть каталогом. Например, если задать ключ /I в команде копирования всех файлов с расширением txt из текущего каталога в несуществующий еще подкаталог TEXT,

XCOPY *.txt TEXT /I

то подкаталог TEXT будет создан без дополнительных запросов.

Ключи /Q, /F и /L отвечают за режим отображения при копировании. При задании ключа /Q имена файлов при копировании не отображаются, ключа /F — отображаются полные пути источника и результата. Ключ /L обозначает, что отображаются только файлы, которые должны быть скопированы (при этом само копирование не производится).

С помощью ключа /H можно копировать скрытые и системные файлы, а с помощью ключа /R — заменять файлы с атрибутом "Только для чтения". Например, для копирования всех файлов из корневого каталога диска C: (включая системные и скрытые) в каталог SYS на диске D:, нужно ввести следующую команду:

XCOPY C: *.* D:\SYS /H

Ключ /T позволяет применять XCOPY для копирования только структуры каталогов источника, без дублирования находящихся в этих каталогах файлов, причем пустые каталоги и подкаталоги не включаются. Для того, чтобы все же включить пустые каталоги и подкаталоги, нужно использовать комбинацию ключей /T /E.

Используя XCOPY можно при копировании обновлять только уже существующие файлы (новые файлы при этом не записываются). Для этого применяется ключ /U. Например, если в каталоге C:\2 находились файлы a.txt и b.txt, а в каталоге C:\1 — файлы a.txt, b.txt, c.txt и d.txt, то после выполнения команды

XCOPY C:\1 C:\2 /U

в каталоге C:\2 по-прежнему останутся лишь два файла a.txt и b.txt, содержимое которых будет заменено содержимым соответствующих файлов из каталога C:\1. Если с помощью XCOPY копировался файл с атрибутом "Только для чтения", то по умолчанию у файла-копии этот атрибут снимется. Для того, чтобы копировать не только данные, но и полностью атрибуты файла, необходимо использовать ключ /K.

Ключи /Y и /-Y определяют, нужно ли запрашивать подтверждение перед заменой файлов при копировании. /Y означает, что такой запрос нужен, /-Y — не нужен.

Команда DIR

Еще одной очень полезной командой является DIR [диск:][путь][имя_файла] [ключи], которая используется для вывода информации о содержимом дисков и каталогов.

Параметр [диск:][путь] задает диск и каталог, содержимое которого нужно вывести на экран.

Параметр [имя_файла] задает файл или группу файлов, которые нужно включить в список.

Например, команда

DIR C:*.bat

выведет на экран все файлы с расширением bat в корневом каталоге диска C:. Если задать эту команду без параметров, то выводится метка диска и его серийный номер, имена (в коротком и длинном вариантах) файлов и подкаталогов, находящихся в текущем каталоге, а также дата и время их последней модификации. После этого выводится число файлов в каталоге, общий объем (в байтах), занимаемый файлами, и объем свободного пространства на диске. Например: Том в устройстве C имеет метку PHYS1_PART2

Серийный номер тома: 366D-6107

Содержимое папки C:\aditor

```
.      <ПАПКА>    25.01.00  17:15 .
..     <ПАПКА>    25.01.00  17:15 ..
TEMPLT02.DAT      227 07.08.98  1:00 templt02.dat
UNINST1.000       1 093 02.03.99  8:36 UNINST1.000
HILITE.DAT        1 082 18.09.98 18:55 hilite.dat
TEMPLT01.DAT       48 07.08.98  1:00 templt01.dat
UNINST0.000      40 960 15.04.98  2:08 UNINST0.000
TTABLE.DAT        357 07.08.98  1:00 ttable.dat
ADITOR.EXE        461 312 01.12.99 23:13 aditor.exe
README.TXT        3 974 25.01.00 17:26 readme.txt
ADITOR.HLP        24 594 08.10.98 23:12 aditor.hlp
ТЕКСТО~1.TXT       0 11.03.01  9:02 Текстовый файл.txt
      11 файлов    533 647 байт
      2 папок     143 261 696 байт свободно
```

С помощью ключей команды DIR можно задать различные режимы расположения, фильтрации и сортировки. Например, при использовании ключа /W перечень файлов выводится в широком формате с максимально возможным числом имен файлов или каталогов на каждой строке.

Например:

Том в устройстве C имеет метку PHYS1_PART2

Серийный номер тома: 366D-6107

Содержимое папки C:\aditor

```
[.]      [..]      TEMPLT02.DAT      UNINST1.000  HILITE.DAT
TEMPLT01.DAT      UNINST0.000  TTABLE.DAT  ADITOR.EXE  README.TXT
ADITOR.HLP  ТЕКСТО~1.TXT
      11 файлов    533 647 байт
      2 папок     143 257 600 байт свободно
```

С помощью ключа /A[:]атрибуты можно вывести имена только тех каталогов и файлов, которые имеют заданные атрибуты (R — "Только чтение", A — "Архивный", S — "Системный", H — "Скрытый", префикс "-" имеет значение HE). Если ключ /Aиспользуется более чем с одним значением атрибута, будут выведены имена только тех файлов, у которых все атрибуты совпадают с заданными. Например, для вывода имен всех файлов в корневом каталоге диска C:, которые одновременно являются скрытыми и системными, можно задать команду

DIR C:\ /A:HS

а для вывода всех файлов, кроме скрытых — команду

DIR C:\ /A:-H

Отметим здесь, что атрибуту каталога соответствует буква D, то есть для того, чтобы, например, вывести список всех каталогов диска C:, нужно задать команду

DIR C: /A:D

Ключ /O[:]сортировка задает порядок сортировки содержимого каталога при выводе его командой DIR. Если этот ключ опущен, DIR печатает имена файлов и каталогов в том порядке, в котором они содержатся в каталоге. Если ключ /O задан, а параметр сортировка не указан, то DIR выводит имена в алфавитном порядке. В параметре сортировка можно использовать

следующие значения: N — по имени (алфавитная), S — по размеру (начиная с меньших), E — по расширению (алфавитная), D — по дате (начиная с более старых), A — по дате загрузки (начиная с более старых), G — начать список с каталогов. Префикс "-" означает обратный порядок. Если задается более одного значения порядка сортировки, файлы сортируются по первому критерию, затем по второму и т.д.

Ключ /S означает вывод списка файлов из заданного каталога и его подкаталогов.

Ключ /B перечисляет только названия каталогов и имена файлов (в длинном формате) по одному на строку, включая расширение. При этом выводится только основная информация, без итоговой. Например:

```
templ02.dat
UNINST1.000
hilite.dat
templ01.dat
UNINST0.000
ttable.dat
aditor.exe
readme.txt
aditor.hlp
```

Текстовый файл.txt

Команды MKDIR и RMDIR

Для создания нового каталога и удаления уже существующего пустого каталога используются команды MKDIR [диск:]путь и RMDIR [диск:]путь [ключи] соответственно (или их короткие аналоги MD и RD). Например:

```
MKDIR "C:\Примеры"
RMDIR "C:\Примеры"
```

Команда MKDIR не может быть выполнена, если каталог или файл с заданным именем уже существует. Команда RMDIR не будет выполнена, если удаляемый каталог не пустой.

Команда DEL

Удалить один или несколько файлов можно с помощью команды

```
DEL [диск:][путь]имя_файла [ключи]
```

Для удаления сразу нескольких файлов используются групповые знаки ? и *. Ключ /S позволяет удалить указанные файлы из всех подкаталогов, ключ /F — принудительно удалить файлы, доступные только для чтения, ключ /A[:атрибуты] — отбирать файлы для удаления по атрибутам (аналогично ключу /A[:атрибуты] в команде DIR).

Команда REN

Переименовать файлы и каталоги можно с помощью команды RENAME (REN). Синтаксис этой команды имеет следующий вид:

```
REN [диск:][путь][каталог1|файл1] [каталог2|файл2]
```

Здесь параметр каталог1|файл1 определяет название каталога/файла, которое нужно изменить, а каталог2|файл2 задает новое название каталога/файла. В любом параметре команды REN можно использовать групповые символы ? и *. При этом представленные шаблонами символы в параметре файл2 будут идентичны соответствующим символам в параметре файл1. Например, чтобы изменить у всех файлов с расширением txt в текущей директории расширение на doc, нужно ввести такую команду:

```
REN *.txt *.doc
```

Если файл с именем файл2 уже существует, то команда REN прекратит выполнение, и произойдет вывод сообщения, что файл уже существует или занят. Кроме того, в команде REN нельзя указать другой диск или каталог для создания результирующих каталога и файла. Для этой цели нужно использовать команду MOVE, предназначенную для переименования и перемещения файлов и каталогов.

Команда MOVE

Синтаксис команды для перемещения одного или более файлов имеет вид:

MOVE [/Y|/Y] [диск:][путь]имя_файла1[,...] результирующий_файл

Синтаксис команды для переименования папки имеет вид:

MOVE [/Y|/Y] [диск:][путь]каталог1 каталог2

Здесь параметр результирующий_файл задает новое размещение файла и может включать имя диска, двоеточие, имя каталога, либо их сочетание. Если перемещается только один файл, допускается указать его новое имя. Это позволяет сразу переместить и переименовать файл.

Например,

MOVE "C:\Мои документы\список.txt" D:\list.txt

Если указан ключ /-Y, то при создании каталогов и замене файлов будет выдаваться запрос на подтверждение. Ключ /Yотменяет выдачу такого запроса.

В чем значимость командной строки?

Некоторые возможности ОС Windows доступны только при использовании командной строки. Кроме того, ключи командной строки перекрывают параметры [Peecmpa Windows](#).

Практические действия.

Способы запуска

- Самый быстрый и простой – нажать клавишу с логотипом Windows + R;
– откроется окно *Выполнить*;
- введите **cmd**, нажмите *OK*;
- введите нужную команду, нажмите *Enter*.
- XP: *Пуск* → *Программы* → *Стандартные* → *Командная строка*.
- Vista: *Пуск* → *Все программы* → *Стандартные* → *Командная строка*.

Работа с мышью в окне командной строки

Чтобы в окне командной строки можно было работать мышью, щелкните правой кнопкой мыши по строке заголовка;

- из открывшегося контекстного меню выберите *Свойства*;
- в открывшемся окне *Свойства* на вкладке *Общие* в разделе *Редактирование* установите флажок *Выделение мышью* → *OK*;
- в появившемся окне *Изменение свойств* установите переключатель → *Сохранить свойства для других окон с тем же именем* → *OK*.

После этого можно выделять мышью текстовое содержимое окна командной строки, нажатием *Enter* копировать в буфер обмена, а затем вставлять, например, в файл *.txt, или *.doc, или *.rtf. Но самое главное, после этого можно вставлять – предварительно скопированные из текстового файла в буфер обмена – длинные текстовые строки команд, щелкнув правой кнопкой мыши по строке заголовка командной строки и выбрав из контекстного меню *Изменить* → *Вставить*.

Как изменить цвет окна командной строки

По умолчанию фоновый цвет окна интерпретатора команд (и в Windows XP, и в Vista) мрачного черного цвета – бр-р-р! Но вы можете изменить его.

Для этого:

- запустите интерпретатор команд;
- щелкните правой кнопкой мыши по строке заголовка;
- из открывшегося контекстного меню выберите *Свойства*;
- в открывшемся окне *Свойства* на вкладке *Цвета* установите переключатель *Фон рисунка*;
- внизу в строке палитры выделите нужный цвет → *OK*;
- в появившемся окне *Изменение свойств* установите переключатель → *Сохранить свойства для других окон с тем же именем* → *OK*.

Контрольные вопросы:

- 1) Перечислить системные процессы Windows.
- 2) Дать определения драйвера устройства в различных контекстах.
- 3) Пояснить принцип функционирования уровня абстрагирования от оборудования.
- 4) Перечислить основные компоненты исполнительной системы Windows.
- 5) Дать определения подсистемам Windows.
- 6) Перечислить возможные места хранения назначенных политик безопасности
- 7) Отобразить переменные среды в Windows двумя способами: из командной оболочки и окна свойств системы
- 8) Задать переменную среды с различными вариантами динамически формируемых значений.

Список литературы, рекомендуемый к использованию по данной теме:

Основная: 1-4

Дополнительная: 1-3

Интернет-ресурсы: 1

Методическая: 1-3

Программное обеспечение: 1-4

Практическая работа №5

Тема: Угрозы безопасности информации в информационно-вычислительных системах
Основные понятия и положения защиты информации в информационно-вычислительных системах

1. Подготовка к занятию

1. Изучить (повторить) теоретический материал.
2. Ознакомиться с программой работы.
3. Подготовить отчет о работе.
4. Ответить на контрольные вопросы.

2. Распределение времени занятия:

Всего: 90 мин

Вступительная часть 2 мин

Проверка готовности студентов к занятию 5 мин

Основная часть – 70 мин

Проверка выполнения практического занятия 10 мин

Заключительная часть 3 мин

3. Формируемые компетенции: ПК-1, ПК-3

4. Правила работы в лаборатории

К работе в лаборатории допускаются лица, изучившие правила и меры безопасности, сдавшие зачет по ним и усвоившие порядок выполнения лабораторной работы.

4.1. Требования безопасности перед началом работ

- ЗАПРЕЩАЕТСЯ: переодеваться, пользоваться огнем, курить, принимать пищу в лаборатории.

- Убедиться в целостности электрических розеток и разъемов. В лаборатории необходимо быть в сменной обуви.

- Включение компьютера производить только после получения допуска по выполняемой работе и разрешения преподавателя или лаборанта.

4.2. Требования безопасности во время работы

- выполняя практическое занятие, студенты обязаны использовать только вычислительную технику, периферийное оборудование, соединительные кабели, измерительное оборудование и носители информации, непосредственно относящиеся к данному лабораторному занятию;

- подключение и отключение составляющих вычислительного комплекса производить только при полном снятии напряжения со всех составляющих вычислительного комплекса;

- при обнаружении неисправностей в оборудовании немедленно отключить источники питания и доложить об этом руководителю занятий или лаборанту.

4.3. Требования безопасности по окончании работы

- доложить руководителю занятий или лаборанту о завершении работ;
- привести в порядок и сдать рабочее место лаборанту, и доложить руководству

Организация пакетных файлов и сценариев в ОС Windows

Пакетный файл – это неформатированный текстовый файл ASCII, содержащий одну или несколько команд ОС.

Имена пакетных файлов имеют расширения **.cmd** или **.bat**. ОС при работе с пакетным файлом последовательно обрабатывает его команды после ввода его имени в строке командной оболочки или запуска из другой программы.

Сценарий – это программа, состоящая из набора инструкций для работы приложения или служебной утилиты. Сценарий – разновидность пакетного файла. Инструкции в сценариях обычно выражаются с использованием правил и синтаксиса соответствующего приложения или служебной утилиты в сочетании с простыми управляющими операторами, такими как операторы циклов и условные операторы.

Пакетные файлы и сценарии часто называют командными файлами, содержащими любые команды. Некоторые команды, такие как **For**, **Goto** и **If**, позволяют выполнять обработку условий в пакетных файлах. Другие команды позволяют управлять вводом и выводом, а также запускать другие пакетные файлы.

При организации пакетных файлов и сценариев применяют **переменные**, задающие поведение командной оболочки или ОС и **пакетные параметры** командного интерпретатора, которые используются в пакетном файле для получения информации о настройках среды. Поведение среды командной оболочки или всей ОС задают с помощью двух типов переменных среды: **системных** и **локальных**.

- **Системные переменные** определяют поведение глобальной среды ОС.
- **Локальные переменные** определяют поведение среды в конкретном экземпляре командного интерпретатора *Cmd.exe*.

Системные переменные среды задаются заранее в ОС Windows XP и доступны для всех ее процессов. Только пользователи с привилегиями администратора могут изменять эти переменные.

Локальные переменные среды доступны в случае, когда пользователь, для которого они были созданы, входит в систему. В частности, локальные переменные реестра **HKEY_CURRENT_USER** подходят только для текущего пользователя, но определяют поведение глобальной среды ОС.

В следующем списке представлены различные типы переменных в порядке убывания их приоритета:

1. встроенные системные переменные,
2. системные переменные реестра **HKEY_LOCAL_MACHINE**,
3. локальные переменные реестра **HKEY_CURRENT_USER**,
4. все переменные среды и пути, указанные файле **Autoexec.bat**,
5. все переменные среды и пути, указанные в сценарии входа в систему, если он имеется,
6. переменные, используемые интерактивно в пакетном файле или сценарии.

Чтобы иметь возможность подставить значение в переменную среды из командной строки или в пакетном файле (сценарии), следует заключить имя соответствующей переменной (Приложение 1) в символы процентов (%), например **Set MyPath=%CD%**. Символы процентов указывают на то, что командный интерпретатор должен обратиться к значению переменной без посимвольного ее разложения и сравнения.

Командный интерпретатор **Cmd.exe** может оперировать переменными с %0 по %9. При использовании пакетных параметров переменная %0 заменяется именем пакетного файла, а переменные с %1 по %9 — на соответствующие аргументы командной строки. Для доступа к переменным больше %9 необходимо воспользоваться командой **Shift**. Параметр %* ссылается на все аргументы, которые передаются пакетному файлу, за исключением %0. В качестве примера, рассмотрим копирование содержимого из каталога 1 (*Folder1*) в каталог 2 (*Folder2*), где параметр %1 заменяется значением *Folder1*, а параметр %2 соответственно значением *Folder2*. В пакетном файле **Mybatch.bat** следует ввести следующую строку:

Xcopy %1*.* %2

Используйте пакетный файл **Mybatch.bat** следующим образом:

Mybatch.bat *C:\folder1* *D:\folder2*

Результат будет таким же, как и при записи в пакетный файл строки:

Xcopy *C:\folder1*.** *D:\folder2*

С пакетными параметрами можно также использовать модификаторы. **Модификаторы** используют информацию о текущем диске и каталоге как часть или полное имя файла (каталога).

Синтаксис модификатора: %~ху, где х — символьное сокращение действия, определяемое модификатором, у — идентификатор переменной (в диапазоне от 1 до 9). В табл. 1 и 2 описаны модификаторы, выполняемые ими действия, и даны возможные комбинации модификаторов и квалификаторов для получения более сложных результатов. В этих таблицах %1 и переменную среды **PATH** можно заменить другими значениями пакетных параметров.

Таблица 1. Модификаторы и выполняемые ими действия

	Модификатор	Описание
--	-------------	----------

№ п.п.		
1	%~1	расширение %1 и удаление любых кавычек (" ")
2	%~f1	замена %1 полным путем
3	%~d1	замена %1 именем диска
4	%~p1	замена %1 путем
5	%~n1	замена %1 именем файла
6	%~x1	замена %1 расширением имени файла
7	%~s1	замена путем, содержащим только короткие имена
8	%~a1	Замена %1 атрибутами файла
9	%~t1	замена %1 датой и временем модификации файла
10	%~z1	замена %1 размером файла
11	%~\$PATH:1	поиск в каталогах, перечисленных в переменной среды PATH, замена %1 полным именем первого найденного файла. Если переменная среды не определена или поиск не обнаружил файлов, модификатор выдает пустую строку.

Таблица 2. Комбинации модификаторов и квалификаторов

№ п.п.	Модификатор	Описание
1	%~dp1	замена %1 именем диска и путем
2		%~px1 замена %1 именем файла и расширением
3	%~dp\$PATH:1	поиск в каталогах, перечисленных в переменной среды PATH, и замена %1 именем диска и путем к первому найденному файлу.
4	%~ftza1	замена %1 строкой, аналогичной результату работы команды Dir

Еще один модификатор, являющийся уникальным, имеет вид %*. Он представляет все аргументы, переданные пакетному файлу. Этот модификатор не используется в комбинации с модификатором %~.

Конвейеры команд и «каналы», рассмотренные в предыдущих лабораторных работах (Часть I, Приложения 1 и 2) являются инструментами для расширения функционала пакетных файлов и сценариев при их построении и организации.

Сервер сценариев ОС Windows XP позволяет быстро запустить пакетный файл или сценарий, имя которого введено в командной строке оболочки.

Сервер сценариев

- служит контроллером средств обработки сценариев в ОС Windows XP;
- не требует много памяти;

- является идеальным средством, как для интерактивных, так и для пакетных сценариев.

Существуют две версии сервера сценариев, доступных в окне командной оболочки:

- **Wscript.exe** — позволяет задавать параметры выполнения сценариев в окне свойств;
- **Cscript.exe** — позволяет задавать параметры выполнения сценариев с помощью ключей командной строки.

Для разработки сценариев ОС Windows XP следует использовать редакторы сценариев **JScript** или **VBScript** (в составе **Visual Basic Scripting Edition**). При запуске сценария из командной строки, сервер сценария читает и передает содержимое указанного файла зарегистрированному обработчику сценариев. Для определения языка сценария используется расширение имени файла (**.vbs** для **VBScript**, **.js** для **JScript**). Благодаря этому, разработчик сценария не обязан знать точные программные идентификаторы (**ProgID**) различных обработчиков сценариев. Сопоставление расширения имени файла сценария с программным идентификатором и запуск конкретного обработчика сценариев осуществляется непосредственно сервером сценариев ОС Windows XP.

Простейшим сценарием, не требующим применения среды Visual Basic, является сценарий входа в систему, представляющий собой файл, связываемый с одной или несколькими учетными записями пользователей. Обычно сценарий входа является пакетным файлом, который автоматически выполняется при каждом входе пользователя в систему. Сценарии входа используются для настройки рабочей среды пользователя при входе и позволяют администратору задавать основные параметры рабочей среды пользователя без непосредственного его участия.

Поскольку пакетные файлы могут включать в себя любые команды, их конвейеры и «каналы», при большом количестве условий и циклов последствия некорректной работы пакетного файла могут быть непредсказуемыми для ОС, и возможно как следствие разрушительными. Поэтому для организации пакетного файла разработчику необходимо четко представлять себе, что именно и каким образом должно происходить в системе при работе этого файла, какая последовательность действий реализуется в результате выполнения задуманного сценария и как на эти действия реагирует ОС.

Помимо рассмотренных в предыдущих лабораторных работах команд, которые могут быть использованы при организации пакетного файла, существует ряд дополнительных, функционал которых напоминает операторы языков программирования высокого уровня. К их числу относятся: **At**, **Call**, **Doskey**, **Echo**, **Endlocal**, **For**, **Goto**, **If**, **Pause**, **Rem**, **Set**, **Setlocal** и **Shift**.

В настоящей лабораторной работе предполагается ознакомление с основными командами, используемыми в качестве инструментов организации пакетных файлов, создание командного файла в формате ASCII, реализующего определенный сценарий работы системы, а также оценка

возможности использования его в качестве сценария входа в систему. Лабораторная работа выполняется на виртуальной машине в среде ОС Windows XP.

Задание 1. Команда Set

Команда **Set** предназначена для отображения и установки переменных среды в ОС Windows XP.

Синтаксис

команды **Set**:

Set [*переменная*=[*строка*]],

где

параметр:

переменная — задает имя переменной, значение которой требуется присвоить или изменить.

строка — задает строковое значение для указанной переменной.

Команда SET поддерживает два дополнительных ключа:

SET /A *выражение*

SET /P variable=[promptString]

Ключ /A указывает, что строка справа от знака равенства является числовым выражением, значение которого вычисляется. Обработчик выражений очень прост и поддерживает следующие операции, перечисленные в порядке убывания приоритета (Табл. 3).

Таблица 3. Операции обработчика выражений команды Set

№ п.п.	Операция	Описание
1	()	группировка
2	! ~ -	унарные операторы
3	* / % +	арифметические операторы
4	<< >>	двоичный сдвиг
5	&	двоичное И
6	^	двоичное исключающее ИЛИ
7		двоичное ИЛИ
8	= *= /= %= += -= &= ^= = <<= >>=	присвоение
9	,	разделитель операторов

При использовании любых логических или двоичных операторов необходимо заключить строку выражения в кавычки. Любые нечисловые строки в выражении рассматриваются как имена переменных среды, значения которых преобразуются в числовой вид перед использованием. Если переменная с указанным именем не определена в системе, вместо нее подставляется нулевое значение. Это позволяет выполнять арифметические операции со значениями переменных среды, причем не нужно вводить знаки % для получения значений. Если команда SET /A вызывается из командной строки, а не из пакетного файла, она выводит окончательное значение выражения. Слева от любого оператора присваивания должно стоять имя переменной

среды. Числовые значения рассматриваются как десятичные, если перед ними не стоит префикс 0x для шестнадцатеричных чисел, и 0 для восьмеричных чисел. Например, числа 0x12, и 022 обозначают десятичное число 18. Обратите внимание на запись восьмеричных числе: 08 и 09 не являются допустимыми числами, так как в восьмеричной системе исчисления цифры 8 и 9 не используются. Если команда с ключом /a вызывается из командной строки, а не из пакетного файла, она выводит окончательное значение выражения. Ключ /P позволяет установить значение переменной для входной строки, введенной пользователем. Показывает указанное приглашение promptString перед чтением введенной строки. Приглашение promptString может быть пустым. Подстановка переменной среды может быть расширена следующим образом:

```
%PATH:str1=str2%
```

расширит действие переменной среды PATH, заменяя каждое вхождение "str1" в расширенном результате на "str2". "str2" может быть пустой строкой для эффективного удаления вхождений "str1" из расширенного вывода. "str1" может начинаться со звездочки, и в этом случае это будет соответствовать любому началу расширенного вывода до первого вхождения оставшейся части "str1". Можно также использовать строки расширения.

%PATH:~10,5%

расширит действие переменной среды PATH, затем использует только 5 символов, которые начинаются с 11-го символа (пропустив 10) расширенного результата. Если длина не указана, по умолчанию используется оставшееся значение переменной. Если оба значения (длина и число пропускаемых символов) отрицательны, следующим используемым значением будет длина значения переменной среды, добавленной к указанному значению пропуска или указанной длины.

%PATH:~-10%

извлекает	последние	10	символов	переменной	PATH.
%PATH:~0,-2%					

извлекает все символы переменной PATH, за исключением 2-х последних. Наконец, добавлена поддержка связывания времени выполнения для переменных среды окружения. По умолчанию эта поддержка отключена. Ключ /V командной строки CMD.EXE позволяет включать и выключать ее. Связывание времени выполнения для переменных среды окружения полезно при обходе ограничений раннего связывания, которое происходит при первом чтении текстовой строки, а не при ее выполнении. Следующий пример демонстрирует возникающую проблему при использовании раннего связывания переменных:

```
set VAR=before
```

```
if "%VAR%" == "before" (
```

```
set                                     VAR=after
if  "%VAR%" ==  "after"  @echo  Тело  внутреннего  оператора  сравнения
)
```

Данное сообщение не будет выводиться, т.к. %VAR% в ОБОИХ выражениях if подставляется в момент первого использования в первом if, в том числе и в тело первого ветвления if, которое является составным выражением. В if внутри составного выражения в действительности сравниваются значения "before" и "after", что заведомо ложно. Следующий пример демонстрирует подобную ошибку:

```
set                                     LIST=
for  %i  in  (*)  do  set  LIST=%LIST%  %i
echo                                     %LIST%
```

в данном случае список файлов текущей папки никогда не будет построен. Вместо этого, значением переменной LIST будет имя последнего найденного файла. И вновь, это случилось потому, что %LIST% подставляется всего один раз - в момент обработки выражения FOR, когда список еще пуст. Фактически, приведенный фрагмент эквивалентен следующему примеру:

```
for  %i  in  (*)  do  set  LIST=  %i
```

в котором имя последнего найденного файла сохраняется в переменной LIST. Связывание времени выполнения для переменных среды окружения происходит при использовании специального символа (восклицательного знака), обозначающего проведение сопоставления во время выполнения. Если включена поддержка связывания времени выполнения, то для достижения ожидаемых результатов приведенные выше фрагменты должны быть изменены следующим образом:

```
set                                     VAR=before
if  "%VAR%" ==  "before"  (
set                                     VAR=after
if  "!VAR!" ==  "after"  @echo  Тело  внутреннего  оператора  сравнения
)
set                                     LIST=
for  %i  in  (*)  do  set  LIST=!LIST!  %i
echo                                     %LIST%
```

При включенной расширенной обработке команд доступны несколько переменных среды, которые не отображаются в списке, отображаемом при вызове команды SET. Значения этих переменных вычисляются динамически каждый раз при их извлечении. Если пользователь явно задаст переменные с одним из этих имен, то это определение перекрывает соответствующее динамическое определение, описанное ниже:

%CD% - раскрывается в строку текущей директории.
 %DATE% - раскрывается в текущую дату, используя тот же формат команды DATE.
 %TIME% - раскрывается в текущую дату, используя формат команды TIME.
 %RANDOM% - раскрывается в случайное десятичное число между 0 и 32767.
 %ERRORLEVEL% - раскрывается в текущее значение ERRORLEVEL
 %CMDEXTVERSION% - раскрывается в текущее значение версии расширенной обработки команд.
 %CMDCMDLINE% - раскрывается в исходную командную строку, которая вызвала командный процессор.

Таблица 4. Динамические значения команды Set

Значение	Описание действия
	%Cd% раскрывается в строку текущей директории
	%Date% раскрывается в текущую дату
	%Time% раскрывается в текущее время
%Random%	раскрывается в случайное десятичное число в диапазоне от 0 до 32767
%Errorlevel%	раскрывается в текущее значение ErrorLevel
%Cmdextversion%	раскрывается в текущее значение версии расширенной обработки команд
%Cmdcmdline%	раскрывается в исходную командную строку, которая вызвала текущее окно командной оболочки

Содержание задания

1. Отобразите переменные среды двумя способами: из командной оболочки и окна свойств системы (Пуск | Панель управления | Система).
2. Задайте переменную среды, содержащую определенный путь к месту назначения, выбранный самостоятельно.
3. Проверьте наличие в системе переменной среды, заданной в предыдущем пункте задания.
4. Выведите значение выражения, определенного в соответствии с вариантом задания (табл. 9), в качестве переменной среды Result.
5. Задайте переменную среды с различными вариантами динамически формируемых значений (табл. 4). Варианты динамических значений выберите самостоятельно.

При выполнении задания используйте следующие инструкции:

- по каждому из пунктов задания в окне командной оболочки наберите соответствующую команду с необходимыми ключами,

- нажмите **Enter** для ввода,
- изучите полученный результат и сделайте вывод о проделанной работе,
- запишите полученную информацию в отчет, заполнив табл. 5.

Таблица 5. Результаты выполнения команды Set

№ п.п.	Команда с ключами	Результат и вывод по способу применения команды
1		
2		
3		
4		
5		

Задание 2. Команды Rem, Echo

Команда **Rem** добавляет комментарии в пакетных файлах и не выводит комментарии на экран. При обработке пакетной программы строки, начинающиеся с Rem, пропускаются.

Синтаксис

команды **Rem**:

Rem [*текст*],

где

параметр:

текст — задает строку символов, используемую в качестве комментария.

Команда **Echo** выводит на экран сообщения или задание режима вывода на экран.

Синтаксис

команды **Echo**:

Echo [{**on** | **off**}]

[*сообщение*],

где

параметр:

{**on** | **off**} — включает или отключает режим отображения на экране информации о работе команд.

сообщение — задает текст для вывода на экран.

Для вывода сообщений из нескольких строк без вывода дополнительных команд между ними следует использовать несколько последовательных команд **Echo** после команды **Echo off**:

Cls

@Echo

off

Echo.

Rem *****

Эта

пакетная

программа

Rem *****

иллюстрирует

возможности

Rem *****

команды

Echo

Echo.

Echo *****

This

batch

program

Echo *** illustrates possibilities of ***
Echo ***** the Echo command *****
Echo.

Pause

Содержание задания

1. Создайте пакетный файл, воспользовавшись любым текстовым редактором. Имя пакетного файла выберете самостоятельно.
2. Введите в созданный пакетный файл текст, приведенного выше примера.
3. Сохраните текст пакетного файла.

При выполнении задания используйте следующие инструкции:

- воспользовавшись командой Start и указав путь к пакетному файлу,
- запустите его на выполнение, нажав Enter для ввода,
- изучите пример и полученный с его помощью результат, обратив внимание на то, что команда Echo с точкой (.) в конце выводит на экран пустую строку, а символ «коммерческое И» (@) перед командой Echo отключает режим отображения команд.
- сделайте соответствующий вывод и запишите его в отчет.

Задание 3. Утилита For

Утилита **For** командной среды предназначена для выполнения циклических операций для заданного множества операндов.

Синтаксис команды **For**:

For {*% переменная* | *%%переменная*} in (*множество*)
do команда [*ПараметрыКоманднойСтроки*]

где параметр:

%переменная - любые символы, кроме цифр 0–9 используется для выполнения команды из строки в окне командной оболочки;

%%переменная - любые символы, используется для выполнения команды в пакетном файле.

Переменные должны учитывать регистр.

(*множество*) — обязательный параметр, задающий один или группу файлов, каталогов, диапазон значений или текстовых строк, указывающих на файлы, подлежащие обработке заданной командой. Например, допустимыми являются следующие варианты: (*.doc), (*.doc *.txt *.me), (jan*.doc jan*.rpt feb*.doc feb*.rpt), (ar??1991.* ar??1991.*). Скобки являются обязательными для обозначения множества.

команда — обязательный параметр, задающий команду, которая будет выполнена для каждого операнда множества.

ПараметрыКоманднойСтроки — задает параметры командной строки, которые используются в указанной командой.

Если расширения командного интерпретатора разрешены, что делается по умолчанию, то существуют дополнительные формы команды **For**.

For /D {% *переменная* | %%*переменная*} in (*множество*)
do команда [*ПараметрыКоманднойСтроки*]

Если множество содержит подстановочные знаки, команда выполняется для всех подходящих имен каталогов, а не имен файлов.

For /R [[*диск* :]*путь*] {% | %%} *переменная* in (*множество*)
do команда [*ПараметрыКоманднойСтроки*]

Выполнение команды для каталога [*диск* :]*путь*, а также для всех подкаталогов этого пути. Если после ключа /R не указано имя каталога, выполнение команды начинается с текущего каталога.

Если вместо набора указана только точка (.), команда выводит список всех подкаталогов.

For /L {% | %%} *переменная* in (*НачальноеЗначение*#,*шаг*#, *КонечноеЗначение*#)
do команда [*ПараметрыКоманднойСтроки*]

Множество раскрывается в последовательность чисел с заданными началом, концом и шагом приращения. Так, набор (1,1,5) раскрывается в (1 2 3 4 5), а набор (5,-1,1) заменяется на (5 4 3 2 1)

FOR /F ["ключи"] %*переменная* IN (*множество*) **DO** команда [*параметры*]

FOR /F ["options"] %variable IN ("literal string") **DO** command [command-parameters]

FOR /F ["options"] %variable IN ('command') **DO** command [command-parameters]

или, если использован параметр **usebackq**:

FOR /F ["options"] %variable IN (filenameset) **DO** command [command-parameters]

FOR /F ["options"] %variable IN ('literal string') **DO** command [command-parameters]

FOR /F ["options"] %variable IN (`command`) **DO** command [command-parameters]

Набор содержит имена одного или нескольких файлов, которые по очереди открываются, читаются и обрабатываются. Обработка состоит в чтении файла, разбивки его на отдельные строки текста и выделения из каждой строки заданного числа подстрок (в том числе нуля).

Затем найденная подстрока используется в качестве значения переменной при выполнении основного тела цикла. По умолчанию ключ /F выделяет из каждой строки файла первое слово, очищенное от окружающих его пробелов. Пустые строки в файле пропускаются.

Необязательный параметр "ключи" служит для переопределения заданных по умолчанию правил обработки строк. Ключи представляют собой заключенную в кавычки строку, содержащую указанные параметры. Ключевые слова:

eo! = c - определение символа комментариев в конце строки (допускается

задание только одного символа).

`skip=n` - число пропускаемых при обработке строк в начале файла.

`delims=xxx` - определение набора разделителей для замены заданных по умолчанию пробела и знака табуляции.

`tokens=x,y,m-n` - определение номеров подстрок, выделяемых из каждой строки файла и передаваемых для выполнения в тело цикла. При использовании этого ключа создаются дополнительные переменные. Формат `m-n` представляет собой диапазон подстрок с номерами от `m` по `n`. Если последний символ в строке `tokens=` является звездочкой, создается дополнительная переменная, значением которой будет весь оставшийся текст в строке после обработки последней подстроки.

`usebackq` - применение новой семантики, при которой строки, заключенные в обратные кавычки, выполняются как команды, строки, заключенные в прямые одиночные кавычки, являются строкой литералов команды, а строки, заключенные в двойные кавычки, используются для выделения имен файлов в списках имен файлов.

Поясняющий

пример:

FOR /F "eol=; tokens=2,3* delims=, " %i in (myfile.txt) do @echo %i %j %k

Эта команда обрабатывает файл `myfile.txt`, пропускает все строки, которые начинаются с символа точки с запятой, и передает вторую и третью подстроки из каждой строки в тело цикла, причем подстроки разделяются запятыми и/или пробелами. В теле цикла переменная `%i` используется для второй подстроки, `%j` - для третьей, а `%k` получает все оставшиеся подстроки после третьей.

Имена файлов, содержащие пробелы, необходимо заключать в двойные кавычки. Для того чтобы использовать двойные кавычки, необходимо использовать параметр `usebackq`, иначе двойные кавычки будут восприняты как границы строки для обработки.

Переменная `%i` явно описана в инструкции `for`, а переменные `%j` и `%k` описываются неявно с помощью ключа `tokens=`. Ключ `tokens=` позволяет извлечь из одной строки файла до 26 подстрок, при этом, не допускается использование переменных больших чем буквы 'z' или 'Z'. Следует помнить, что имена переменных **FOR** являются глобальными, поэтому одновременно не может быть активно более 52 переменных.

Синтаксис команды **FOR /F** также позволяет обработать отдельную строку, с указанием параметра `filenameset`, заключенным в одиночные кавычки. Строка будет обработана как единая строка из входного файла.

Наконец, команда **FOR /F** позволяет обработать строку вывода другой команды. Для этого следует ввести строку вызова команды в апострофах вместо набора имен файлов в скобках. Строка передается для выполнения обработчику команд `CMD.EXE`, а вывод этой команды

записывается в память и обрабатывается так, как будто строка вывода взята из файла.

Например, следующая команда:

```
FOR /F "usebackq delims==" %i IN (`set`) DO @echo %i
```

выведет перечень имен всех переменных среды, определенных в настоящее время в системе.

Операции подстановки ссылок на переменные команды FOR также расширены. Допускается

применение следующих синтаксических конструкций:

%~I - из переменной %I удаляются обрамляющие кавычки ("")

%~fI - переменная %I расширяется до полного имени файла

%~dI - из переменной %I выделяется только имя диска

%~pI - из переменной %I выделяется только путь к файлу

%~nI - из переменной %I выделяется только имя файла

%~xI - из переменной %I выделяется расширение имени файла

%~sI - полученный путь содержит только короткие имена

%~aI - переменная %I расширяется до атрибутов файла

%~tI - переменная %I расширяется до даты /времени файла

%~zI - переменная %I расширяется до размера файла

%~\$PATH:I - проводится поиск по каталогам, заданным в переменной среды PATH, и переменная %I заменяется на полное имя первого найденного файла. Если переменная PATH не определена или в результате поиска не найден ни один файл, то этот модификатор заменяется на пустую строку

При объединении нескольких операторов можно получить следующие результаты:

%~dpI - переменная %I заменяется только на имя диска и путь

%~nxI - переменная %I заменяется только на имя файла и его расширение

%~fsI - переменная %I заменяется только на полный путь с краткими именами

%~dp\$PATH:I - проводится поиск по каталогам, заданным в переменной среды PATH, и переменная %I заменяется на имя диска и путь к первому найденному файлу.

%~ftzaI - переменная %I заменяется на строку, выдаваемую командой DIR

В приведенных выше примерах переменные %I и PATH можно заменить на любые другие допустимые значения. В таких синтаксических конструкциях за символами %~ может следовать

любое имя переменной цикла FOR.

Рекомендуется использовать заглавные буквы для имен таких переменных, например, %I, что делает их более заметными, а также позволяет отличать их от обозначений модификаторов, которые не различают используемый регистр.

Содержание задания

- Скопируйте файлы каталога, путь к которому задайте самостоятельно, в точку

назначения, заданную путем d:\Temp\. При копировании воспользуйтесь любым методом, изученным ранее.

- К каждому из файлов, местоположение которых определено путем d:\Temp\, добавьте символ «!» в начале имени, воспользовавшись командой циклической обработки данных.
- Подсчитать количество каталогов на локальном диске, воспользовавшись командой циклической обработки данных, в процессе выполнения выводя результат в переменную среды, выбранную самостоятельно.
- Модифицируйте пакетный файл, полученный в предыдущем задании, воспользовавшись командой циклической обработки данных таким образом, чтобы в процессе его выполнения отображалось определенное количество раз выражение «***** the For command *****».

При выполнении задания используйте следующие инструкции: по каждому из пунктов задания в окне командной оболочки наберите соответствующую команду с необходимыми ключами, нажмите Enter для ввода, изучите полученный результат и сделайте вывод о проделанной работе, запишите полученную информацию в отчет, заполнив табл. 6.

Таблица 6. Результаты выполнения команды For

№ п.п.	Команда с ключами	Результат и вывод по способу применения команды
1		
2		
3		
4		
5		

Задание 4. Команда Goto

Команда Goto, эквивалентная безусловному переходу в языке высокого уровня, в пакетной программе реализует передачу правления ОС Windows XP в строку, определенную символьной меткой. Когда метка найдена, выполнение продолжается со следующей за ней строки. Команда Goto неразрывно связана с описанной выше командой If, в совокупности обеспечивающей возможность сложных программных конструкций с переходами.

Синтаксис

команды

Goto:

Goto метка,

где

параметр:

метка это строка в пакетной программе, на которую выполняется переход. В пакетном файле она должна начинаться с двоеточия (:). Если строка начинается с двоеточия, все присутствующие в

ней команды обработаны не будут. Синтаксически она может включать пробелы, но не может включать другие разделители, такие как точка с запятой или знак равенства. При этом используются только первые восемь знаков метки (метки: `hithere0`, `hithere01` и `hithere02` воспринимаются интерпретатором команд как эквивалентные).

Если расширения командного интерпретатора разрешены и в команде Goto используется метка :Eof, управление будет передано в конец текущего пакетного файла для выхода из него без назначения метки. В синтаксис команды обязательно должно быть включено двоеточие (:), то есть

Goto	:Eof.
------	-------

Дополнительная информация по данной команде, а также примеры ее использования доступны в справке ОС Windows XP (Пуск | Справка и поддержка) в соответствующем разделе. Справку также можно получить, набрав в окне командной оболочки строку `Goto /?` и нажав Enter для ввода.

Содержание задания

- Модифицируйте существующий пакетный файл, введя в него следующий текст:

Pause

Echo.

Format

A:

If not

Errorlevel

1

Goto

End

Echo.

Echo *** Error of formatting

Rem *** Ошибка форматирования

:End

Echo.

Echo *** The end of batch program

Rem *** Конец пакетной программы

Echo.

Pause

- Сохраните текст пакетного файла.

При выполнении задания используйте следующие инструкции: воспользовавшись командой **Start** и указав путь к пакетному файлу,

- запустите его на выполнение, нажав **Enter** для ввода,
- изучите пример и полученный с его помощью результат,
- сделайте вывод о проделанной работе и запишите его в отчет.

Задание 5. Команда Call

Вызов одного пакетного файла из другого без завершения его выполнения осуществляется

командой Call. Эта команда эквивалентна вызову процедуры из основного тела программы. Она принимает метки в качестве объекта вызова и используется только в сценариях или пакетных файлах; при вызове из командной строки команда Call игнорируется.

Синтаксис команды Call:

Call [[диск:][путь] имя_файла [пакетные_параметры]] [:метка [аргументы]],

где параметр:

[диск:][путь] имя_файла — задает имя и расположение пакетного файла.
 пакетные_параметры — задает данные командной строки, используемые программой пакетной обработки, включая параметры командной строки, имена файлов, пакетные параметры (%0-%9) или переменные (например, %baud%).

:метка — указывает метку, на которую должно быть передано управление программы пакетной обработки. При использовании с этим параметром создается новый контекст пакетного файла, а управление передается инструкции, следующей за указанной меткой.

аргументы — задает данные командной строки, которые передаются в новую программу пакетной обработки, начинающуюся с :метки, включая параметры командной строки, имена файлов, пакетные параметры или переменные.

Необходимо отметить, что при использовании команды Call символы перенаправления ввода-вывода и «каналы» не допускаются. Кроме того, может быть реализована подстановка переменных на основе модификаторов. При этом разрешаются все варианты синтаксических конструкций, приведенных в табл. 1 и 2.

Дополнительная информация по данной команде, а также примеры ее использования доступны в справке ОС Windows XP (Пуск | Справка и поддержка) в соответствующем разделе. Справку также можно получить, набрав в окне командной оболочки строку Call /? и нажав Enter для ввода.

Содержание задания

- Создайте новый (дочерний) пакетный файл, воспользовавшись любым текстовым редактором. Имя пакетного файла выберите самостоятельно.
- Введите в дочерний пакетный файл процедуру форматирования гибкого диска, учитывающую переход в начало процедуры в случае ошибки, из приведенного выше примера.
- Модифицируйте родительский пакетный файл, удалив из него лишние команды и добавив ссылку на дочерний пакетный файл для его вызова.
- Сохраните тексты обоих пакетных файлов.

При выполнении пунктов 1-4 задания используйте следующие инструкции:

- воспользовавшись командой Start и указав путь к родительскому файлу, запустите его на выполнение, нажав Enter для ввода,
- изучите полученный результат и сделайте вывод о проделанной работе,

- запишите полученную информацию в отчет.
- 5. Вспомните команду форматирования **Format** и ее параметры.
- 6. Модифицируйте родительский и дочерний файлы таким образом, чтобы осуществилась передача из родительского файла двух значений параметров (%переменная) команды Format (табл. 9), находящейся внутри дочернего файла. Обратите внимание на то, что в таблице 9 передаваемые параметры команды Format имеют числовое (%0-%9), а не символьное представление.
- 7. Сохраните тексты обоих пакетных файлов.

При выполнении пунктов 5-7 задания используйте следующие инструкции:

- воспользовавшись командой Start и указав путь к родительскому файлу с параметрами для команды Format, запустите его на выполнение, нажав Enter для ввода,
- изучите полученный результат и сделайте вывод о проделанной работе,
- перенесите тексты модифицированных пакетных файлов, а также значения используемых пакетных параметров в отчет.

Таблица 9. Варианты для заданий.

Вар. №	Set.				Format	
	Переменная среды				Параметры команды Format	
	a	b	c	Result	%1	%2
1	24	11	35	$(a + b - c) * 10$	/v:System	/a:512
2	AA	01	C1	$a * 5 - b / 5 + c$	/a:512	/q
3	12	33	10	$a * 4 / c - b * 2$	/v:IICT	/a:1024
4	25	A3	B4	$a - b * 3 - c / 2$	/a:1024	/q
5	49	02	65	$a - b * b + c / 5$	/v:VMgroup	/a:2048
6	21	99	12	$(b * (a + c)) / 3$	./a:2048	/q
7	BC	BC	CB	$10 + (a - b) * c$	/v:MyCore	/a:4096
8	01	94	04	$(b - c) / (a * 9)$	/a:4096	/q
9	84	D2	2A	$a / 10 - (b * c)$	/v:Useless	/a:8192
10	10	39	92	$a * a + b - c / 2$	/v:MyDisk	/q
11	D1	CC	1C	$a * b * c - b / c$	/v:Temp	/q
12	FF	00	F1	$(a - c) * b + 25$	/q	/v:Apps
13	CA	DA	FA	$a * b * (FA - c)$	/v:Double	/q
14	45	78	87	$((c - b) * a) / 8$	/q	/v:HomeUse
15	88	88	00	$(a - b) / (1 - c)$	/v:MyDocs	/q
16	75	93	02	$(b + c) * 2 - a$	/q	/v:VMComp
17	A1	CD	0E	$a * a * b * b * c * c$	/v:MyList	/a:512

18	C4	EA	E3	$(a + b + c) * a$	/a:512	/q
19	44	55	33	$c * (b - a) / 10$	/v:Admin	/a:1024
20	B3	E5	DE	$(a * b * c) / 25$	/a:1024	/q
21	BB	ED	AE	$(a - b) * 2 - c$	/v:SysCore	/a:2048
22	55	56	31	$(b - a) * 31 - c$	/a:2048	/q
23	D1	EC	EE	$(c - b) * 5 / a$	/v:Kernel	/a:4096
24	D6	E6	FE	$(a + b) * c - 11$	/a:4096	/q
25	71	65	32	$(a - b) * c / 32$	/v:User	/a:8192
26	84	32	10	$(a * (b + c)) / 5$	/a:8192	/q

Контрольные вопросы:

- 1.Перечислить системные процессы Windows.
- 2.Дать определения драйвера устройства в различных контекстах.
- 3.Пояснить принцип функционирования уровня абстрагирования от оборудования.
- 4.Перечислить основные компоненты исполнительной системы Windows.
- 5.Дать определения подсистемам Windows.
- 6.Перечислить возможные места хранения назначенных политик безопасности
- 7.Отобразить переменные среды в Windows двумя способами: из командной оболочки и окна свойств системы
- 8.Задать переменную среды с различными вариантами динамически формируемых значений.

Список литературы, рекомендуемый к использованию по данной теме:

Основная: 1-4

Дополнительная: 1-3

Интернет-ресурсы: 1

Методическая: 1-3

Программное обеспечение: 1-4

Практическая работа №6

Тема: Архитектура ОС Linux Архитектура ОС Windows

1. Подготовка к занятию

1. Изучить (повторить) теоретический материал.
2. Ознакомиться с программой работы.
3. Подготовить отчет о работе.
4. Ответить на контрольные вопросы.

2. Распределение времени занятия:

Всего: 90 мин

Вступительная часть 2 мин

Проверка готовности студентов к занятию 5 мин

Основная часть – 70 мин

Проверка выполнения практического занятия 10 мин

Заключительная часть 3 мин

3. Формируемые компетенции: ПК-1, ПК-3

4. Правила работы в лаборатории

К работе в лаборатории допускаются лица, изучившие правила и меры безопасности, сдавшие зачет по ним и усвоившие порядок выполнения лабораторной работы.

4.1. Требования безопасности перед началом работ

- ЗАПРЕЩАЕТСЯ: переодеваться, пользоваться огнем, курить, принимать пищу в лаборатории.

- Убедиться в целостности электрических розеток и разъемов. В лаборатории необходимо быть в сменной обуви.

- Включение компьютера производить только после получения допуска по выполняемой работе и разрешения преподавателя или лаборанта.

4.2. Требования безопасности во время работы

- выполняя практическое занятие, студенты обязаны использовать только вычислительную технику, периферийное оборудование, соединительные кабели, измерительное оборудование и носители информации, непосредственно относящиеся к данному лабораторному занятию;

- подключение и отключение составляющих вычислительного комплекса производить только при полном снятии напряжения со всех составляющих вычислительного комплекса;

- при обнаружении неисправностей в оборудовании немедленно отключить источники питания и доложить об этом руководителю занятий или лаборанту.

4.3. Требования безопасности по окончании работы

- доложить руководителю занятий или лаборанту о завершении работ;

- привести в порядок и сдать рабочее место лаборанту, и доложить руководству

1. Ознакомление с оснасткой «Локальные пользователи и группы»

Порядок выполнения: Первым шагом при создании пользовательской консоли администрирования является придание ей функций контроля и управления учетными записями пользователей и групп. В среде ОС Windows XP учетные записи локальных пользователей и групп управляются одноименной оснасткой «Локальные пользователи и группы».

Учетная запись пользователя содержит все сведения, определяющие пользователя. К этим сведениям относятся имя пользователя и пароль, требуемые для входа в систему, имена групп, членом которых он является, а также права и разрешения, которые он имеет при работе и доступе к системным ресурсам системы.

Существует два основных типа учетных записей пользователей, доступных на компьютере с ОС Windows XP: учетная запись администратора и учетная запись с ограниченными правами. Кроме того, штатная учетная запись гостя доступна для пользователей, не имеющих собственных учетных записей в системе.

Учетная запись администратора предназначена для тех уполномоченных пользователей, кто может вносить изменения на уровне системы, устанавливать приложения и иметь полный доступ ко всем файлам и другим учетным записям. Она позволяет:

- создавать, изменять и удалять любые учетные записи,
- создавать и изменять имена и пароли пользователей,
- обеспечить наличие, по крайней мере, одного пользователя с учетной записью администратора, когда уже нет других учетных записей с подобными правами.

Учетная запись с ограниченными правами предназначена для пользователей, которым запрещается изменять большинство настроек системы, а также удалять важные файлы. Пользователь с такой учетной записью:

- не может устанавливать приложения и оборудование, но имеет доступ к уже существующим программам в системе,
- не может изменять имя и тип собственной учетной записи,
- может создавать, изменять или удалять собственный пароль,
- может изменять рисунок, назначенный учетной записи.

Необходимо помнить, что некоторые приложения могут работать некорректно для пользователей с ограниченными правами. В этом случае следует сменить тип учетной записи на администратора.

Учетная запись гостя предназначена для пользователей, не имеющих собственных учетных записей в системе. У записи гостя нет пароля, что позволяет быстро входить в систему, в частности, для проверки электронной почты или просмотра Интернета. Пользователь с учетной записью гостя:

- не может устанавливать приложения и оборудование, но имеет доступ к уже

существующим программам в системе,

- не может изменить собственной тип учетной записи,
- может изменять рисунок, назначенный учетной записи.

Совокупность пользователей, компьютеров и контактов называется группой. Существуют группы безопасности, которые используются для управления доступом или в качестве списков рассылки и группы распространения, применяемые только в электронной почте.

Набор учетных записей пользователей представляет собой учетную запись группы. При включении учетной записи пользователя в определенную группу соответствующий пользователь получает все права и разрешения, предоставленные этой группе.

Содержание задания

- 1.Создайте пользовательскую консоль администрирования одним из изученных ранее способов и сконфигурируйте ее должным образом, приняв во внимание ее целевую принадлежность.
- 2.Добавьте оснастку «Локальные пользователи и группы» в корень консоли администрирования.
- 3.Создайте две учетные записи для двух разных пользователей. Имена, описание и пароли выберите самостоятельно.
- 4.Установите флажки «Потребовать смену пароля при следующем входе в систему» для первого пользователя и «Запретить смену пароля пользователем» для второго.
- 5.Создайте локальную группу, имя и описание которой выберите самостоятельно.
- 6.Поместите в новую локальную группу созданных ранее пользователей, воспользовавшись диалоговым окном Свойства каждого из них.
- 7.Поместите в новую локальную группу пользователя Администратор, воспользовавшись диалоговым окном Свойства этой группы (см. пункт 5 секции). Обратите внимание на то, что в этой локальной группе уже присутствуют учетные записи двух созданных ранее пользователей.
- 8.Выбрав самостоятельно имя и путь к месту расположения, сохраните и закройте консоль администрирования ММС.

При выполнении заданий секции используйте следующие инструкции:

- перенесите последовательность выполняемых действий по каждому из пунктов 1-8 в отчет (возможно приведение графических фрагментов, сделанных с экрана, в качестве демонстрационного материала),
- дважды войдите в систему под именами пользователей, соответствующих созданным учетным записям,
- обратите внимание на то, каким образом осуществляется влияние установленных в пункте 4 флажков на работу компьютера при смене пользователя,
- сделайте вывод о проделанной работе и запишите его в отчет.

2. Взаимосвязь утилиты «Учетные записи пользователей» с оснасткой «Локальные пользователи и группы» при смене типа учетной записи

Содержание задания

1. В Панели управления загрузите утилиту «Учетные записи пользователей», позволяющую изменять параметры и пароли учетных записей пользователей непосредственно в ОС Windows XP.
2. Загрузите пользовательскую консоль администрирования, созданную в Секции А текущего задания, с возможностью ее редактирования.
3. Создайте новую ограниченную учетную запись, воспользовавшись утилитой «Учетные записи пользователей».
4. Перейдите в оснастку «Локальные пользователи и группы» загруженной консоли администрирования и отметьте, в какую из локальных групп была по умолчанию определена вновь созданная учетная запись.
5. В окне утилиты «Учетные записи пользователей» измените тип данной учетной записи с ограниченного на административный.
6. Снова перейдите в оснастку «Локальные пользователи и группы» загруженной консоли администрирования и убедитесь в том, что смена типа учетной записи привела к перемещению пользователя из группы Пользователи в Администраторы, тем самым, расширяя его права.
7. Удалите этого пользователя из групп Пользователи и Администраторы.
8. Сохраните и закройте консоль администрирования ММС.

При выполнении заданий используйте следующие инструкции:

- перенесите последовательность выполняемых действий по каждому из пунктов 1-8 в отчет (возможно приведение графических фрагментов, сделанных с экрана, в качестве демонстрационного материала),
- обратите внимание на то, каким образом осуществляется взаимосвязь утилиты «Учетные записи пользователей» с оснасткой «Локальные пользователи и группы» при смене типа учетной записи,
- сделайте вывод о проделанной работе и запишите его в отчет.

3. Возможности оснастки «Локальные пользователи и группы» при работе с профилями пользователей

Профиль пользователя определяет параметры конфигурации рабочей среды, например, индивидуальную настройку экрана, сетевых подключений и доступа к принтеру. Эти параметры могут быть заданы самим пользователем или системным администратором. Существуют

следующие виды профилей пользователя:

- локальный профиль, представляющий собой машинную запись об авторизованном пользователе, автоматически создаваемую на компьютере при первом входе на рабочую станцию или сервер, и хранящийся на локальном жестком диске. Любые изменения локального профиля действительны только в рамках компьютера, на котором они произведены.
- перемещаемый профиль это находящийся на сервере профиль пользователя, который загружается на локальный компьютер при входе в систему и обновляется на локальном компьютере и сервере при выходе из нее. Если локальная копия профиля является более новой, чем копия на сервере, то пользователь имеет возможность использовать ее при очередном входе в систему.
- обязательный профиль это перемещаемый профиль, в котором можно задать конкретные параметры для отдельных пользователей или их группы. Этот профиль загружается всякий раз, когда осуществляется вход в систему и не обновляется при выходе из нее. Он может быть изменен только членом группы Администратор.

Содержание задания

1. Загрузите пользовательскую консоль администрирования с возможностью ее редактирования.
2. На локальном диске D: создайте папку, выбрав ее имя самостоятельно, для хранения в ней профилей пользователей. Эта папка верхнего уровня, которая будет содержать индивидуальные папки профилей пользователей.
3. Создайте вложенную внутрь папку, предназначенную для непосредственного хранения профиля пользователя.
4. Скопируйте профиль Администратора в созданную в пункте 3 секции папку, воспользовавшись системными средствами ОС Windows XP в окне свойств системы (Панель управления | Система). При этом в диалоговом окне Копирование профиля разрешите использование для Всех пользователей.
5. В существующую консоль добавьте локальную оснастку «Общие папки». В доступных командах вложенной директории «Общие ресурсы» оснастки кликните Новый общий ресурс... и, следуя инструкциям Мастера создания общей папки, откройте Всем пользователям полный доступ для директорий, созданных в пунктах 2 и 3 секции. При этом новый общий ресурс должен появиться в столбце «Общая папка» в области сведений консоли администрирования. На вкладке Безопасность диалогового окна Свойства новых общих ресурсов нажмите кнопку Дополнительно и во вновь появившемся диалоговом окне установите флажок Наследовать от родительского объекта...

6. В виде профиля пользователя подключите только что скопированный в нестандартную папку профиль Администратора к одному из ранее созданных в Секции А пользователей, воспользовавшись его диалоговым окном Свойства.

7. В виде домашней папки подключите сетевой диск «Н» на нестандартную папку с профилем Администратора к другому из ранее созданных в Секции А пользователей, воспользовавшись его диалоговым окном Свойства.

8. Создайте Новый вид панели задач таким образом, чтобы имелась возможность добавлять нового пользователя или группу, а изменять, удалять или переименовывать их возможности не было.

9. Сохраните и закройте консоль администрирования MMC.

При выполнении заданий секции используйте следующие инструкции:

- перенесите последовательность выполняемых действий по каждому из пунктов 1-9 в отчет (возможно приведение графических фрагментов, сделанных с экрана, в качестве демонстрационного материала),
- дважды войдите в систему под именами пользователей, соответствующих созданным учетным записям,
- проверьте результат подключения профилей в пунктах 6 и 7 секции: нажмите Пуск | Выполнить, наберите в появившемся окне SendTo и нажмите Enter для ввода, в строке Адрес: появившегося окна отметьте полный путь папки SendTo, удостоверяющий правильную загрузку подключенного профиля.
- обратите внимание на то, каким образом осуществляется влияние подключенных в профилей на работу компьютера при смене пользователя,
- сделайте вывод о проделанной работе и запишите его в отчет.

4. Основные возможности оснастки «Редактор объекта групповой политики»

Административное средство Active Directory предназначено для решения повседневных задач сетевого управления, в число которых входят: создание, удаление, изменение, перемещение и предоставление разрешений на объекты каталога. Объектами управления могут являться подразделения, пользователи, контакты, группы, компьютеры, принтеры, а также объекты общих файлов.

Active Directory обеспечивает системного администратора иерархическим представлением сети и едиными возможностями управления всеми сетевыми объектами ОС Windows 2000 Server, где она является основой групповой политики. Для реализации указанных возможностей в ОС Windows 2000 Server имеются три оснастки «Пользователи и компьютеры Active Directory», «Домены и доверие Active Directory» и «Сайты и службы Active Directory»,

первые две из которых предназначены для создания объектов групповой политики домена или подразделения, третья — соответственно, для создания объекта групповой политики сайта. Для запуска этих оснасток из ОС Windows XP необходимо осуществить подключение к удаленному столу сервера с установленными средствами администрирования ОС Windows 2000 Server, поскольку служба каталогов Active Directory в ОС Windows XP является по умолчанию не доступной.

Понятие групповая политика в области администрирования применяется не только к пользователям и компьютерам, но и к серверам, контроллерам доменов и другим компьютерам под управлением ОС Windows XP. По умолчанию групповая политика применяется к домену и влияет на все его компьютеры и пользователи. Обеспечение групповой политики подразумевает конфигурирование системы и задания набора условий и параметров групповой политики, приводящих к определенным ограничениям или разрешениям в ОС.

Параметры групповой политики определяют различные компоненты конфигурации пользователя, в частности, окружения пользовательского рабочего стола (программы, доступные пользователям, программы, отображающиеся на рабочем столе, и параметры меню Пуск) и конфигурации компьютера, включая параметры, применяемые вне зависимости от того, кто работает на этих компьютерах. Чтобы создать частную конфигурацию компьютера для определенной группы пользователей используется оснастка «Редактор объекта групповой политики».

Набор указанных параметров групповой политики содержится в объекте групповой политики, который, в свою очередь, связан с выбранными объектами Active Directory — сайтами, доменами или подразделениями. Объектами групповой политики являются документы, создаваемые оснасткой. Они хранятся на уровне домена и оказывают влияние на пользователей и компьютеры доменов и подразделений. Кроме того, каждый компьютер с ОС Windows XP имеет единственную, хранящуюся локально, группу параметров, которая называется локальным объектом групповой политики.

Все многообразие параметров групповой политики содержится в соответствующих расширениях оснастки «Групповая политика», посредством которых системному администратору предоставляются широкие возможности по управлению процессами и ресурсами ОС Windows XP:

- на основе реестра, используя расширение Административные шаблоны (А). При этом создается файл, содержащий параметры реестра, записанные в область базы данных реестра пользователя, в разделе HKEY_CURRENT_USER и в разделе HKEY_LOCAL_MACHINE для локального компьютера.
- посредством расширения Назначение сценариев (В). Групповая политика

указывает сценарии входа/выхода пользователей из системы и загрузки/завершения работы.

- используя Редактор перенаправления папок (C). Групповая политика имеет возможность перенаправить системные папки «Мои документы» и «Мои рисунки», из папки «Documents and Settings» локального компьютера в новое место расположения в сети.
- на основе расширения Установка программ (D), которое позволяет назначать, публиковать и восстанавливать приложения.
- посредством расширения Параметры безопасности (E), позволяющей устанавливать ограничения на использование программ, политику открытого ключа, а также осуществлять управление политикой безопасности IP.

Административные шаблоны это текстовые файлы с расширением .adm, содержащие сведения о политике для элементов, расположенных в папке «Административные шаблоны» оснастки. В ОС Windows XP доступно четыре файла административных шаблонов, приведенных в табл. 1.

Файлы административные шаблонов состоят из иерархии категорий и подкатегорий, которые вместе определяют отображение параметров групповой политики. В них содержатся следующие сведения:

- размещение параметров реестра, соответствующих каждому параметру административного шаблона групповой политики,
- величина параметров или ограничений, связанных с каждым параметром административного шаблона,
- значение по умолчанию для большинства параметров,
- объяснение функции каждого параметра,
- версии ОС Windows, поддерживающие каждый параметр.

Таблица 1. Административные шаблоны ОС Windows XP

Шаблон (.adm)	Справка по параметрам	Описание
System	%systemroot%\help\system.chm	В групповой политике шаблон установлен по умолчанию для клиентов ОС Windows 2000 и XP
Inetres	%systemroot%\ help\inetres.chm	В групповой политике шаблон Internet Explorer установлен по умолчанию для клиентов ОС Windows 2000 и XP
Wmplayer	%systemroot%\help\wmplay.chm	Параметры WMP для клиентов ОС Windows 2000 и XP
Conf	%systemroot%\help\conf1.chm	Параметры программы NetMeeting для клиентов ОС Windows 2000 и XP

В среде ОС Windows XP имеется возможность использования сценариев посредством двух серверов Wscript.exe или Cscript.exe, поддерживающих как Visual Basic Scripting Edition

(расширение .vbs), так и JScript (расширение .js) файлы. В частности, в средствах оснастки «Групповая политика» имеется два расширения, расположенные в узлах консоли «Конфигурация компьютера | Конфигурация Windows» или «Конфигурация пользователя | Конфигурация Windows», позволяющие развертывать сценарии с использованием указанных серверов ОС Windows XP. Эти расширения следующие:

- сценарии (запуск/завершение) — расширение, посредством которого можно указать локально выполняемый сценарий при запуске и завершении работы компьютера.
- сценарии (вход/выход из системы) — расширение, посредством которого можно указать выполняемый сценарий при входе и выходе пользователя из системы. Эти сценарии запускаются с правами пользователя, а не администратора.

Перенаправление папки используется для перемещения некоторых специальных папок, например «Мои документы» и «Мои рисунки», в заданное место в сети, для их последующего доступа с разных узлов. В ОС Windows XP возможны следующие специальные папки для перенаправления (табл. 2).

Таблица 2. Специальные папки ОС Windows XP

Специальная папка	Примечания
Application Data	Параметры групповой политики управляют поведением папки «Application Data» при включении кэширования на стороне клиента. Параметры расположены в дереве консоли «Групповая политика» в Административных шаблонах\Сеть\Автономные файлы.
Рабочий стол	Папка может быть перенаправлена независимо от всех остальных специальных папок.
Мои документы	Особенности и преимущества перенаправления этой папки описаны ниже.
Мои документы\Мои рисунки	Эта папка может быть перенаправлена независимо от предыдущей папки «Мои документы» или совместно с ней, как это происходит по умолчанию. Именно эта комбинация является рекомендуемой.
Главное меню	При перенаправлении папки «Главное меню» ее подпапки всегда перенаправляются вместе с ней.

Некоторые из преимуществ, описанных ниже, относятся к перенаправлению любой специальной папки, однако перенаправление папки «Мои документы» может быть особенно удобным, поскольку со временем размер этой папки может увеличиваться.

- При использовании перемещаемого профиля пользователя его частью является только сетевой путь к папке «Мои документы», но не сама папка. Поэтому ее содержимое не нужно копировать и перемещать между клиентом и сервером каждый раз при входе пользователя в систему или его выходе, что делает процессы входа и выхода сравнительно быстрее.
- Даже если пользователь входит в сеть с различных компьютеров, все его документы всегда доступны.
- Технология автономных файлов обеспечивает пользователям доступ к папке «Мои

документы» даже при отсутствии подключения к сети. Это особенно полезно для пользователей, использующих мобильные компьютеры.

- Имеется возможность архивировать данные, хранящиеся на сервере, при управлении перемещаемыми профилями. Это является более безопасным, поскольку не требуется вмешательство пользователя.
- Системный администратор может устанавливать дисковые квоты с помощью групповой политики, ограничивая дисковое пространство, выделенное пользователю для специальных папок.
- Данные пользователя могут быть перенаправлены на жесткий диск локального компьютера с другого жесткого диска, на котором хранятся системные файлы ОС. Это может обезопасить пользовательские файлы, если необходимо будет ее переустанавливать.

Кроме всего прочего, в ОС Windows XP имеется возможность предоставления исключительных прав на специальные папки. Если на вкладке Параметры диалогового окна свойств каждой папки установить флажок Предоставить права монопольного доступа к папке «Мои документы», пользователь и локальная система получают полный контроль над папкой, и никто другой, включая администратора, не будет иметь на нее никаких прав. В противном случае, если этот параметр отключен, то разрешения для папки не изменяются, а используются, применяемые по умолчанию разрешения. Еще одна возможность заключается в том, что на специальные папки могут быть расширены дополнительные разрешения, полный список которых доступен в справке ОС Windows XP. Установка программного обеспечения является неотъемлемой процедурой при работе с любой ОС. Для этого используется одноименная оснастка «Установка программного обеспечения», которая помогает определить способ установки и сопровождения приложений. Также с ее помощью можно управлять приложением внутри объекта групповой политики посредством службы каталогов Active Directory. Приложения управляются в одном из двух режимов: назначения или публикации. Приложение назначается, когда необходимо, чтобы оно было установлено на всех узлах сети. Например, требуется, чтобы на всех компьютерах аудитории было установлено одно и то же приложение. Поскольку объект групповой политики управляет всеми пользователями аудитории, при назначении приложения в объекте групповой политики, оно одновременно объявляется на всех компьютерах, но при этом фактически не устанавливается. Устанавливаются лишь только необходимые данные для создания ярлыка этого приложения в меню Пуск, а в реестре осуществляется связывание расширения его документа с ним самим. При первом выборе приложения на загрузку, а также, если пользователь, не запускавший приложение ранее, выбирает его документ для работы, приложение устанавливается автоматически с

одновременным открытием этого документа.

Назначенное в системе приложение можно удалить, но оно будет объявлено снова при следующем входе. Если выбрать его в меню Пуск, оно будет повторно автоматически установлено.

Приложение публикуется, если необходимо сделать его доступным для тех пользователей, управляемых объектом групповой политики, кто хочет установить это приложение. При этом у пользователей имеется выбор самостоятельно решать, устанавливать приложение или нет. Например, если приложение публикуется для пользователей, желающих его установить, им следует для этого открыть компонент «Установка и удаление программ» на панели управления и произвести установку. В случае если пользователям не удалось установить приложение с помощью этого компонента, но файлы с соответствующим расширением связаны с приложением, оно будет установлено при первой попытке открыть файл с этим расширением. Безопасность компьютера, уязвимость системы безопасности, а также различного вида угрозы заботят не только профессионалов в области информационных технологий, но и рядовых пользователей компьютеров. Многие организации и отдельные пользователи имеют постоянные подключения к Интернету, что подвергает их компьютеры рискам заражения вирусами, несанкционированного проникновения, атак на службы и другим угрозам. Существуют некоторые правила, называемые политиками или параметрами безопасности, которые предназначены для обеспечения защиты ресурсов одного или нескольких компьютеров в сети. Параметры безопасности позволяют контролировать:

- проверку подлинности пользователей при входе в сеть или отдельный узел,
- ресурсы, которые пользователи могут использовать,
- включение и отключение записи действий пользователя или группы в журнале событий,
- принадлежность к группам.

Настраиваются параметры безопасности, используя средства диспетчера настройки безопасности. К этим средствам относятся:

- шаблоны безопасности,
- анализ и настройка безопасности,
- программа командной строки Secedit.exe,
- локальная политика безопасности,
- расширение «Параметры безопасности» для групповой политики.

Расширение «Параметры безопасности» позволяет пользователям изменять настройку безопасности в оснастке «Групповая политика», влияющей, в свою очередь, одновременно на все узлы сети посредством объекта групповой политики. Однако чтобы установить или изменить

отдельные параметры безопасности на отдельных компьютерах, используется средство «Локальная политика безопасности», включающее политику аудита, назначение прав пользователя и локальные параметры безопасности. Чтобы применить несколько параметров безопасности одновременно, имеется возможность определить их с помощью шаблонов безопасности и затем применить к системе с помощью средства «Анализ и настройка безопасности» или программы Secedit.exe (Пуск | Выполнить | Secedit.exe), а также импортировать готовый шаблон в соответствующую локальную или групповую политику.

В заключение следует отметить, что в пакете обновления SP2 для ОС Windows XP с целью повышения безопасности вводятся некоторые изменения параметров безопасности. Обобщая нововведения, параметры безопасности сгруппированы по соответствующим областям (табл. 3).

Таблица 3. Области безопасности ОС Windows XP

Область безопасности	Описание
Политики учетных записей	Политика паролей, политика блокировки учетной записи и политика Kerberos
Локальные политики	Политика аудита, назначение прав пользователя и параметры безопасности
Журнал событий	Параметры журналов событий приложений, системных событий и событий безопасности
Группы с ограниченным доступом	Состав групп с особыми требованиями к безопасности
Системные службы	Параметры запуска и разрешения для системных служб
Реестр	Разрешения для разделов реестра
Файловая система	Разрешения для файлов и папок

Содержание задания

Прежде чем непосредственно перейти к ознакомлению с возможностями изучаемой оснастки, следует обратить внимание на имеющиеся в системе способы ее открытия в изолированном виде. Имеется возможность открыть оснастку через меню Пуск | Выполнить. Для этого необходимо ввести gpedit.msc и нажать Enter для ввода. Кроме того, загрузка оснастки возможна в среде командной оболочки, различные варианты которой отображены ниже (табл. 4). Поэкспериментируйте с имеющимися возможностями загрузки изолированной оснастки «Групповая политика».

Таблица 4. Варианты загрузки групповой политики посредством командной оболочки

№ п.п.	Пример команды	Описание
1	gpedit.msc /gpcomputer:"Имя_компьютера"	редактирование групповой политики локального компьютера.
2	gpedit.msc /gpcomputer:"Имя_компьютера.WingTipToys.com"	редактирование локального объекта групповой политики на локальном компьютере, имя которого задается в формате DNS.
3	gpedit.msc /gpoobject:"LDAP://CN={31B2F340016D-11D2-945F00C04FB984F9},CN=Policies,CN=System,DC=WingTipToys,DC=com" в фигурных скобках {16-байтное число} — глобальный уникальный идентификатор GUID	редактирование объекта групповой политики с применением Active Directory.

1. Загрузите пользовательскую консоль администрирования, с возможностью ее редактирования.
2. Добавьте оснастку «Редактор объекта групповой политики» в корень консоли администрирования, выбрав в поле Объекта групповой политики значение Локальный компьютер.
3. Откройте поочередно все ветви дерева консоли в узле Политика «Локальный компьютер» и изучите где и какие параметры групповой политики располагаются. Обратите внимание на то, что некоторые параметры находятся в состоянии Включено, другие, напротив — Отключено или Не определено (Не задано).
4. В левой части окна консоли выберите «Политика Локальный компьютер | Конфигурация компьютера | Конфигурация Windows | Параметры безопасности | Локальные политики | Назначение прав пользователей / Параметры безопасности».
5. Изучите локальные политики Назначение прав пользователя и Параметры безопасности.
6. Сохраните и закройте консоль администрирования MMC.

При выполнении пункта 6 используйте следующие инструкции:

- последовательно перебирая каждую из локальных политик, изучите ее содержимое, принадлежность к пользователю, а также состояние, в котором она находится,
- запишите полученную информацию в отчет, заполнив табл. 5.
- сделайте вывод о проделанной работе и запишите его в отчет.

Таблица 5. Локальная политика безопасности ОС Windows XP

№ п.п.	Политика		Привилегией обладает	Состояние
	название	описание		
1	Изменение системного времени	Определяет, какие пользователи и группы могут изменять время и дату компьютера.	Администратор Опытный пользователь	Включено

n				

5. Возможности оснастки «Групповая политика» при настройке локального узла

Как утверждалось ранее, с помощью параметров узла Конфигурация компьютера устанавливаются политики, которые применяются к компьютеру независимо от того, кто его использует для входа в сеть, в то время как, с помощью параметра Конфигурация пользователя устанавливаются политики, которые применяются к каждому пользователю, работающему на компьютере.

Таким образом, существует ряд параметров групповой политики предназначенных исключительно для изменений, направленных на создание специальных настроек среды пользователя ОС Windows XP, в частности, для придания системе уникального вида. В качестве примера, с помощью групповой политики можно удалить значки с рабочего стола, изменить содержимое меню Пуск и упростить структуру панели управления.

Содержание задания

1. Загрузите пользовательскую консоль администрирования, созданную ранее, с возможностью ее редактирования.
2. В левой части окна консоли выберите «Политика Локальный компьютер | Конфигурация пользователя | Административные шаблоны | Панель задач и меню «Пуск» / Панель управления / Рабочий стол».
3. Изучите параметры политик | Административные шаблоны | Панель задач и меню «Пуск» / Панель управления / Рабочий стол.
4. Выберите любые пять параметров в каждой из политик Панель задач и меню «Пуск» / Панель управления / Рабочий стол. Измените выбранные параметры на противоположные, перезагрузите компьютер и отметьте полученные визуальные изменения графического интерфейса.
5. Сохраните и закройте консоль администрирования ММС.

При выполнении пункта 3 используйте следующие инструкции:

- самостоятельно последовательно выбирая политики указанных административных шаблонов, в общем количестве не менее десяти, измените их состояние, делая его активным,
- исследуйте влияние смены состояния на внешний вид соответствующего элемента графического интерфейса ОС Windows XP,
- запишите полученную информацию в отчет, заполнив табл. 6.
- сделайте вывод о проделанной работе и запишите его в отчет.
- продемонстрируйте преподавателю конечный результат изменения параметров административных шаблонов, применяемых для настройки уникального вида элементов

графического интерфейса.

Таблица 6. Настройка некоторых административных шаблонов

№ п.п.	Административный шаблон		Результат при активации политики
	Название шаблона	Описание политики	
1	Рабочий стол	Не показывать значок Internet Explorer (IE)	При изменении состояния политики в положение Включено значок IE убирается с рабочего стола и панели быстрого запуска
n			

Контрольные вопросы:

- 1.Перечислить системные процессы Windows.
- 2.Дать определения драйвера устройства в различных контекстах.
- 3.Пояснить принцип функционирования уровня абстрагирования от оборудования.
- 4.Перечислить основные компоненты исполнительной системы Windows.
- 5.Дать определения подсистемам Windows.
- 6.Перечислить возможные места хранения назначенных политик безопасности
- 7.Отобразить переменные среды в Windows двумя способами: из командной оболочки и окна свойств системы
- 8.Задать переменную среды с различными вариантами динамически формируемых значений.

Список литературы, рекомендуемый к использованию по данной теме:

Основная: 1-4

Дополнительная: 1-3

Интернет-ресурсы: 1

Методическая: 1-3

Программное обеспечение: 1-4

Практическая работа №7

Тема: Прерывания, ресурсы и ядро ОС Управление файлами и вводом-выводом в ОС

1. Подготовка к занятию

1. Изучить (повторить) теоретический материал.
2. Ознакомиться с программой работы.
3. Подготовить отчет о работе.
4. Ответить на контрольные вопросы.

2. Распределение времени занятия:

Всего: 90 мин

Вступительная часть 2 мин

Проверка готовности студентов к занятию 5 мин

Основная часть – 70 мин

Проверка выполнения практического занятия 10 мин

Заключительная часть 3 мин

3. Формируемые компетенции: ПК-1, ПК-3

4. Правила работы в лаборатории

К работе в лаборатории допускаются лица, изучившие правила и меры безопасности, сдавшие зачет по ним и усвоившие порядок выполнения лабораторной работы.

4.1. Требования безопасности перед началом работ

- ЗАПРЕЩАЕТСЯ: переодеваться, пользоваться огнем, курить, принимать пищу в лаборатории.

- Убедиться в целостности электрических розеток и разъемов. В лаборатории необходимо быть в сменной обуви.

- Включение компьютера производить только после получения допуска по выполняемой работе и разрешения преподавателя или лаборанта.

4.2. Требования безопасности во время работы

- выполняя практическое занятие, студенты обязаны использовать только вычислительную технику, периферийное оборудование, соединительные кабели, измерительное оборудование и носители информации, непосредственно относящиеся к данному лабораторному занятию;

- подключение и отключение составляющих вычислительного комплекса производить

только при полном снятии напряжения со всех составляющих вычислительного комплекса;

- при обнаружении неисправностей в оборудовании немедленно отключить источники питания и доложить об этом руководителю занятий или лаборанту.

4.3. Требования безопасности по окончании работы

- доложить руководителю занятий или лаборанту о завершении работ;
- привести в порядок и сдать рабочее место лаборанту, и доложить руководству

Введение

ОС Linux - это многопользовательская, многозадачная, многотерминальная операционная система (ОС) из семейства UNIX, под управлением которой могут одновременно выполняться несколько задач. Она предназначена для работы на серверах и рабочих станциях, обеспечивает подключение дополнительных терминалов и допускает этот режим использования графических оболочек.

UNIX-серверы предназначены для хранения и обработки больших объемов информации. Особенно эффективно использование UNIX-серверов при распределенной обработке данных. Для этого разработаны системы распределенных вычислений в соответствии со стандартом CORBA. К таким системам относятся системы управления базами данных (СУБД типа Oracle, Informix), файл-серверы, FTP-серверы, WWW-серверы и др., которые поддерживаются ОС Linux. В распределенных системах информация может находиться на различных рабочих станциях, различных дисках, программные модули могут функционировать на различных компьютерах, но система работает таким образом, что это составляет единое целое. При обработке больших объемов информации используется технология клиент - сервер, при которой пользователь работает только с той информацией, которая ему необходима. Развитием технологии клиент - сервер является технология интеллектуальных агентов.

ОС Linux является сетевой операционной системой для 32-х или 64-х разрядных платформ. Она обеспечивает масштабируемость в диапазоне от игровых приставок (Sony Play Station) до кластерных серверов Internet.

ОС Linux не связана с конкретной моделью компьютеров. Её ядро реализовано на языке высокого уровня (языке СИ), что позволяет достаточно легко переносить эту систему с одной платформы на другую. Система распространяется по лицензии GNU либо подобным свободным лицензиям, обеспечивается как коммерческое, так и свободное сопровождение через Internet. Поставка исходных модулей системы обеспечивает возможность адаптации прикладных программ в случае перехода на другую платформу и дает возможность контроля кодов, реализующих несанкционированный доступ. В разработке системы приняло участие большое количество специалистов, зарегистрировавших свои авторские права, что дает гарантии ее немонополизации.

Подключение персональных компьютеров (ПК) в вычислительную сеть с UNIX - серверами может осуществляться по протоколу TCP/IP, при этом пользователи получают следующие возможности:

- 1) использование UNIX-сервера, как файл - сервера;
- 2) эмуляция на ПК удаленного терминала (режим TELNET);

3) организация системы клиент - сервер (рабочая станция формирует SQL - запросы, сервер их обрабатывает);

4) непосредственный обмен файлами между ПК по протоколу FTP;

5) организация распределенных вычислений по стандарту CORBA.

Все действия в ОС UNIX оформлены как процессы. Процесс представляет собой совокупность выполняемых программ или одну выполняемую программу, которые вызываются при исполнении системной команды. Процесс может породить один или несколько других процессов, которые могут выполняться параллельно. ОС Linux поддерживает многопроцессорную архитектуру для параллельного выполнения процессов.

1. Цель работы

Целью работы является изучение архитектуры и принципов функционирования многопользовательской многозадачной операционной системы Linux, особенности ее использования в качестве сервера и рабочей станции.

2. Теоретическая часть

Система включает следующие основные компоненты.

Ядро. Выполняет функции управления памятью, процессорами. Осуществляет диспетчеризацию выполнения всех программ и обслуживание внешних устройств. Все действия, связанные с вводом/выводом и выполнением системных операций, выполняются с помощью системных вызовов. Системные вызовы реализуют программный интерфейс между программами и ядром. Имеется возможность динамического конфигурирования ядра.

Диспетчер процессов Init. Активизирует процессы, необходимые для нормальной работы системы и производит их начальную инициализацию. Обеспечивает завершение работы системы, организует сеансы работы пользователей, в том числе, для удаленных терминалов.

Интерпретатор команд Shell. Анализирует команды, вводимые с терминала либо из командного файла, и передает их для выполнения в ядро системы. Команды обычно имеют аргументы и параметры, которые обеспечивают модернизацию выполняемых действий. Shell является также языком программирования, на котором можно создавать командные файлы (shell-файлы). При входе в ОС пользователь получает копию интерпретатора shell в качестве родительского процесса. Далее, после ввода команды пользователем создается порожденный процесс, называемый процессом-потомком. Т.е. после запуска ОС каждый новый процесс функционирует только как процесс - потомок уже существующего процесса. В ОС Linux имеется возможность динамического порождения и управления процессами.

Shell - интерпретатор в соответствии с требованиями стандарта POSIX поддерживает графический экраный интерфейс, реализованный средствами языка программирования Tcl/Tk.

Обязательным в системе является интерпретатор Bash, полностью соответствующий стандарту POSIX. В качестве Shell может быть использована оболочка **mc** с интерфейсом, подобным Norton Commander.

Сетевой графический интерфейс X-сервер (X-Windows). Обеспечивает поддержку

графических оболочек.

Графические оболочки KDE, Gnome. Отличительными свойствами KDE являются: минимальные требования к аппаратуре, высокая надежность, интернационализация. Базовые библиотеки KDE (qt, kde-libs) признаны одними из лучших продуктов по созданию графического интерфейса, обеспечивают простое написание программ с использованием передовых технологий. Gnome имеет развитые графические возможности, но более требователен к аппаратным средствам.

Сетевая поддержка NFS, SMB, TCP/IP. NFS - программный комплекс PC-NFS (Network File System) для выполнения сетевых функций. PC-NFS ориентирован для конкретной ОС персонального компьютера (PC) и включает драйверы для работы в сети и дополнительные утилиты. SMB - сетевая файловая система, совместимая с Windows NT. TCP/IP - протокол контроля передачи данных (Transfer Control Protocol/Internet Protocol). Сеть по протоколам TCP/IP является неотъемлемой частью ОС семейства UNIX. Поддерживаются любые сети, от локальных до Internet, с использованием только встроенных сетевых средств.

Инструментальные средства программирования. Основой средств программирования является компилятор GCC или его экспериментальные версии EGCS и PGCC для языков C и C++; модули поддержки других языков программирования (Objective C, Фортран, Паскаль, Modula-3, Ада, Java и др.); интегрированные среды и средства визуального проектирования: Kdevelop, Xwpe; средства адаптации привязки программ AUTOCONFIG, AUTOMAKE.

2.1. Регистрация пользователя в системе

Для входа пользователя с терминала в многопользовательскую операционную систему LINUX необходимо зарегистрироваться в качестве пользователя. Для этого нужно после сообщения

Login:

ввести системное имя пользователя, например, "student". Если имя задано верно, выводится запрос на ввод пароля:

Password:

Наберите пароль "student" и нажмите клавишу *Enter*.

Если имя или пароль указаны неверно, сообщение *login* повторяется. Значение пароля проверяется в системном файле *password*, где приводятся и другие сведения о пользователях. После правильного ответа появляется приветствие LINUX и приглашение:

student@linux:>

Вы получили доступ к ресурсам ОС LINUX.

2.2. Выход из системы

exit - окончание сеанса пользователя.

2.3. Выполнение простых команд

Формат команд в ОС LINUX следующий:

имя команды [аргументы] [параметры] [метасимволы]

Имя команды может содержать любое допустимое имя файла; аргументы - одна или несколько букв со знаком минус (-); параметры - передаваемые значения для обработки; метасимволы интерпретируются как специальные операции. В квадратных скобках указываются необязательные части команд.

Введите команду **echo**, которая выдает на экран свои аргументы:

echo good morning

и нажмите клавишу *Enter*. На экране появится приветствие "*good morning*" – аргумент команды **echo**. Командный интерпретатор *shell* вызвал команду **echo**, реализованную в виде программы на языке СИ, и передал ей аргументы. После этого интерпретатор команд вывел знак-приглашение. Синтаксис команды **echo**:

echo [-n] [arg1] [arg2] [arg3]...

Команда помещает в стандартный вывод свои аргументы, разделенные пробелами и завершаемые символом перевода строки. При наличии флага *-n* символ перевода строки исключается.

who [am i] - получение информации о работающих пользователях.

В квадратных скобках указываются аргументы команды, которые можно опустить. Ответ представляется в виде таблицы, которая содержит следующую информацию:

- идентификатор пользователя;
- идентификатор терминала;
- дата подключения;
- время подключения.

date - вывод на экран текущей даты и текущего времени.

cal [[месяц]год] - календарь; если календарь не помещается на одном экране, то используется команда **cal год | more** и клавишей пробела производится страничный вывод информации.

man <название команды> - вызов электронного справочника об указанной команде. Выход из справочника - нажатие клавиши Q.

Команда **man man** сообщает информацию о том, как пользоваться справочником.

tty - сообщение имени специального файла стандартного вывода, соответствующего терминалу пользователя.

cat <имя файла> - вывод содержимого файла на экран. Команда **cat > text.1** создает

новый файл с именем `text.1`, который можно заполнить символьными строками, вводя их с клавиатуры. Нажатие клавиши *Enter* создает новую строку. Завершение ввода - нажатие *Ctrl - d*. Команда **cat text.1 > text.2** пересылает содержимое файла `text.1` в файл `text.2`. Слияние файлов осуществляется командой **cat text.1 text.2 > text.3**.

ls [-alrstu] [имя] - вывод содержимого каталога на экран. Если аргумент не указан, выдается содержимое текущего каталога.

Аргументы команды:

- a - выводит список всех файлов и каталогов, в том числе и скрытых;
- l - выводит список файлов в расширенном формате, показывая тип каждого элемента, полномочия, владельца, размер и дату последней модификации;
- r - выводит список в порядке, обратном заданному;
- s - выводит размеры каждого файла;
- t - перечисляет файлы и каталоги в соответствии с датой их последней модификации;
- u - перечисляет файлы и каталоги в порядке, обратном их последней модификации.

rm <имя файла> - удаление файла (файлов). Команда **rm text.1 text.2 text.3** удаляет файлы `text.1`, `text.2`, `text.3`. Другие варианты этой команды - **rm text.[123]** или **rm text.[1-3]**.

wc [имя файла] - вывод числа строк, слов и символов в файле.

clear - очистка экрана.

2.4. Группирование команд

Группы команд или сложные команды могут формироваться с помощью специальных символов (метасимволов):

& - процесс выполняется в фоновом режиме, не дожидаясь окончания предыдущих процессов;

? - шаблон, распространяется только на один символ;

***** - шаблон, распространяется на все оставшиеся символы;

| - программный канал - стандартный вывод одного процесса является стандартным вводом другого;

> - переадресация вывода в файл;

< - переадресация ввода из файла;

; - если в списке команд команды отделяются друг от друга точкой с запятой, то они выполняются друг за другом;

&& - эта конструкция между командами означает, что последующая команда выполняется только при нормальном завершении предыдущей команды (код возврата 0);

|| - последующая команда выполняется только, если не выполнилась предыдущая команда (код возврата 1);

() - группирование команд в скобки;

{ } - группирование команд с объединенным выводом;

[] - указание диапазона или явное перечисление (без запятых);

>> - добавление содержимого файла в конец другого файла.

Примеры.

who | wc - подсчет количества работающих пользователей командой **wc** (word count - счет слов);

cat text.1 > text.2 - содержимое файла text.1 пересылается в файл text.2;

mail student < file.txt - электронная почта передает файл file.txt всем пользователям, перечисленным в командной строке;

cat text.1,text.2 - просматриваются файлы text.1 и text.2;

cat text.1 >> text.2 - добавление файла text.1 в конец файла text.2;

cc primer.c & - трансляция СИ - программы в фоновом режиме. Имя выполняемой программы по умолчанию a.out.

cc -o primer.o primer.c - трансляция СИ-программы с образованием файла выполняемой программы с именем primer.o;

rm text.* - удаление всех файлов с именем text;

{cat text.1; cat text.2} | lpr - просмотр файлов text.1 и text.2 и вывод их на печать;

ps [al] [number] - команда для вывода информации о процессах:

-a - вывод информации обо всех активных процессах, запущенных с вашего терминала;

-l - полная информация о процессах;

number - номер процесса.

Команда **ps** без параметров выводит информацию только об активных процессах, запущенных с данного терминала, в том числе и фоновых. На экран выводится подробная информация обо всех активных процессах в следующей форме:

```
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
```

1 S 200 210 7 0 2 20 80 30 703a 03 0:07 cc

1 R 12 419 7 11 5 20 56 20 03 0:12 ps

F - флаг процесса (1 - в оперативной памяти, 2 - системный процесс, 4 - заблокирован в ОЗУ, 20 - находится под управлением другого процесса, 10 - подвергнут свопингу);

S - состояние процесса (O - выполняется процессором, S - задержан, R - готов к выполнению, I - создается);

UID - идентификатор пользователя;

PID - идентификатор процесса;

PPID - номер родительского процесса;

C - степень загрузки процессора;

PRI - приоритет процесса, вычисляется по значению переменной NICE и чем больше число, тем меньше его приоритет;

NI - значение переменной NICE для вычисления динамического приоритета, принимает величины от 0 до 39;

ADDR - адрес процесса в памяти;

SZ - объем ОЗУ, занимаемый процессом;

WCHAN - имя события, до которого процесс задержан, для активного процесса - пробел;

TTY - номер управляющего терминала для процесса;

TIME - время выполнения процесса;

CMD - команда, которая породила процесс.

nice [-приращение приоритета] команда[аргументы] - команда изменения приоритета. Каждое запущенное задание (процесс) имеет номер приоритета в диапазоне от 0 до 39, на основе которого ядро вычисляет фактический приоритет, используемый для планирования процесса. Значение 0 представляет наивысший приоритет, а 39 - самый низший. Увеличение номера приоритета приводит к понижению приоритета, присвоенного процессу. Команда **nice -10 ls -l** увеличивает номер приоритета, присвоенный процессу **ls -l** на 10.

renice 5 1836 - команда устанавливает значение номера приоритета процесса с идентификатором 1836 равным 5. Увеличить приоритет процесса может только администратор системы.

kill [-sig] <идентификатор процесса> - прекращение процесса до его программного завершения. sig - номер сигнала. Sig = -15 означает программное (нормальное) завершение процесса, номер сигнала = -9 - уничтожение процесса. По умолчанию sig = -9. Вывести себя из системы можно командой **kill -9 0**. Пользователь с низким приоритетом может прервать процессы, связанные только с его терминалом.

mc - вызов файлового менеджера (программы - оболочки) *Midnight Commander*, аналогичного *Norton Commander*.

sort [-dr] - сортировка входных файлов и вывод результата на экран.

3. Порядок выполнения работы

1. Ознакомиться с теоретической частью к лабораторной работе.
2. Зарегистрироваться в системе LINUX.
3. Определить день недели, в который Вы родились.
4. Получить подробную информацию обо всех активных процессах.
5. Используя редактор VI (см. приложение), создать два текстовых файла (с расширением TXT) и командой CAT просмотреть их на экране.
6. Получить информацию о работающих пользователях, подсчитать их количество и запомнить в файле.
7. Объединить текстовые файлы в единый файл и посмотреть его на экране.
8. Посмотреть приоритет своего процесса и уменьшить скорость его выполнения за счет повышения номера приоритета.
9. Используя редактор VI, написать программу на языке СИ и запустить ее на трансляцию в фоновом режиме.
10. Показать преподавателю исходный текст программы на языке СИ, текстовый файл, файл с сохранением количества пользователей.
11. Продемонстрировать выполнение СИ - программы.
12. Удалить свои файлы и выйти из системы.

4. Контрольные вопросы

1. Перечислите основные функции и назначение многопользовательской многозадачной операционной системы LINUX и ее отличительные особенности от однопрограммной системы DOS.
2. Какое назначение имеет ядро системы и интерпретатор команд?
3. В чем заключается понятие "процесс" и какие операции можно выполнить над процессами?
4. Как задаются и выполняются простые и сложные команды?
5. Какие функции выполняет командный интерпретатор *Shell*?

5.Список литературы, рекомендуемый к использованию по данной теме:

Основная: 1-4

Дополнительная: 1-3

Интернет-ресурсы: 1

*Методическая:*1-3

Программное обеспечение: 1-4

Практическая работа №8

Тема: Управление памятью в ОС Управление процессами в ОС

1. Подготовка к занятию

1. Изучить (повторить) теоретический материал.
2. Ознакомиться с программой работы.
3. Подготовить отчет о работе.
4. Ответить на контрольные вопросы.

2. Распределение времени занятия:

Всего: 90 мин

Вступительная часть 2 мин

Проверка готовности студентов к занятию 5 мин

Основная часть – 70 мин

Проверка выполнения практического занятия 10 мин

Заключительная часть 3 мин

3. Формируемые компетенции: ПК-1, ПК-3

4. Правила работы в лаборатории

К работе в лаборатории допускаются лица, изучившие правила и меры безопасности, сдавшие зачет по ним и усвоившие порядок выполнения лабораторной работы.

4.1. Требования безопасности перед началом работ

- ЗАПРЕЩАЕТСЯ: переодеваться, пользоваться огнем, курить, принимать пищу в лаборатории.

- Убедиться в целостности электрических розеток и разъемов. В лаборатории необходимо быть в сменной обуви.

- Включение компьютера производить только после получения допуска по выполняемой работе и разрешения преподавателя или лаборанта.

4.2. Требования безопасности во время работы

- выполняя практическое занятие, студенты обязаны использовать только вычислительную технику, периферийное оборудование, соединительные кабели, измерительное оборудование и носители информации, непосредственно относящиеся к данному лабораторному занятию;

- подключение и отключение составляющих вычислительного комплекса производить

только при полном снятии напряжения со всех составляющих вычислительного комплекса;

- при обнаружении неисправностей в оборудовании немедленно отключить источники питания и доложить об этом руководителю занятий или лаборанту.

4.3. Требования безопасности по окончании работы

- доложить руководителю занятий или лаборанту о завершении работ;
- привести в порядок и сдать рабочее место лаборанту, и доложить руководству

- **Файловая структура системы LINUX**

- В операционной системе LINUX файлами считаются обычные файлы, каталоги, а также специальные файлы, соответствующие периферийным устройствам (каждое устройство представляется в виде файла). Доступ ко всем файлам однотипный, в том числе, и к файлам периферийных устройств. Такой подход обеспечивает независимость программы пользователя от особенностей ввода/вывода на конкретное внешнее устройство.

- Файловая структура LINUX имеет иерархическую древовидную структуру. В корневом каталоге размещаются другие каталоги и файлы, включая 5 основных каталогов:

- bin - большинство выполняемых командных программ и *shell* - процедур;
- tmp - временные файлы;
- usr - каталоги пользователей (условное обозначение);
- etc - преимущественно административные утилиты и файлы;
- dev - специальные файлы, представляющие периферийные устройства; при

добавлении периферийного устройства в каталог /dev должен быть добавлен соответствующий файл (черта / означает принадлежность корневому каталогу).

- Текущий каталог - это каталог, в котором в данный момент находится пользователь. При наличии прав доступа, пользователь может перейти после входа в систему в другой каталог. Текущий каталог обозначается точкой (.); родительский каталог, которому принадлежит текущий, обозначается двумя точками (..).

- Полное имя файла может включать имена каталогов, включая корневой, разделенных косой чертой, например: /home/student/file.txt. Первая косая черта обозначает корневой каталог, и поиск файла будет начинаться с него, а затем в каталоге home, затем в каталоге student.

- Один файл можно сделать принадлежащим нескольким каталогам. Для этого используется команда **ln (link)**:

- **ln <имя файла 1> <имя файла 2>**

- Имя 1-го файла - это полное составное имя файла, с которым устанавливается связь; имя 2-го файла - это полное имя файла в новом каталоге, где будет использоваться эта связь. Новое имя может не отличаться от старого. Каждый файл может иметь несколько связей, т.е. он может использоваться в разных каталогах под разными именами. Команда **ln** с аргументом -s создает символическую связь:

- **ln -s <имя файла 1> <имя файла 2>**

- Здесь имя 2-го файла является именем символической связи. Символическая связь является особым видом файла, в котором хранится имя файла, на который символическая связь ссылается. LINUX работает с символической связью не так, как с обычным файлом - например, при выводе на экран содержимого символической связи появятся данные файла, на который эта символическая связь ссылается.

- В LINUX различаются 3 уровня доступа к файлам и каталогам:

- 1) доступ владельца файла;
- 2) доступ группы пользователей, к которой принадлежит владелец файла;
- 3) остальные пользователи.
- Для каждого уровня существуют свои байты атрибутов, значение которых

расшифровывается следующим образом:

- r – разрешение на чтение;
- w – разрешение на запись;
- x – разрешение на выполнение;
- – – отсутствие разрешения.
- Первый символ байта атрибутов определяет тип файла и может интерпретироваться со следующими значениями:
 - – – обычный файл;
 - d – каталог;
 - l – символическая связь;
 - b – блок-ориентированный специальный файл, который соответствует таким периферийным устройствам, как накопители на магнитных дисках;
 - c – байт-ориентированный специальный файл, который может соответствовать таким периферийным устройствам как принтер, терминал.
- В домашнем каталоге пользователь имеет полный доступ к файлам (READ, WRITE, EXECUTE; r, w, x).
- Атрибуты файла можно просмотреть командой **ls -l** и они представляются в следующем формате:

-
- d rwx rwx rwx
- ||||
- |||| Доступ для остальных пользователей
- ||| Доступ к файлу для членов группы
- || Доступ к файлу владельца
- | Тип файла (директория)
-
- Пример. Командой **ls -l** получим листинг содержимого текущей директории student:
-
- - rwx --- --- 2 student 100 Mar 10 10:30 file_1
- - rwx --- r-- 1 adm 200 May 20 11:15 file_2
- - rwx --- r-- 1 student 100 May 20 12:50 file_3
-
- После байтов атрибутов на экран выводится следующая информация о файле:
- - число связей файла;
- - имя владельца файла;
- - размер файла в байтах;
- - дата создания файла (или модификации);
- - время;
- - имя файла.
- Атрибуты файла и доступ к нему, можно изменить командой:
- **chmod <коды защиты> <имя файла>**
- Коды защиты могут быть заданы в числовом или символьном виде. Для символьного кода используются:
- знак плюс (+) - добавить права доступа;
- знак минус (-) - отменить права доступа;
- r,w,x - доступ на чтение, запись, выполнение;
- u,g,o - владельца, группы, остальных.
- Коды защиты в числовом виде могут быть заданы в восьмеричной форме. Для контроля установленного доступа к своему файлу после каждого изменения кода защиты нужно проверять свои действия с помощью команды **ls -l**.
-
- Примеры:

- **chmod g+rw,o+r file.1** - установка атрибутов чтения и записи для группы и чтения для всех остальных пользователей;
- **ls -l file.1** - чтение атрибутов файла;
- **chmod o-w file.1** - отмена атрибута записи у остальных пользователей;
- **>letter** - создание файла letter. Символ > используется как для переадресации, так и для создания файла;
- **cat** - вывод содержимого файла;
- **cat file.1 file.2 > file.12** - конкатенация файлов (объединение);
- **mv file.1 file.2** - переименование файла file.1 в file.2;
- **mv file.1 file.2 file.3 directory** - перемещение файлов file.1, file.2, file.3 в указанную директорию;
- **rm file.1 file.2 file.3** - удаление файлов file.1, file.2, file.3;
- **cp file.1 file.2** - копирование файла с переименованием;
- **mkdir namedir** - создание каталога;
- **rm dir_1 dir_2** - удаление каталогов dir_1 dir_2;
- **ls [acdfgilqrstv CFR] namedir** - вывод содержимого каталога; если в качестве namedir указано имя файла, то выдается вся информация об этом файле. Значения аргументов:
 - l — список включает всю информацию о файлах;
 - t — сортировка по времени модификации файлов;
 - a — в список включаются все файлы, в том числе и те, которые начинаются с точки;
 - s — размеры файлов указываются в блоках;
 - d — вывести имя самого каталога, но не содержимое;
 - r — сортировка строк вывода;
 - i — указать идентификационный номер каждого файла;
 - v — сортировка файлов по времени последнего доступа;
 - q — непечатаемые символы заменить на знак ?;
 - c — использовать время создания файла при сортировке;
 - g — то же что -l, но с указанием имени группы пользователей;
 - f — вывод содержимого всех указанных каталогов, отменяет флаги -l, -t, -s, -r и активизирует флаг -a;
 - C — вывод элементов каталога в несколько столбцов;
 - F — добавление к имени каталога символа / и символа * к имени файла, для которых разрешено выполнение;
 - R — рекурсивный вывод содержимого подкаталогов заданного каталога.
- **cd <namedir>** - переход в другой каталог. Если параметры не указаны, то происходит переход в домашний каталог пользователя.
- **pwd** - вывод имени текущего каталога;
- **grep [-vcilns] [шаблон поиска] <имя файла>** - поиск файлов с указанием или без указания контекста (шаблона поиска).
 - Значение ключей:
 - v — выводятся строки, не содержащие шаблон поиска;
 - c — выводится только число строк, содержащих или не содержащих шаблон;
 - i — при поиске не различаются прописные и строчные буквы;
 - l — выводятся только имена файлов, содержащие указанный шаблон;
 - n — перенумеровать выводимые строки;
 - s — формируется только код завершения.
- Примеры.
 - 1. Напечатать имена всех файлов текущего каталога, содержащих последовательность "student" и имеющих расширение .txt:
 - **grep -l student *.txt**

- 2. Определить имя пользователя, входящего в ОС LINUX с терминала tty23:
- **who | grep tty23**
-
- **3. Порядок выполнения работы**
- 1. Ознакомиться с файловой структурой ОС LINUX. Изучить команды работы с файлами.
- 2. Используя команды ОС LINUX, создать два текстовых файла.
- 3. Полученные файлы объединить в один файл и его содержимое просмотреть на экране.
- 4. Создать новую директорию и переместить в нее полученные файлы.
- 5. Вывести полную информацию обо всех файлах и проанализировать уровни доступа.
- 6. Добавить для всех трех файлов право выполнения членам группы и остальным пользователям.
- 7. Просмотреть атрибуты файлов.
- 8. Создать еще один каталог.
- 9. Установить дополнительную связь объединенного файла с новым каталогом, но под другим именем.
- 10. Создать символическую связь.
- 11. Сделать текущим новый каталог и вывести на экран расширенный список информации о его файлах.
- 12. Произвести поиск заданной последовательности символов в файлах текущей директории и получить перечень соответствующих файлов.
- 13. Получить информацию об активных процессах и имена других пользователей.
- 14. Сдать отчет о работе и удалить свои файлы и каталоги.
- 15. Выйти из системы.
-
- **4. Контрольные вопросы**
- 1. Что считается файлами в ОС LINUX?
- 2. Объясните назначение связей с файлами и способы их создания.
- 3. Что определяет атрибуты файлов и каким образом их можно просмотреть и изменить?
- 4. Какие методы создания и удаления файлов, каталогов Вы знаете?
- 5. В чем заключается поиск по шаблону?
- 6. Какой командой можно получить список работающих пользователей и сохранить его в файле?

5.Список литературы, рекомендуемый к использованию по данной теме:

Основная: 1-4

Дополнительная: 1-3

Интернет-ресурсы: 1

Методическая: 1-3

Программное обеспечение: 1-4

Практическая работа №9

Тема: Концептуальные основы ОС Принципы построения ОС

1. Подготовка к занятию

1. Изучить (повторить) теоретический материал.
2. Ознакомиться с программой работы.
3. Подготовить отчет о работе.
4. Ответить на контрольные вопросы.

2. Распределение времени занятия:

Всего: 90 мин

Вступительная часть 2 мин

Проверка готовности студентов к занятию 5 мин

Основная часть – 70 мин

Проверка выполнения практического занятия 10 мин

Заключительная часть 3 мин

3. Формируемые компетенции: ПК-1, ПК-3

4. Правила работы в лаборатории

К работе в лаборатории допускаются лица, изучившие правила и меры безопасности, сдавшие зачет по ним и усвоившие порядок выполнения лабораторной работы.

4.1. Требования безопасности перед началом работ

- ЗАПРЕЩАЕТСЯ: переодеваться, пользоваться огнем, курить, принимать пищу в лаборатории.
- Убедиться в целостности электрических розеток и разъемов. В лаборатории необходимо быть в сменной обуви.
- Включение компьютера производить только после получения допуска по выполняемой работе и разрешения преподавателя или лаборанта.

4.2. Требования безопасности во время работы

- выполняя практическое занятие, студенты обязаны использовать только вычислительную технику, периферийное оборудование, соединительные кабели, измерительное оборудование и носители информации, непосредственно относящиеся к данному лабораторному занятию;
- подключение и отключение составляющих вычислительного комплекса производить только при полном снятии напряжения со всех составляющих вычислительного комплекса;
- при обнаружении неисправностей в оборудовании немедленно отключить источники питания и доложить об этом руководителю занятий или лаборанту.

4.3. Требования безопасности по окончании работы

- доложить руководителю занятий или лаборанту о завершении работ;
- привести в порядок и сдать рабочее место лаборанту, и доложить руководству

- **Общие теоретические сведения**

- *K Desktop Environment (Среда рабочего стола K)* KDE предназначена для поддержания тех же функциональных возможностей графического интерфейса, какие предоставляют и другие популярные системы, например MacOS и Windows. Кроме выполнения стандартных функций, KDE обладает рядом специфических характеристик, которые расширяют возможности графической среды. Для Linux разработано несколько диспетчеров окон, таких, как *olwm*, *fvwm*, *afterstep* и другие. Однако, их возможности не идут ни в какое сравнение с возможностями KDE.

- **Оконная среда KDE**

- Как и большинство оконных менеджеров, KDE представляет собой интегрированную среду, содержащую базовые средства для решения ряда повседневных задач. Например, с помощью **KDE** можно выполнять ряд операций:

- • Размещение на рабочем столе ярлыков гибких дисков для их монтирования, размонтирования и работы с ними.
- • Отображение в графическом виде файловой структуры и перемещение по ней.
- • Сопоставление приложений с файлами определенных типов. При этом если щелкнуть на выбранном файле, автоматически будет загружаться нужное приложение.
- • Создание на рабочем столе ярлыков принтеров. Если мышью перетащить к такому ярлыку файл, он будет распечатан.

- В состав KDE входит не только рабочий стол, но и целый набор приложений и утилит для работы с ним. В стандартном дистрибутиве KDE имеется более сотни программ — от игр и системных утилит до целых блоков офисных программ. Кроме того, приложения KDE могут взаимодействовать друг с другом для упрощения выполнения всевозможных операций.

- **Компоненты рабочего стола KDE.**

- Рабочий стол KDE разделен на три основные части - "поверхность" рабочего стола, панель и линейку задач. Основная рабочая область среды KDE называется рабочим столом. Это тот фон, на котором отображаются все другие компоненты. На рабочем столе можно размещать ярлыки программ, документов и устройств, к которым чаще всего приходится обращаться. Это позволяет легко получать доступ к соответствующим объектам для работы с ними. Кроме той области, что отображается на экране, KDE предоставляет дополнительное виртуальное рабочее пространство для выполнения программ. По умолчанию поддерживается четыре виртуальных рабочих стола. Виртуальный рабочий стол — это, по сути, другой экран, на который можно переключиться для того, чтобы запустить приложение или выполнить еще какую-то работу. Программы и окна легко перемещаются между различными виртуальными рабочими столами. Дополнительные возможности, предоставляемые за счет использования виртуальных рабочих столов, могут быть использованы самыми разными программами. При этом нет необходимости сворачивать и разворачивать окна выполняемых приложений. Можно просто отложить выполняемое приложение в таком виде, как есть, а затем вернуться к нему по завершении выполнения.

- **2.2.1 Панель**

- Панель располагается в нижней части экрана. На панели размещаются кнопки, позволяющие выполнять основные процедуры KDE, а также ярлыки наиболее часто используемых программ. Одним из особо важных элементов на панели является кнопка Application Starter (Запуск Приложений), которая расположена (по умолчанию) в левой части панели. Это кнопка с литерой "K" над изображением зубчатого колеса. С ее помощью можно открыть меню, в котором представлены все приложения, установленные на данную систему. Кроме того, это же меню может быть использовано для доступа к некоторым другим разделам KDE, таким, как диалоговая справка и *Панель Управления (Control Panel)*.

- На панели размещен переключатель виртуальных рабочих столов *Пейджер*,

Панель Задач (Taskbar) и Часы (Clock). Панель задач отображает открытые на текущем рабочем столе окна. Чтобы получить немедленный доступ к программе, нужно просто щелкнуть в соответствующем месте на панели задач.

- Запустить на выполнение программу можно одним из перечисленных ниже способов.
- **• Щелкнуть кнопкой на панели.** Некоторые программы представлены по умолчанию на панели в виде ярлыков или кнопок, например эмулятор виртуальных рабочих столов, панель управления, вызов справки и текстовый редактор.
- **• Щелкнуть на элементе рабочего стола.** По умолчанию на рабочем столе размещается только два объекта. Это *Корзина* и ярлык рабочего каталога. Пользователи сами размещают на рабочем столе наиболее нужные и часто используемые программы.
- **• Выбрать программу из меню запуска приложений.** Достаточно щелкнуть на литере "К" и выбрать тот пункт меню, который соответствует запускаемому приложению.
- **• Использовать диспетчер файлов.** В окне диспетчера файлов нужно выбрать соответствующий файл и щелкнуть на нем мышью.
- Можно, конечно, запустить программу на выполнение в командной строке окна терминала – задать название программы. Можно также нажатием клавиш <Alt+F2> вызвать окно запуска программ и ввести туда название программы.
- Ряд полезных программ облегчает работу пользователя.
- В первую очередь, это программа эмуляции терминала *konsole*, позволяющая открывать окна и получать доступ к стандартной командной строке. На панели имеется соответствующая кнопка с изображением маленького монитора и ракушки.
- Справку в диалоговом режиме можно получить, если щелкнуть на кнопке панели с изображением спасательного круга. Справка включает в себя разные темы, как, например, программа-гид для начинающих пользователей и система контекстного поиска для используемых в KDE приложений.
- Просмотреть файловую систему или получить доступ к ресурсам World Wide Web можно, используя окно диспетчера файлов. Для того чтобы диспетчер файлов отобразил в своем окне содержимое рабочего каталога, нужно щелкнуть на папке панели с изображением домика.
- Щелканье по кнопке ► удаляет панель с экрана. Эта кнопка остается при этом на экране, так что можно вернуть панель обратно. Это свойство действует только на открытый в данный момент рабочий стол; другие рабочие столы сохраняют вид мини - или главной панели.
- *Список задач* - кнопка, расположенная справа от меню приложений (обозначена пиктограммой монитора), несет меню, содержащее все активные на данный момент окна, отсортированные по имени. Это позволяет легко и быстро найти необходимое окно и уменьшает захламленность экрана при работе с несколькими окнами.

- 2.3.2 Настройка KDE

- *Центр управления (Control Center)* (кнопка с изображением гаечного ключа) составляет основу всей системы настроек KDE. В ее входит множество панелей для всевозможных компонентов рабочей среды и даже некоторых приложений KDE.
- В центре управления используется деление на группы, щелкнув на знаке "плюс" в углу группы, можно увидеть список входящих в группу компонентов. Щелкнув на знаке "минус" в том же углу группы, этот список можно свернуть. Доступ к любому диалогу с раскрытым деревом меню можно получить при помощи Preferences (Предпочтения) и из меню запуска программ Start Application.
- Большинство диалоговых окон имеют кнопку вызова справки. В самом простом случае это контекстная справка. Для ее получения нужно щелкнуть мышью на знаке вопроса на рамке окна. Курсор мыши при этом изменит свой вид на стрелку с большим знаком вопроса. Если теперь щелкнуть на том элементе диалогового окна, с которым возникли трудности, появится прямоугольник желтого цвета с текстом справки. Для получения более детальной

справки можно воспользоваться опцией Help (Справка) на левой панели. Наконец, если возникли проблемы с поиском необходимого диалогового окна, на этой же панели нужно выбрать опцию Search (Поиск). Затем нужно ввести ключевое слово, по которому и будет осуществляться поиск.

- **Control Center**

- KDE предоставляет широкие возможности по модифицированию внешнего вида окон и рабочей области, включая отображение фона, ярлыков, шрифтов и тому подобное. Не представляет труда и управление работой отдельных компонентов, вроде того же рабочего стола или окна. Например, можно управлять реакцией элемента на щелчок мышью, процессом загрузки и отображения выбранных окон, выбирать хранитель экрана. Все эти и многие другие возможности может предоставить рассматриваемая группа Control Center.

- Для настройки параметров работы рабочего стола и окон следует выбрать опцию нужного диалога настройки под названием Desktop на дереве центра управления.

- Изменение схемы цветов

- Диалоговое окно выбора цвета Appearance&Themes ⇒ Colors (Цвета) предназначено для изменения используемой цветовой схемы для окон KDE и других графических приложений.

- Цветовая схема включает в себя 18 пунктов выбора цвета для различных элементов окна программы и установки контрастов. В области предварительного просмотра отображаются все элементы окна, реагирующие на изменение цветовой схемы. Как только пользователь меняет параметры или установки, в области просмотра отражаются внесенные изменения. Можно выбрать уже готовую цветовую схему из списка Color Scheme (Схема Цветов).

- Для изменения какой-то конкретной установки нужно выбрать соответствующий элемент из выпадающего меню области цветов Widget Color (Декорация) или щелкнуть в нужной части окна предварительного просмотра. После того как элемент выбран, можно изменить его цвет. Для этого достаточно щелкнуть на кнопке и выбрать понравившийся цвет из появившегося диалогового окна выбора цвета.

- Контраст изменяется при помощи позиционирования специального рычажка контраста, который может размещаться в диапазоне от Low (Низкий) до High (Высокий). Эти установки применяются при отображении трехмерных рамок вокруг элементов интерфейса приложений KDE.

- Для подтверждения выбора следует щелкнуть на кнопке Apply (Применить). Если приходится часто менять цветовые установки, бывает полезно внести изменения в список цветовых схем. Для этого нужно щелкнуть на кнопке Save Scheme и задать название для своей схемы. Для удаления схемы из списка нужно выделить ее и щелкнуть на кнопке Remove (Удалить).

- Изменение фона

- Для изменения цвета фона или фоновой узоры рабочего стола нужно на дереве опций центра управления последовательно выбрать Control Center⇒Appearance&Themes⇒Background. В результате появится диалоговое окно, имеющее три основные области:

- • список виртуальных рабочих столов;
- • окно предварительного просмотра;
- • окно настройки параметров.

- Каждый виртуальный стол в KDE имеет собственные настройки фона. Для каждого такого стола можно выбрать фон с одноцветной или двухцветной палитрой, а также фоновый узор. Если используется фоновый узор, можно задать способ его отображения. Можно также выбрать несколько узоров и автоматически переключаться между ними. Доступны и более усовершенствованные опции, позволяющие сочетать цвета и узоры, а также поддерживать динамические настройки фона.

- В процессе внесения изменений в установки они отображаются в окне

предварительного просмотра.

- Виртуальные рабочие столы
- Производить настройку параметров виртуальных рабочих столов в KDE можно в диалоговом окне Control Center ⇒ Desktop ⇒ Multiple Desktops.
- Указатель Number of Desktops (Количество рабочих столов) показывает, сколько виртуальных рабочих столов доступно. Их число может изменяться в диапазоне от одного до шестнадцати. Здесь же можно задать название для рабочего стола, которое потом будет отображено в списке окон (Window List) или использовано в настройках панели.
- Хранитель экрана
- Диалоговое окно выбора хранителя

экрана Appearance&Themes ⇒ Screensaver позволяет выбрать хранитель экрана и осуществить настройку его параметров. Опции настройки бывают глобальные, как, например, опция установки времени запуска хранителя экрана, и индивидуальные — для каждого отдельного хранителя. Диалоговое окно выбора хранителя экрана имеет три основные секции:

- • окно предварительного просмотра;
- • список программ — хранителей экрана;
- • опции настройки.
- Необходимо выбрать название нужной программы из предложенного списка. Для настройки параметров необходимо щелкнуть на кнопке Setup (Настройка) и в появившемся диалоговом окне произвести установку нужных характеристик.
- Для установки интервала времени, через который будет запускаться хранитель экрана, нужно ввести в поле опции Settings (Установки) величину данного интервала в минутах.
- Параметр Priority (Приоритет) позволяет определить распределение процессором времени на работу хранителя экрана. Это пример того, как в Linux организована многозадачность. Если нужно, чтобы у хранителя был наивысший приоритет (например, для качественного вывода анимации), следует передвинуть рычажок в позицию High (Высокий). Если же, наоборот, необходимо обеспечить высокий приоритет других процессов, нужна позиция для рычажка приоритетности хранителя — Low (Низкий).
- Для того чтобы проверить выполненные установки, следует щелкнуть на кнопке Test (Просмотр). Для подтверждения сделанного выбора нажмите кнопку OK или Apply.
- Настройка диспетчера окон
- С помощью

опций Control Center Appearance&Themes ⇒ Desktop/Window behavior (Поведение Окон) центра управления можно устанавливать поведение диспетчера окон. Эти настройки определяют способ отображения окон в случае их перемещения и изменения размера, а также управляют процессом разворачивания, размещения и выделения окон при работе с диспетчером окон. Опции в верхней части диалогового окна позволяют выполнить настройку параметров, задающих режим перемещения окна и изменения его размеров, а также определяют функциональность команды Maximize (Развернуть). Можно задать такой режим отображения окна при перемещении или изменении его размера, что окно будет отображаться вместе со всем своим содержимым или же в виде прозрачной рамки. Если выбран режим отображения всего содержимого окна, процесс перемещения или изменения размеров окна будет требовать дополнительного времени для обновления отображаемых на экране элементов.

- Если используются окна с изменяемыми размерами, можно выбрать режим обновления содержимого окна при каждом изменении его размера. Для этого следует воспользоваться установками Resize (Изменение размера). Для выбора частоты обновления можно воспользоваться специальным рычажком. Если сделан выбор, отличный от None (Никакой), то каждый раз, при изменении размеров окна, его содержимое будет обновляться. Это дает возможность отслеживать процесс заполнения окна программой и позволяет выбрать оптимальные размеры последнего.

- Window behavior/Moving – меню установок размещения окна на экране Placement (Расположение) позволяет определить место на экране, где будет

отображаться окно. Поддерживаются такие методы.

- • **Smart (Умный)** — минимизируется перекрытие между окнами.
- • **Cascade (Каскад)** — первое окно отображается в левом верхнем углу.

Следующее окно отображается сдвинутым немного вправо и вниз, так что окна практически полностью перекрываются. И так далее. Окна расположены, как карты в руке при игре в преферанс.

- • **Random (Произвольный)** — окна располагаются на экране в произвольном порядке.

- *Метод получения фокуса* (т.е. метод выделения отдельных окон или элементов) является, пожалуй, индивидуальным методом настроек KDE. С помощью этого метода определяется, какое из открытых окон активно и какие следует выполнить действия при активизации окна. Более детально это выглядит так.

- • **Click to focus (Передача фокуса щелчком)**. Окно получает фокус (т.е. становится активным) при щелчке на нем мышью. При этом окно автоматически выводится на первый план по отношению к другим окнам. Такой метод используется по умолчанию.

- • **Focus follows mouse (Фокус за мышью)**. Окно получает фокус при непосредственном обращении к нему (это можно сделать с помощью указателя мыши, используя комбинацию клавиш <Alt+Tab> и тому подобное). При этом окно может подниматься поверх других окон, а может и не подниматься. Перемещение указателя мыши на рабочую область за пределы окна не означает потерю последним фокуса. При выборе опции Auto Raise (Всплывать автоматически) окно будет всплывать на экране при перемещении в его область курсора в течение нескольких миллисекунд. Число этих миллисекунд устанавливается с помощью рычажка Delay (Задержка). Если выбрана опция Click Raise (Всплывать при щелчке), окно будет подниматься поверх других окон при щелчке в любой части окна. В противном случае такая реакция окна будет наблюдаться только при щелчке на его заголовке. Это исключительно полезный метод передачи фокуса, поскольку позволяет набирать текст в одном окне и одновременно читать содержимое другого окна, расположенного частично поверх указанного.

- • **Focus Under Mouse (Фокус под мышью)**. Окно получает фокус при любом перемещении на него указателя мыши. При этом комбинация клавиш <Alt+Tab> может и не помочь.

- • **Focus Strictly Under Mouse (Фокус только под мышью)**. Окно получает фокус, только если указатель мыши находится внутри окна. Если указатель мыши находится в рабочей области, где нет окон, ни одно из окон не получит фокус.

- **Использование окон**

- Открытое окно состоит из следующих элементов.

- *Window menu (Меню управления окном)* - В левом верхнем углу каждого окна находится пиктограмма манипулирования окном. При щелчке на ней появляется меню, содержащее команды с помощью которых можно манипулировать данным окном. Maximize (Максимизировать) увеличит окно до максимально возможного размера. Minimize (Минимизировать) сделает ваше окно невидимым. Move(Переместить) позволяет передвигать окно с помощью мыши. Size (Изменить размер) позволит вам увеличить или уменьшить окно. Shade – свернет окно до заголовка. To desktop...(На рабочий стол) позволит перевести окно на другой рабочий стол. Выберите рабочий стол, на который вы хотите переместить это окно. Окно при этом исчезнет. Для того чтобы увидеть его снова, выберите имя на Линейке задач, или щелкните на соответствующую кнопку рабочего стола на панели KDE. Close (Заккрыть) закроет данное окно. Always on Top – оставляет окно поверх всех открытых окон.

- Использование панели меню каждого окна в KDE очень просто. Щелкните на команду, и она будет исполнена. При нажатии на правую кнопку мыши появится контекстное меню, позволяющее вывести на экран панель меню. Можете отсоединить меню от окна и оставить его "плавать" по экрану.

- Ниже панели меню находятся пиктограммы инструментов, которые позволяют исполнять различные команды. Можно передвинуть инструментальную панель - влево, вправо, вверх, вниз, и, конечно, она тоже может "плавать".

-
- **3. Порядок выполнения работы**
- 1. Запустите Центр управлений.
- 2. Поменяйте Фон, сначала на одноцветный, а затем вставьте фоновое изображение.
- 3. Установите хранитель экрана, на своё усмотрение, и режим ожидания равный минуте.
- 4. Сделайте так, чтобы окна передвигались вместе со всем их содержимым.
- 5. Задайте звуковой щелчок, подтверждающий нажатие каждой клавиши.
- 6. Измените ширину линейки панели.
- 7. Запустите диспетчер приложений. И запустите программу текстового процессора KWord.
- 8. В другом рабочем столе откройте программу растрового редактора Paint.
- 9. Откройте KWord и наберите следующий текст:
- The Quick Brown Fox Jumps Over The Lazy Dog, используя два разных стиля по вашему выбору. Сохраните этот файл в домашнем каталоге пользователя, закройте KWord.
- 10. Откройте ваш домашний каталог пользователя Konqueror'ом, создайте в нем каталог, скопируйте ваш текстовый файл в этот каталог.
- 11. Ознакомьтесь с содержанием домашнего каталога, скопируйте с дискеты файлы.
- 12. Получите справку об интересующем вас объекте.
- 13. Создайте любой рисунок с помощью Paint, чтобы в нем были ВСЕ фигуры (1. эллипс, 2. окружность, 3. линия, 4. прямоугольник, 5. круг) хотя бы по одному разу и присутствовало не менее четырех цветов.
- 14. Сохраните файл с рисунком в домашнем каталоге, закройте Paint.
- 15. Скопируйте файл с рисунком в тот же созданный вами каталог.
- 16. Измените атрибуты доступа к созданным файлам.
- 17. Покажите преподавателю ваши файлы, затем удалите их.
-
- **4. Контрольные вопросы**
- 1. Перечислите стандартные функции KDE.
- 2. Что является компонентом рабочего стола KDE?
- 3. Назовите функции панели рабочего стола.
- 4. Как получить справку в диалоговом режиме?
- 5. Какие функции предоставляет центр управления KDE?
- **5.Список литературы, рекомендуемый к использованию по данной теме:**
- *Основная:* 1-4
- *Дополнительная:* 1-3
- *Интернет-ресурсы:* 1
- *Методическая:* 1-3
- *Программное обеспечение:* 1-4

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

для обучающихся по организации и проведению самостоятельной работы
по дисциплине «Безопасность операционных систем»

для студентов направления подготовки
10.03.01 «Информационная безопасность»

Направленность (профиль):

«Организация и технологии защиты информации (по отрасли или в сфере профессиональной
деятельности)»

СОДЕРЖАНИЕ

1. Цель и задачи освоения дисциплины	3
2. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы	3
1. Теоретическая часть: Файловые системы.....	6
1.1. Файловые системы Windows	6
2. Практическая часть	16
2.1. Вопросы по разделу	16
Упражнение 1. Преобразование файловой системы FAT16 или FAT32 в NTFS с помощью Windows XP	17
Файловая система операционной системы UNIX.....	20
1.5. Файловые системы операционной системы Linux	24
1.6. Виртуальные файловые системы.....	25
Редактирование реестра	34
Восстановление реестра	35
Восстановление разделов реестра	35
Восстановление всего реестра	35
Организация пакетных файлов и сценариев в ОС Windows.....	51
Задание 1. Команда Set	55
Содержание задания	58
Задание 2. Команды Rem, Echo	59
Содержание задания	60
Задание 3. Утилита For	60
Содержание задания	63
Задание 4. Команда Goto	64
Содержание задания	65
Задание 5. Команда Call	65
Содержание задания	66

ВВЕДЕНИЕ

Целью изучения дисциплины «Безопасность операционных систем» имеет целью обучить студентов принципам построения защиты информации в ОС и анализа надежности защиты ОС, а также приобретение набора общекультурных и общепрофессиональных компетенций будущего бакалавра по направлению подготовки 10.03.01 Информационная безопасность

Задачами дисциплины являются:

- теоретическая и практическая подготовка бакалавров к настройке параметров безопасности различных типов операционных систем;
- объяснение принципов построения подсистем защиты в ОС различной архитектуры, средств и методов несанкционированного доступа к ресурсам ОС;
- привитие студентам системного подхода к проблеме защиты информации в ОС; механизмов защиты информации и возможностей по их преодолению.

1. Общая характеристика самостоятельной работы студента

Самостоятельная работа включает те разделы курса, которые не получили достаточного освещения на лекциях по причине ограниченности лекционного времени и большого объема изучаемого материала.

Методическое обеспечение самостоятельной работы по дисциплине состоит из:

- 1) Определения учебных вопросов, которые студенты должны изучить самостоятельно;
 - 2) Подбора необходимой учебной литературы, обязательной для проработки и изучения;
 - 3) Поиска дополнительной научной литературы, к которой студенты могут обращаться по желанию, если у них возникает интерес в данной теме;
 - 4) Определения контрольных вопросов, позволяющих студентам самостоятельно проверить качество полученных знаний;
 - 5) Организации консультаций преподавателя со студентами для разъяснения вопросов, вызвавших у обучающихся затруднения при самостоятельном освоении учебного материала.
- Самостоятельная работа студента – это особым образом организованная деятельность, включающая в свою структуру такие компоненты, как:

- уяснение цели и поставленной учебной задачи;
- четкое и системное планирование самостоятельной работы;
- поиск необходимой учебной и научной информации;
- освоение собственной информации и ее логическая переработка;
- использование методов исследовательской, научно-исследовательской работы для решения поставленных задач;
- выработка собственной позиции по поводу полученной задачи;
- представление, обоснование и защита полученного решения;
- проведение самоанализа и самоконтроля

К основным видам самостоятельной работы студентов относятся:

- работа с понятийным аппаратом;
- конспектирование первоисточников;
- проектирование фрагментов исследовательской деятельности;
- анализ научных исследований по психологической проблематике;
- подготовка научного проекта по теме.

Контроль знаний студентов включает формы текущего и итогового контроля. Текущий контроль осуществляется в процессе изучения курса и включает в себя оценку работы студентов на практических занятиях (круглый стол, собеседование, групповые научные проекты, тестирование), а также подготовку домашнего задания.

2. Технологическая карта самостоятельной работы студента

Коды реализуемых компетенций, индикаторов	Вид деятельности студентов	Итоговый продукт самостоятельной работы	Средства и технологии оценки	Объем часов, в том числе		
				СРС	Контактная работа с преподавателем	Всего
5 семестр						
ИД-1ПК-1 ИД-2ПК-1 ИД-1ПК-3	Подготовка к лабораторной работе	Отчет по лабораторной работе	Собеседование	2.00	2.00	4.00
ИД-1ПК-1 ИД-2ПК-1 ИД-1ПК-3	Подготовка к лекции	Конспект лекций	Собеседование	3.00	2.00	5.00
ИД-1ПК-1 ИД-2ПК-1 ИД-1ПК-3	Подготовка к практическому занятию	Отчет по лабораторной работе	Собеседование	2.00	2.00	4.00
ИД-1ПК-1 ИД-2ПК-1 ИД-1ПК-3	Подготовка к экзамену	Экзамен	Вопросы к экзамену	3.00	2.00	5.00
Итого за семестр				10.00	8.00	18.00
Итого				10.00	8.00	18.00

3. Порядок выполнения самостоятельной работы

3.1 Методические рекомендации по изучению теоретического материала

Комплексное изучение предлагаемой студентам учебной дисциплины «Безопасность операционных систем» предполагает овладение материалами лекций, учебников, программы, творческую работу студентов в ходе проведения практических занятий, а также систематическое выполнение заданий для самостоятельной работы.

В ходе лекций раскрываются основные вопросы в рамках рассматриваемых тем, делаются акценты на наиболее сложные и интересные положения изучаемого материала, которые должны быть приняты студентами во внимание. Материалы лекций являются основой для подготовки студентов к практическим занятиям.

Работа с конспектом лекций

Просмотрите конспект сразу после занятий. Отметьте материал конспекта лекций, который вызывает затруднения для понимания. Попробуйте найти ответы на затруднительные вопросы, используя предлагаемую литературу. Если самостоятельно не удалось разобраться в материале, сформулируйте вопросы и обратитесь на текущей консультации или на ближайшей лекции за помощью к преподавателю.

Каждую неделю отводите время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам и тестам.

Выполнение практических занятий по дисциплине «Безопасность операционных систем»

На первом занятии получите у преподавателя задания по курсу, планы подготовки к практическим занятиям. Обзаведитесь всем необходимым методическим обеспечением.

Перед практическими занятиями изучите теорию вопроса, предполагаемого к исследованию, ознакомьтесь с опытом других исследователей в этом направлении.

Целью практических занятий является:

- закрепление методов приложения теории к решению практических задач анализа и синтеза психологического знания;
- проверка уровня понимания студентами вопросов, рассмотренных на лекциях и по учебной литературе, степени и качества усвоения материала;
- обучение навыкам освоения методики эксперимента и работы с нормативно-справочной литературой;
- восполнение пробелов в пройденной теоретической части курса и оказание помощи в его усвоении.

3.2 Методические рекомендации по конспектированию первоисточников

Изучение и анализ литературных источников является обязательным видом самостоятельной работы студентов. Изучение литературы по избранной теме имеет своей задачей проследить характер постановки и решения определенной проблемы различными авторами, аргументацию их выводов и обобщений, провести анализ и систематизировать полученный материал на основе собственного осмысления с целью выяснения современного состояния вопроса. Проработка отобранного материала обязательно должна идти с одновременным ведением записей прочитанного и своих замечаний. Запись может иметь как форму конспекта, так и выписок, а также картотеку положений, тезисов, идей, методик, что в дальнейшем облегчит классификацию и систематизацию полученного материала. Такого рода записи являются лучшим способом накопления и первичной обработки материал, одной из обязательных форм организации умственного труда.

Планом удобно пользоваться при подготовке к устному выступлению по выбранной теме. Каждый пункт плана должен раскрывать одну из сторон избранной темы, а весь план должен охватывать ее целиком.

Тезисы предполагают сжатое изложение основных положений текста в форме утверждения или отрицания. Они являются более совершенной формой записей и представляют основу для дискуссии. К тому же их легко запомнить.

Конспектирование – процесс мысленной переработки и письменной фиксации информации, в виде краткого изложения основного содержания, смысла какого-либо текста.

Результат конспектирования – запись, позволяющая конспектирующему немедленно или через некоторый срок с нужной полнотой восстановить полученную информацию. Конспект в переводе с латыни означает «обзор». По существу его и составлять надо как обзор, содержащий основные мысли текста без подробностей и второстепенных деталей. Конспект носит индивидуализированный характер: он рассчитан на самого автора и поэтому может оказаться малопонятным для других.

Для того чтобы осуществлять этот вид работы, в каждом конкретном случае необходимо грамотно решить следующие задачи:

1. Сориентироваться в общей композиции текста (уметь определить вступление, основную часть, заключение).
2. Увидеть логико-смысловую канву сообщения, понять систему изложения автором информации в целом, а также ход развития каждой отдельной мысли.
3. Выявить «ключевые» мысли, т. е. основные смысловые вехи, на которые «нанизано» все содержание текста.
4. Определить детализирующую информацию.
5. Лаконично сформулировать основную информацию, не перенося на письмо все целиком и дословно.

Во всяком научном тексте содержится информация 2-х видов: основная и вспомогательная. Основной является информация, имеющая наиболее существенное значение для раскрытия содержания темы или вопроса. К ней относятся: определения научных понятий, формулировки законов, теоретических принципов и т. д. Назначение вспомогательной информации – помочь читателю лучше усвоить предлагаемый материал. К этому типу информации относятся разного рода комментарии. Основную – записываем как можно полнее, вспомогательную, как правило, опускаем. Содержание конспектирования составляет переработка основной информации в целях ее обобщения и сокращения. Обобщить – значит представить ее в более общей, схематической форме, в виде тезисов, выводов, отдельных заголовков, изложения основных результатов и т. п. Читая, мы интуитивно используем некоторые слова и фразы в качестве опорных. Такие опорные слова и фразы называются ключевыми. Ключевые слова и фразы несут основную смысловую и эмоциональную нагрузку содержания текста.

Выбор ключевых слов – это первый этап смыслового свертывания, смыслового сжатия материала.

Важными требованиями к конспекту являются наглядность и обзорность записей и такое их расположение, которое давало бы возможность уяснить логические связи и иерархию понятий.

По форме конспекты подразделяются на формализованные и графические.

1. Формализованные (все записи вносятся в заранее подготовленные таблицы).

Это удобно, во-первых, при конспектировании материалов, когда перечень характеристик описываемых предметов или явлений более или менее постоянен, во-вторых, при подготовке единого конспекта по нескольким источникам. Особенно если есть необходимость сравнения отдельных данных. Разновидностью формализованного конспекта является запись, составленная в форме ответов на заранее подготовленные вопросы, обеспечивающие исчерпывающие характеристики однотипных предметов или явлений.

2. Графические (элементы конспектируемой работы располагаются в таком виде, при котором видна иерархия понятий и взаимосвязь между ними).

По каждой работе может быть не один, а несколько графических конспектов, отображающих книгу в целом и отдельные ее части. Ведение графического конспекта – наиболее совершенный способ изображения внутренней структуры книги, а сам этот процесс помогает усвоению ее содержания.

Можно выделить следующие основные **типы конспектов: плановый, текстуальный, сводный, тематический.**

Плановый – легко получить с помощью предварительно сделанного плана произведения, каждому вопросу плана отвечает определенная часть конспекта:

- а) вопросно-ответный (на пункты плана, выраженные в вопросительной форме, конспект дает точные ответы);
- б) схематичный плановый конспект (отражает логическую структуру и взаимосвязь отдельных положений).

Текстуальный – это конспект, созданный в основном из цитат.

Сводный конспект – сочетает выписки, цитаты, иногда тезисы; часть его текста может быть снабжена планом.

Тематический – дает более или менее исчерпывающий ответ (в зависимости из числа привлеченных источников и другого материала, например, своих же записей) на поставленный вопрос – тему: обзорный; хронологический.

Роль конспекта – чисто учебная: он помогает зафиксировать основные понятия и положения первичного текста и в нужный момент их воспроизвести, например, при написании реферата или подготовке к экзамену.

Способы конспектирования.

- Тезисы – это кратко сформулированные основные мысли, положения изучаемого материала. Тезисы лаконично выражают суть читаемого, дают возможность раскрыть содержание.

Приступая к освоению записи в виде тезисов, полезно в самом тексте отмечать места, наиболее

четко формулирующие основную мысль, которую автор доказывает (если, конечно, это не библиотечная книга). Часто такой отбор облегчается шрифтовым выделением, сделанным в самом тексте.

- **Линейно-последовательная запись текста.** При конспектировании линейно – последовательным способом целесообразно использование плакатно-оформительских средств, которые включают в себя следующие:

- сдвиг текста конспекта по горизонтали, по вертикали;
- выделение жирным (или другим) шрифтом особо значимых слов;
- использование различных цветов;
- подчеркивание;
- заключение в рамку главной информации.

- **Способ «вопросов – ответов».** Он заключается в том, что, поделив страницу тетради пополам вертикальной чертой, конспектирующий в левой части страницы самостоятельно формулирует вопросы или проблемы, затронутые в данном тексте, а в правой части дает ответы на них. Одна из модификаций способа «вопросов – ответов» - таблица, где место вопроса занимает формулировка проблемы, поднятой автором (лектором), а место ответа – решение данной проблемы. Иногда в таблице могут появиться и дополнительные графы: например, «мое мнение» и т. п.

- **Схема с фрагментами** – способ конспектирования, позволяющий ярче выявить структуру текста, - при этом фрагменты текста (опорные слова, словосочетания, пояснения всякого рода) в сочетании с графикой помогают созданию рационально - лаконичного конспекта.

- **Простая схема** – способ конспектирования, близкий к схеме с фрагментами, объяснений к которой конспектирующий не пишет, но должен уметь давать их устно. Этот способ требует высокой квалификации конспектирующего. В противном случае такой конспект нельзя будет использовать.

- **Параллельный способ** конспектирования. Конспект оформляется на двух листах параллельно или один лист делится вертикальной чертой пополам и записи делаются в правой и в левой части листа.

Однако лучше использовать разные способы конспектирования для записи одного и того же материала.

Комбинированный конспект – вершина овладения рациональным конспектированием. При этом умело используются все перечисленные способы, сочетая их в одном конспекте (один из видов конспекта свободно перетекает в другой в зависимости от конспектируемого текста, от желания и умения конспектирующего). Именно при комбинированном конспекте более всего проявляется уровень подготовки и индивидуальность студента.

Принципы составления конспекта прочитанного.

1. Записать все выходные данные источника: автор, название, год и место издания. Если текст взят из периодического издания (газеты или журнала), то записать его название, год, месяц, номер, число, место издания.

2. Выделить поля слева или справа, можно с обеих сторон. Слева на полях отмечаются страницы оригинала, структурные разделы статьи или книги (названия параграфов, подзаголовки и т. п.), формулируются основные проблемы. Справа – способы фиксации прочитанной информации.

Один из видов чтения – углубленное – предполагает глубокое усвоение прочитанного и часто сохранение информации в целях последующего обращения к ней. Эффективность такого чтения повышается, если прочитанное зафиксировано не только в памяти, но и на бумаге.

Психологи утверждают, что записанное лучше и полнее усваивается, прочнее откладывается в памяти. Установлено, что если прочитать 1000 слов и затем записать 50, подытоживающих прочитанное, то коэффициент усвоения будет выше, чем, если прочитать 10000 слов, не записав ни одного. Кроме того, при записи прочитанного формируется навык свертывания информации. И, наконец, чередование чтения и записывания уменьшает усталость, повышает работоспособность и производительность умственного труда.

1. Резюмирование.

Резюме – краткий итог прочитанного, содержащий его оценку. Резюме характеризует основные выводы книги, главные итоги.

Выбор языковых средств для построения резюме-выводов подчинен основной задаче свертывания информации: минимум языковых средств – максимум информации. Это обычно одно – три четких, кратких, выразительных предложения, раскрывающих, по мнению автора, самую суть описываемого объекта.

2. Аннотация.

Аннотация – краткая обобщенная характеристика печатной работы (книги, статьи), включающая иногда и его оценку. Это наикратчайшее изложение содержания первичного документа, дающее общее представление о теме.

Основное ее назначение – дать некоторое представление о книге (статье, научной работе) с тем, чтобы рекомендовать ее определенному кругу читателей или воспользоваться своими записями при выполнении работы исследовательского, реферативного характера. Поэтому аннотации не требуются изложения содержания произведения, в ней лишь перечисляются вопросы, которые освещены в первоисточнике (содержание этих вопросов не раскрывается). Аннотация отвечает на вопрос: «О чем говорится в первичном тексте?», дает представление только о главной теме и перечне вопросов, затрагиваемых в тексте первоисточника.

Текст аннотации не стандартизирован. В научной литературе можно встретить различные требования к составлению аннотаций. Например, текст справочной аннотации может включать следующие сведения:

- тип и название аннотируемого документа (монография, диссертация, сборник, статья и т. п.)
- задачи, поставленные автором аннотируемого документа
- метод, которым пользовался автор (эксперимент, сравнительный анализ, компиляция других источников)
- принадлежность автора к определенной научной школе или направлению
- структуру аннотируемого документа
- предмет и тему произведения, основные положения и выводы автора
- характеристику вспомогательных иллюстративных материалов, дополнений, приложений, справочного аппарата, включая указатели и библиографию.

3.3 Методические указания по подготовке к экзамену

Контроль и оценка знаний студентов является неотъемлемой составной частью учебно-воспитательного процесса в вузе. Экзаменационная сессия – самая ответственная пора в студенческой жизни: она является итогом большой работы студентов в семестре. Экзамен – это метод проверки знаний студентов по части или полному курсу учебной дисциплины, произведенный путем постановки устных или письменных вопросов. Он дает объективную официально фиксируемую оценку успехов студентов за определенный отрезок времени.

Экзамен преследует многогранную цель: во-первых, это – проверка знаний студента, во-вторых, они сами по себе являются важным звеном в овладении наукой, в-третьих, это продолжение учебного процесса; наконец, он имеет большое значение как фактор стимулирования глубокого изучения предмета.

Подготовка к экзамену состоит из двух взаимосвязанных этапов. Первый – систематический труд на протяжении семестра, учебного года, охватывающий все формы учебного процесса: лекции, изучение и конспектирование рекомендованной литературы, активное участие в семинарских (практических) и лабораторных занятиях.

Второй – подготовка непосредственно перед экзаменом. Она позволяет студентам за сравнительно короткий отрезок времени охватить всю перспективу изученного и лучше понять основные закономерности и явления.

Ограниченность времени (3-4 дня) для непосредственной подготовки к экзамену по предмету требует от студентов спокойно, без нервной суеты и спешки еще раз внимательно продумать изученный в течение семестра материал, тщательно отработать вопросы, недостаточно изученные или плохо понятые, с тем, чтобы по возможности устранить все пробелы в своих знаниях.

Готовиться надо по строго продуманному графику, последовательно переходя от темы к теме, не пропуская ни одну из них.

Сложные вопросы, недостаточно уясненные в процессе подготовки к экзамену, необходимо записать и получить на них разъяснения у преподавателей во время предэкзаменационных консультаций.

Следует заметить, что студенты, которые не ходят на консультации из-за отсутствия, по их мнению, сложных вопросов, поступают опрометчиво. На консультациях очень часто лектор не только отвечает на заданные вопросы, но и по собственной инициативе разъясняет наиболее трудные разделы курса.

Итак, основной задачей подготовки студентов к экзамену является систематизация знаний учебного материала, его творческое осмысливание. Важнейшим учебным пособием на этом этапе работы студента является собственный конспект прослушанных лекций и самостоятельно проработанных тем курса.

В соответствии с положением об экзаменах их, как правило, принимают преподаватели, читающие курс лекций.

Получив билет, студент должен хорошо продумать содержание поставленных вопросов. Значительное число неудачных ответов объясняется неясным пониманием поставленной проблемы. Правильное понимание вопроса обеспечит успех при ответе на него. При подготовке к ответу на билет нужно составить развернутый план по каждому вопросу.

Отвечая на вопросы экзаменационного билета, не следует спешить. Надо излагать материал спокойно, не торопясь, владеть собой, следить за построением фраз. Следует избегать подходов издали, общих рассуждений. Рекомендуются строить ответы четко, последовательно, конкретно, по возможности исчерпывающе. Вместе с тем весьма желательно быстро и правильно иллюстрировать свой ответ примерами.

От экзаменуемого требуется: определение понятий, обоснование выдвинутых положений, свободное оперирование фактическим материалом, дать альтернативные подходы по отдельным проблемам. Логичность, стройность, литературная грамотность изложения являются неотъемлемыми чертами полноценного ответа. Нельзя при ответе допускать ни излишней краткости, переходящей в схематизм, ни многословия. И то, и другое не оправдано. Краткость не дает преподавателю возможности понять, владеет ли студент учебным материалом, а многословие может показать, что студент не умеет акцентировать внимание на главном и говорит слишком расплывчато.

Вопросы к экзамену

Базовый уровень:**Вопросы для проверки уровня обученности**

1. Структура машины фон Неймана
2. Принципы построения вычислительных машин
3. Классификация программного обеспечения
4. Понятие операционной системы
5. Иерархическая структура построения ОС
6. Виды систем обработки данных
7. Однопрограммные режимы обработки данных
8. Многопрограммные режимы обработки данных
9. Понятие ресурса и процесса
10. Процессы пользователей и системные процессы
11. Понятие и особенности многозадачности в ОС
12. Трехуровневая архитектура «клиент-сервер»
13. Понятие ядра ОС
14. Адресное пространство процесса
15. Состояния процесса
16. Блок управления процессом
17. Одноразовые операции
18. Родительские и дочерние процессы
19. Потоки
20. Структура реестра Windows
21. Виртуальная память
22. 32х и 64х битная архитектура
23. Режим ядра и пользовательский режим
24. Понятие реестра Windows
25. Архитектура Windows
26. Симметричная и асимметричная процессорная обработка
27. Подсистема окружения
28. Подсистема Windows
29. Объекты ядра Windows
30. Драйверы устройств
31. Прерывания
32. Основные классы прерываний
33. Обработка прерываний процессором
34. Понятие микроядра ОС
35. Задачи протокола аутентификации
36. Локальная и доменная регистрация
37. Протоколы аутентификации Windows
38. Угрозы безопасности ОС.
39. Классификация угроз безопасности ОС. Наиболее распространенные угрозы
40. Требования к защите ОС.
41. Понятие защищенной ОС. Подходы к организации защиты
42. Этапы построения защиты.
43. Административные меры защиты
44. Стандарты безопасности ОС
45. Основные понятия программно-технического уровня информационной безопасности

Уметь, Владеть:

- оцените эффективность и надежность защиты ОС;
- выявите слабости защиты ОС и использовать их для вскрытия защиты;
- спланируйте политику безопасности ОС;

- воспользуйтесь средствами защиты, предоставляемыми ОС;
- проводите анализ и оценивание механизмов защиты.
- постройте защиту ОС Windows, Unix.
- настройте безопасность указанных ОС, а также поиска и устранения проблем в ОС, их обслуживания.

Повышенный уровень:

Знать:

1. Требования к защите компьютерной информации.
2. Классификация требований к системам защиты.
3. Формализованные требования к защите информации от НСД.
4. Общие подходы к построению систем защиты компьютерной информации.
5. Различия требований и основополагающих механизмов защиты от НСД.
6. Требования к защите ОС. Понятие защищенной ОС.
7. Подходы к организации защиты ОС и их недостатки.
8. Этапы построения защиты. Административные меры защиты. Стандарты безопасности ОС.
9. Разграничение доступа в ОС. Субъекты, объекты, методы и права доступа.
10. Привилегии субъектов доступа. Избирательное и полномочное разграничение доступа, изолированная программная среда.
11. Примеры реализации разграничения доступа в современных ОС.
12. Идентификация и аутентификация пользователей ОС.
13. Понятия идентификации и аутентификации пользователей.
14. Аутентификация на основе паролей, методы подбора паролей, средства и методы повышения защищенности ОС от подбора паролей.
15. Аутентификация на основе внешних носителей ключа, биометрических характеристик пользователя.
16. Примеры реализации идентификации и аутентификации в современных ОС.
17. Аудит в ОС. Необходимость аудита.
18. Требования к подсистеме аудита. Примеры реализации аудита в современных ОС.
19. Системы защиты программного обеспечения.

Уметь, Владеть:

- оцените эффективность и надежность защиты ОС;
- выявите слабости защиты ОС и использовать их для вскрытия защиты;
- настройте безопасности указанных ОС;

Список рекомендуемой литературы

1. Основная литература

1. Ларина, Т. Б. Администрирование операционных систем. Управление системой : учебное пособие / Т. Б. Ларина. — Москва : РУТ (МИИТ), 2020. — 71 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/175980>

2. Ларина, Т. Б. Виртуализация операционных систем : учебное пособие / Т. Б. Ларина. — Москва : РУТ (МИИТ), 2020. — 65 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/175964>

2. Дополнительная литература

1. Учеб.пособие для вузов*]. — М. : Финансы и статистика, 2004. — 272с. : ил. — Библиогр.: с. 265. — ISBN 5-279-02436-8

2. Гриценко, Ю.Б. Операционные системы : в 2-х ч. / Ю.Б. Гриценко ; Федеральное агентство по образованию, Томский межвузовский центр дистанционного образования (ТУСУР). Кафедра автоматизации обработки информации (АОИ). – Томск : Томский государственный университет систем управления и радиоэлектроники, 2009. – Ч. 2. – 235 с. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=208655>

3. Методическая литература

1. Безбогов, А.А. Безопасность операционных систем : учебное пособие / А.А. Безбогов, А.В. Яковлев, Ю.Ф. Мартемьянов. – М.: "Издательство Машино-строение-1", 2013.

2. Курс лекций.

3. Сборник лабораторных работ.

4. Интернет-ресурсы

www.netacad.com

5. Программное обеспечение

1. VMWare/VirtualBox

2. Windows7

3. FreeBSD

4. Ubuntu