

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Шебзухова Татьяна Александровна

Должность: Директор Пятигорского института (филиал) Северо-Кавказского  
федерального университета

Дата подписания: 21.05.2025 11:56:27

Уникальный программный ключ: «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

d74ce93cd40e39275c3ba2f58486412a1c8ef96f

Пятигорский институт (филиал) СКФУ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

## **Методические указания**

для обучающихся по организации и проведению самостоятельной работы  
по дисциплине «**ЗАЩИЩЕННЫЕ ЛОКАЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ**»  
для студентов направления подготовки **10.03.01 Информационная  
безопасность**  
направленность (профиль) **Безопасность компьютерных систем**

**Пятигорск, 2025**

## СОДЕРЖАНИЕ

1. Общие положения	3
2. Цель и задачи самостоятельной работы	4
3. Технологическая карта самостоятельной работы студента	5
4. Порядок выполнения самостоятельной работы студентом	5
4.1. Методические рекомендации по работе с учебной литературой	5
4.2. Методические рекомендации по подготовке к практическим и лабораторным занятиям	7
4.3. Методические рекомендации по самопроверке знаний	7
4.4. Методические рекомендации по написанию научных текстов (докладов, докладов, эссе, научных статей и т.д.)	7
4.5. Методические рекомендации по выполнению исследовательских проектов	10
4.6. Методические рекомендации по подготовке к экзаменам и зачетам	13
5. Контроль самостоятельной работы студентов	14
6. Список литературы для выполнения СРС	14

## 1. Общие положения

Самостоятельная работа - планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа студентов (СРС) в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения.

К основным видам самостоятельной работы студентов относятся:

- формирование и усвоение содержания конспекта лекций на базе рекомендованной лектором учебной литературы, включая информационные образовательные ресурсы (электронные учебники, электронные библиотеки и др.);
- написание докладов;
  - подготовка к семинарам, практическим и лабораторным работам, их оформление;
  - составление аннотированного списка статей из соответствующих журналов по отраслям знаний (педагогических, психологических, методических и др.);
- выполнение учебно-исследовательских работ, проектная деятельность;
- подготовка практических разработок и рекомендаций по решению проблемной ситуации;
- выполнение домашних заданий в виде решения отдельных задач, проведения типовых расчетов, расчетно-компьютерных и индивидуальных работ по отдельным разделам содержания дисциплин и т.д.;
- компьютерный текущий самоконтроль и контроль успеваемости на базе электронных обучающих и аттестующих тестов;
- выполнение курсовых работ (проектов) в рамках дисциплин;
- выполнение выпускной квалификационной работы и др.

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей изучаемой дисциплины, объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных качеств студентов и условий учебной деятельности.

Процесс организации самостоятельной работы студентов включает в себя следующие этапы:

- подготовительный (определение целей, составление программы, подготовка методического обеспечения, подготовка оборудования);
- основной (реализация программы, использование приемов поиска информации, усвоения, переработки, применения, передачи знаний, фиксирование результатов, самоорганизация процесса работы);
- заключительный (оценка значимости и анализ результатов, их систематизация, оценка эффективности программы и приемов работы, выводы о направлениях оптимизации труда).

Самостоятельная работа по дисциплине Защищенные локальные вычислительные сети направлена на формирование следующих **компетенций**:

Код	Формулировка:
ПК-4	Способность участвовать в работах по реализации политики информационной безопасности, применять комплексный подход к обеспечению информационной безопасности объекта защиты
ПК-2	Способность применять программные средства системного, прикладного и специального назначения, инструментальные средства, языки и системы программирования для решения профессиональных задач
ПК-12	Способность принимать участие в проведении экспериментальных исследований системы защиты информации

## 2. Цель и задачи самостоятельной работы

Ведущая цель организации и осуществления СРС совпадает с целью обучения студента – формирование набора общенаучных, профессиональных и специальных компетенций будущего бакалавра по соответствующему направлению подготовки

При организации СРС важным и необходимым условием становятся формирование умения самостоятельной работы для приобретения знаний, навыков и возможности организации учебной и научной деятельности. Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СРС являются:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;
- использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на практических и практических занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

### 3. Технологическая карта самостоятельной работы студента

Коды реализуемых компетенций	Вид деятельности студентов	Средства и технологии оценки	Объем часов, в том числе (акад.)		
			СРС	Контактная работа с преподавателем	Всего
<b>3 семестр</b>					
ПК-4 (ИД-1, ИД-2, ИД-3) ПК-2(ИД-1, ИД-2, ИД-3) ПК-12(ИД-1, ИД-2, ИД-3)	Самостоятельное изучение литературы и источников	Собеседование	52,56	5,84	58,4
ПК-4 (ИД-1, ИД-2, ИД-3) ПК-2(ИД-1, ИД-2, ИД-3) ПК-12(ИД-1, ИД-2, ИД-3)	Подготовка практическим занятиям	Защита ПР	3,24	0,36	3,6
ПК-4 (ИД-1, ИД-2, ИД-3) ПК-2(ИД-1, ИД-2, ИД-3) ПК-12(ИД-1, ИД-2, ИД-3)	Написание реферата/доклада	Защита доклада	9	1	10
<b>Итого за 3 семестр</b>			64,8	7,2	72
<b>4 семестр</b>					
ПК-4 (ИД-1, ИД-2, ИД-3) ПК-2(ИД-1, ИД-	Самостоятельное изучение литературы и источников	Собеседование	3,96	0,44	4,4

2,ИД-3) ПК-12(ИД-1,ИД-2,ИД-3)					
ПК-4 (ИД-1,ИД-2,ИД-3) ПК-2(ИД-1,ИД-2,ИД-3) ПК-12(ИД-1,ИД-2,ИД-3)	Подготовка к практическим занятиям	Защита ПР	3,24	0,36	3,6
ПК-4 (ИД-1,ИД-2,ИД-3) ПК-2(ИД-1,ИД-2,ИД-3) ПК-12(ИД-1,ИД-2,ИД-3)	Написание реферата/доклада	Защита доклада	9	1	10
<b>Итого за 4 семестр</b>			<b>16,2</b>	<b>1,8</b>	<b>18</b>
<b>Итого</b>			<b>81</b>	<b>9</b>	<b>90</b>

#### 4. Порядок выполнения самостоятельной работы студентом

##### 4.1. Методические рекомендации по работе с учебной литературой

При работе с книгой необходимо подобрать литературу, научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги.

Важно помнить, что рациональные навыки работы с книгой - это всегда большая экономия времени и сил.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс. Необходимая литература может быть также указана в методических разработках по данному курсу.

Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного уяснения предыдущего, описывая на бумаге все выкладки и вычисления (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода).

При изучении любой дисциплины большую и важную роль играет самостоятельная индивидуальная работа.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного

представления о том, что изучаешь. Полезно составлять опорные конспекты. При изучении материала по учебнику полезно в тетради (на специально отведенных полях) дополнять конспект лекций. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем.

Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при перечитывании записей лучше запоминались.

Опыт показывает, что многим студентам помогает составление листа опорных сигналов, содержащего важнейшие и наиболее часто употребляемые формулы и понятия. Такой лист помогает запомнить формулы, основные положения лекции, а также может служить постоянным справочником для студента.

Чтение научного текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации. От того насколько осознанно читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия.

Выделяют **четыре основные установки в чтении научного текста:**

информационно-поисковый (задача – найти, выделить искомую информацию)

усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить, как сами сведения, излагаемые автором, так и всю логику его рассуждений)

аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему)

творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

*Основные виды систематизированной записи прочитанного:*

Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения;

Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала;

Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;

Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;

Конспектирование – краткое и последовательное изложение содержания прочитанного.

Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

*Методические рекомендации по составлению конспекта:*

1. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта;

2. Выделите главное, составьте план;

3. Кратко сформулируйте основные положения текста, отметьте аргументацию автора;

4. Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.

5. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли.

В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Для уточнения и дополнения необходимо оставлять поля.

Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы.

#### *4.2. Методические рекомендации по подготовке к практическим и лабораторным занятиям*

Для того чтобы практические и лабораторные занятия приносили максимальную пользу, необходимо помнить, что упражнение и решение задач проводятся по вычитанному на лекциях материалу и связаны, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться на практических занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

#### *4.3. Методические рекомендации по самопроверке знаний*

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих задач на практических занятиях и самостоятельно студенту рекомендуется, провести самопроверку усвоенных знаний, ответив на контрольные вопросы по изученной теме.

В случае необходимости нужно еще раз внимательно разобраться в материале.

Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала - умение решать задачи или пройти тестирование по пройденному материалу. Однако следует помнить, что правильное решение задачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

#### 4.4. Методические рекомендации по написанию научных текстов (докладов, эссе, научных статей и т.д.)

Перед тем, как приступить к написанию научного текста, важно разобраться, какова истинная цель вашего научного текста - это поможет вам разумно распределить свои силы и время.

Во-первых, сначала нужно определиться с идеей научного текста, а для этого необходимо научиться либо относиться к разным явлениям и фактам несколько критически (своя идея – как иная точка зрения), либо научиться увлекаться какими-то известными идеями, которые нуждаются в доработке (идея – как оптимистическая позиция и направленность на дальнейшее совершенствование уже известного). Во-вторых, научиться организовывать свое время, ведь, как известно, свободное (от всяких глупостей) время – важнейшее условие настоящего творчества, для него наконец-то появляется время. Иногда именно на организацию такого времени уходит немалая часть сил и талантов.

Писать следует ясно и понятно, стараясь основные положения формулировать четко и недвусмысленно (чтобы и самому понятно было), а также стремясь структурировать свой текст. Каждый раз надо представлять, что ваш текст будет кто-то читать и ему захочется сориентироваться в нем, быстро находить ответы на интересующие вопросы (заодно представьте себя на месте такого человека). Понятно, что работа, написанная «сплошным текстом» (без заголовков, без выделения крупным шрифтом наиболее важным мест и т. п.), у культурного читателя должна вызывать брезгливость и даже жалость к автору (исключения составляют некоторые древние тексты, когда и жанр был иной и к текстам относились иначе, да и самих текстов было гораздо меньше – не то, что в эпоху «информационного взрыва» и соответствующего «информационного мусора»).

Объем текста и различные оформительские требования во многом зависят от принятых в конкретном учебном заведении порядков.

Доклад - это самостоятельное исследование студентом определенной проблемы, комплекса взаимосвязанных вопросов.

Доклад не должна составляться из фрагментов статей, монографий, пособий. Кроме простого изложения фактов и цитат, в доклад е должно проявляться авторское видение проблемы и ее решения.

Рассмотрим основные этапы подготовки  
а студентом.

Выполнение доклада начинается с выбора темы.

Затем студент приходит на первую консультацию к руководителю, которая предусматривает:

- обсуждение цели и задач работы, основных моментов избранной темы;
- консультирование по вопросам подбора литературы;
- составление предварительного плана.

Следующим этапом является работа с литературой. Необходимая литература подбирается студентом самостоятельно.

После подбора литературы целесообразно сделать рабочий вариант плана работы. В нем нужно выделить основные вопросы темы и параграфы, раскрывающие их содержание.

Составленный список литературы и предварительный вариант плана уточняются, согласуются на очередной консультации с руководителем.

Затем начинается следующий этап работы - изучение литературы. Только внимательно читая и конспектируя литературу, можно разобраться в основных вопросах темы и подготовиться к самостоятельному (авторскому) изложению содержания доклада. Конспектируя первоисточники, необходимо отразить основную идею автора и его позицию по исследуемому вопросу, выявить проблемы и наметить задачи для дальнейшего изучения данных проблем.

Систематизация и анализ изученной литературы по проблеме исследования позволяют студенту написать работу.

Рабочий вариант текста доклада предоставляется руководителю на проверку. На основе рабочего варианта текста руководитель вместе со студентом обсуждает возможности доработки текста, его оформление. После доработки доклад сдается на кафедру для его оценивания руководителем.

#### *Требования к написанию доклада*

Написание 1 доклада является обязательным условием выполнения плана СРС по любой дисциплине профессионального цикла.

Тема доклада может быть выбрана студентом из предложенных в рабочей программе или фонде оценочных средств дисциплины, либо определена самостоятельно, исходя из интересов студента (в рамках изучаемой дисциплины). Выбранную тему необходимо согласовать с преподавателем.

Доклад должен быть написан научным языком.

Объем доклада должен составлять 20-25 стр.

#### *Структура доклада:*

- Введение (не более 3-4 страниц). Во введении необходимо обосновать выбор темы, ее актуальность, очертить область исследования, объект исследования, основные цели и задачи исследования.

- Основная часть состоит из 2-3 разделов. В них раскрывается суть исследуемой проблемы, проводится обзор мировой литературы и источников Интернет по предмету исследования, в котором дается характеристика степени разработанности проблемы и авторская аналитическая оценка основных теоретических подходов к ее решению. Изложение материала не должно ограничиваться лишь описательным подходом к раскрытию выбранной темы. Оно также должно содержать собственное видение рассматриваемой проблемы и изложение собственной точки зрения на возможные пути ее решения.

- Заключение (1-2 страницы). В заключении кратко излагаются достигнутые при изучении проблемы цели, перспективы развития исследуемого вопроса

- Список использованной литературы (не меньше 10 источников), в алфавитном порядке, оформленный в соответствии с принятыми правилами. В список использованной литературы рекомендуется включать работы отечественных и зарубежных авторов, в том числе статьи, опубликованные в научных журналах в течение последних 3-х лет и ссылки на ресурсы сети Интернет.

- Приложение (при необходимости).

#### *Требования к оформлению:*

- текст с одной стороны листа;
- шрифт Times New Roman;
- кегль шрифта 14;
- межстрочное расстояние 1,5;
- поля: сверху 2,5 см, снизу – 2,5 см, слева - 3 см, справа 1,5 см;
- доклад должен быть представлен в сброшюрованном виде.

#### *Порядок защиты доклада:*

Защита доклада проводится на практических занятиях, после окончания работы студента над ним и исправления всех недочетов, выявленных преподавателем в ходе консультаций. На защиту доклада отводится 5-7 минут времени, в ходе которого студент должен показать свободное владение материалом по заявленной теме. При защите доклада приветствуется использование мультимедиа-презентации.

#### *Оценка доклада*

Доклад оценивается по следующим критериям:

- соблюдение требований к его оформлению;

- необходимость и достаточность для раскрытия темы приведенной в тексте доклада информации;
- умение студента свободно излагать основные идеи, отраженные в докладе;
- способность студента понять суть задаваемых преподавателем и сокурсниками вопросов и сформулировать точные ответы на них.

*Критерии оценки:*

*Оценка «отлично»* выставляется студенту, если в докладе студент исчерпывающе, последовательно, четко и логически стройно излагает материал; свободно справляется с задачами, вопросами и другими видами применения знаний; использует для написания доклада современные научные материалы; анализирует полученную информацию; проявляет самостоятельность при написании доклада.

*Оценка «хорошо»* выставляется студенту, если качество выполнения доклада достаточно высокое. Студент твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопросы по теме доклада.

*Оценка «удовлетворительно»* выставляется студенту, если материал доклада излагается частично, но пробелы не носят существенного характера, студент допускает неточности и ошибки при защите доклада, дает недостаточно правильные формулировки, наблюдаются нарушения логической последовательности в изложении материала.

*Оценка «неудовлетворительно»* выставляется студенту, если он не подготовил доклад или допустил существенные ошибки. Студент неуверенно излагает материал доклада, не отвечает на вопросы преподавателя.

*Описание шкалы оценивания*

Максимально возможный балл за весь текущий контроль устанавливается равным 55. Текущее контрольное мероприятие считается сданным, если студент получил за него не менее 60% от установленного для этого контроля максимального балла. Рейтинговый балл, выставляемый студенту за текущее контрольное мероприятие, сданное студентом в установленные графиком контрольных мероприятий сроки, определяется следующим образом:

Уровень выполнения контрольного задания	Рейтинговый балл (в % от максимального балла за контрольное задание)
Отличный	100
Хороший	80
Удовлетворительный	60
Неудовлетворительный	0

#### *4.5. Методические рекомендации по выполнению исследовательских проектов*

Исследовательская проектная работа – это групповая работа, для выполнения которой необходим выбор и приложение научной методики к поставленной задаче, получение собственного теоретического или экспериментального материала, на основании которого необходимо провести анализ и сделать выводы об исследуемом явлении. Выполнение проекта – это всегда коллективная, творческая практическая работа, предназначенная для получения определенного продукта или научно-технического результата. Такая работа подразумевает четкое, однозначное формирование поставленной задачи, определение сроков выполнения намеченного, определение требований к разрабатываемому объекту.

Выполнение 1 группового проекта является обязательным условием выполнения самостоятельной работы по любой дисциплине профессионального цикла. Тема проектного задания может быть выбрана студентом из предложенных в рабочей программе или фонде

оценочных средств дисциплины, либо определена самостоятельно, исходя из интересов студента (в рамках изучаемой дисциплины). Выбранную тему необходимо согласовать с преподавателем.

*Требования по выполнению и оформлению проекта*

При выполнении проекта приветствуется работа в группе (2-3 человека). Проект – это исследовательская работа, в ходе которой студенты должны продемонстрировать владение навыками научного исследования, умения проводить анализ, обобщать информацию, делать выводы, предлагать свои решения проблемы, рассматриваемой в проекте.

При подготовке материалов проекта студенты должны продемонстрировать владение современными методами компьютерной обработки данных.

*Критерии оценки работы участника проекта.*

Для каждого из участников проекта оцениваются:

- профессиональные теоретические знания в соответствующей области;
- умение работать со справочной и научной литературой, осуществлять поиск необходимой информации в Интернет;
- умение работать с техническими средствами;
- умение пользоваться соответствующими выполняемому проекту информационными технологиями;
- умение готовить материалы проекта для презентации: составлять и редактировать тексты, формировать презентацию проекта;
- умение работать в команде;
- умение публично представлять результаты собственной деятельности;
- коммуникабельность, инициативность, творческие способности.

*Критерии выставления оценки участникам проекта*

Оценка	Профессиональные компетенции	Компетенции, связанные с использованием соответствующих выполняемому проекту технических средств и информационных технологий	Иные универсальные компетенции (коммуникабельность, инициативность, умение работать в «команде», управленческие навыки и т.д.)	Отчетность
«Отлично»	Работа выполнена на высоком профессиональном уровне. Представленный материал в основном фактически верен, допускаются негрубые фактические неточности. Студент свободно отвечает на вопросы, связанные с проектом.	Технические средства и информационные технологии освоены и использованы для реализации проекта полностью	Студент проявил инициативу, творческий подход, способность к выполнению сложных заданий, навыки работы в коллективе, организационные способности.	Проект представлен полностью и в срок.
«Хорошо»	Работа выполнена на	Обнаруживаются	Студент	Проект

Оценка	Профессиональные компетенции	Компетенции, связанные с использованием соответствующих выполняемому проекту технических средств и информационных технологий	Иные универсальные компетенции (коммуникабельность, инициативность, умение работать в «команде», управленческие навыки и т.д.)	Отчетность
	достаточно высоком профессиональном уровне. Допущено до 4–5 фактических ошибок. Студент отвечает на вопросы, связанные с проектом, но недостаточно полно.	некоторые ошибки в использовании соответствующих технических средств и информационных технологий	достаточно полно, но без инициативы и творческих находок выполнил возложенные на него задачи.	представлен достаточно полно и в срок, но с некоторыми недоработками.
«Удовлетворительно»	Уровень недостаточно высок. Допущено до 8 фактических ошибок. Студент может ответить лишь на некоторые из заданных вопросов, связанных с проектом.	Обнаруживает недостаточное владение навыками работы с техническими средствами и соответствующим и информационным и технологиями	Студент выполнил большую часть возложенной на него работы.	Проект сдан со значительным опозданием (более недели) и не полностью
«Неудовлетворительно»	Работа не выполнена или выполнена на низком уровне. Допущено более 8 фактических ошибок. Ответы на связанные с проектом вопросы обнаруживают непонимание предмета и отсутствие ориентации в материале проекта.	Навыков работы с техническими средствами нет, информационные технологии не освоены	Студент практически не работал, не выполнил свои задачи или выполнил лишь отдельные не существенные поручения в групповом проекте.	Проект не сдан.

*Студенты должны:* защитить проект в режиме презентации, предъявить файлы выполненного проекта, уметь рассказать о технологиях, использованных ими при выполнении проекта, дать оценку работы каждого члена группы (*если проект групповой*).

Максимально возможный балл за весь текущий контроль устанавливается равным **55**. Текущее контрольное мероприятие считается сданным, если студент получил за него не менее 60% от установленного для этого контроля максимального балла. Рейтинговый

балл, выставляемый студенту за текущее контрольное мероприятие, сданное студентом в установленные графиком контрольных мероприятий сроки, определяется следующим образом:

Уровень выполнения контрольного задания	Рейтинговый балл (в % от максимального балла за контрольное задание)
Отличный	100
Хороший	80
Удовлетворительный	60
Неудовлетворительный	0

#### *4.6. Методические рекомендации по подготовке к экзаменам и зачетам*

Изучение многих общепрофессиональных и специальных дисциплин завершается экзаменом. Подготовка к экзамену способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к экзамену, студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На экзамене студент демонстрирует то, что он приобрел в процессе обучения по конкретной учебной дисциплине.

Экзаменационная сессия - это серия экзаменов, установленных учебным планом. Между экзаменами интервал 3-4 дня. Не следует думать, что 3-4 дня достаточно для успешной подготовки к экзаменам.

В эти 3-4 дня нужно систематизировать уже имеющиеся знания. На консультации перед экзаменом студентов познакомят с основными требованиями, ответят на возникшие у них вопросы. Поэтому посещение консультаций обязательно.

Требования к организации подготовки к экзаменам те же, что и при занятиях в течение семестра, но соблюдаться они должны более строго. Во-первых, очень важно соблюдение режима дня; сон не менее 8 часов в сутки, занятия заканчиваются не позднее, чем за 2-3 часа до сна. Оптимальное время занятий - утренние и дневные часы. В перерывах между занятиями рекомендуются прогулки на свежем воздухе, неутомительные занятия спортом. Во-вторых, наличие хороших собственных конспектов лекций. Даже в том случае, если была пропущена какая-либо лекция, необходимо вовремя ее восстановить (переписать ее на кафедре), обдумать, снять возникшие вопросы для того, чтобы запоминание материала было осознанным. В-третьих, при подготовке к экзаменам у студента должен быть хороший учебник или конспект литературы, прочитанной по указанию преподавателя в течение семестра. Здесь можно эффективно использовать листы опорных сигналов.

Вначале следует просмотреть весь материал по сдаваемой дисциплине, отметить для себя трудные вопросы. Обязательно в них разобраться. В заключение еще раз целесообразно повторить основные положения, используя при этом листы опорных сигналов.

Систематическая подготовка к занятиям в течение семестра позволит использовать время экзаменационной сессии для систематизации знаний.

### **Контроль самостоятельной работы студентов**

Контроль самостоятельной работы проводится преподавателем в аудитории.

Предусмотрены следующие виды контроля: собеседование, оценка доклада, оценка презентации, оценка участия в круглом столе, оценка выполнения проекта.

Подробные критерии оценивания компетенций приведены в Фонде оценочных средств для проведения текущей и промежуточной аттестации.

### Список литературы для выполнения СРС

#### Основная литература:

- 1 Воронцова, В. Л. Лекции по системному и прикладному программному обеспечению: Учебное пособие / В. Л. Воронцова, И. А. Журавлева; Ставроп. гос. ун-т. – Ставрополь: СГУ, 2004. – 208 с. – Библиогр: с. 205-206
- 2 Защита информации в операционных системах: лабораторные работы: учеб.- метод. пособие / М. В. Романкова / сост.: И. В. Зайцева / Федеральное агентство по образованию Ставроп. гос. ун-т. – Ставрополь : Изд-во СГУ, 2008. – 99 с. – Библиогр.: с. 96-97
- 3 Программно-аппаратные средства обеспечения информационной безопасности: учеб. пособие / сост. В. И. Петренко, Н. В. Куклева ; Федер. агентство по образованию, Ставроп. гос. ун-т, Ч. 2, Защита в операционных системах. – Ставрополь: Изд-во СГУ, 2007. – 154 с. – Библиогр: с. 154

#### Дополнительная литература:

- 1 Безручко, В. Т. Практикум по курсу "Информатика". Работа в Windows, Word, Excel [Учеб. пособие для вузов\*]. – М.: Финансы и статистика, 2004. – 272с.: ил. – Библиогр.: 265. – ISBN 5-279-02436-8
- 2 Гриценко, Ю.Б. Операционные системы: в 2-х ч. / Ю.Б. Гриценко ; Федеральное агентст по образованию, Томский межвузовский центр дистанционного образования (ТУСУР) Кафедра автоматизации обработки информации (АОИ). Томск: Томский государственный университет систем управления и радиоэлектроники, 2009. – Ч. 2. – 2 с.– Режим доступа: по подписке.– URL: <http://biblioclub.ru/index.php?page=book&id=208655>

#### Методические рекомендации для самостоятельной работы студентов по дисциплине

1. Методические рекомендации по выполнению практических работ по дисциплине «Защищенные локальные вычислительные сети».

2. Методические рекомендации по организации самостоятельной работы студентов по дисциплине «Защищенные локальные вычислительные сети».

#### Интернет-ресурсы:

1. <http://el.ncfu.ru/> – система управления обучением ФГАОУ ВО СКФУ. Дистанционная поддержка дисциплины «Цифровая грамотность и обработка больших данных»

2. <http://www.un.org> - Сайт ООН Информационно-коммуникационные технологии

3. <http://www.intuit.ru> – Интернет-Университет Компьютерных технологий.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Пятигорский институт (филиал) СКФУ

## **Методические указания**

по выполнению практических работ

по дисциплине

**«ЗАЩИЩЕННЫЕ ЛОКАЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ»**

для направления подготовки **10.03.01 Информационная безопасность**  
направленность (профиль) **Безопасность компьютерных систем**

**Пятигорск**  
**2025**

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
<b>1. Цель и задачи изучения дисциплины.....</b>	<b>3</b>
<b>2. Оборудование и материалы.....</b>	<b>3</b>
<b>3. Наименование практических работ.....</b>	<b>3</b>
<b>4. Содержание практических работ.....</b>	<b>4</b>
Практическая работа 1.....	4
Практическая работа 2.....	8
Практическая работа 3.....	12
Практическая работа 4.....	16
Практическая работа 5.....	18
Практическая работа 6.....	21
Практическая работа 7.....	25
Практическая работа 8.....	28
Практическая работа 9.....	30
<b>5. Учебно-методическое и информационное обеспечение дисциплины.....</b>	<b>33</b>
<b>5.1. Перечень основной и дополнительной литературы, необходимой для освоения дисциплины.....</b>	<b>33</b>
<b>5.2. Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине (модулю).....</b>	<b>33</b>
<b>5.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля).....</b>	<b>33</b>

## ВВЕДЕНИЕ

### 1. Цель и задачи изучения дисциплины

Целью изучения дисциплины «Защищенные локальные вычислительные сети» 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»

### 2. Оборудование и материалы

Аппаратные средства: персональный компьютер;

Программные средства: Альт Рабочая станция 10, Альт Рабочая станция К, Альт «Сервер», Пакет офисных программ - Р7-Офис.

Учебный класс оснащен IBM-совместимыми компьютерами, объединенными в локальную сеть. Локальная сеть учебного класса имеет постоянный доступ к сети Internet по выделенной линии. Для проведения лабораторных работ необходимо следующее программное обеспечение: операционная система Альт Рабочая станция, пакет офисных программ Р7-Офис.

### 3. Наименование лабораторных работ

Не предусмотрено рабочим планом

#### 5.4 Наименование практических занятий

№ Темы дисциплины	Наименование тем дисциплины, их краткое содержание	Объем часов	Из них практическая подготовка, часов
3 семестр			
<b>Тема 1: Безопасность компьютерных сетей</b>			
1	«Конфигурирование и проверка IPsec VPN между двумя пунктами с помощью интерфейса командной строки	2	
1	Настройка межсетевого экрана Cisco ASA 5505	2	
<b>Тема 3: Протоколирование и аудит</b>			
3	Исследование процессов создания бюджетов пользователей в ОС Windows NT и Unix	2	
3	Исследование методов интеграции сетей Microsoft и Unix с использованием сервера Samba	2	
<b>Тема 5: Системы безопасности ОС Linux</b>			
5	Изучение базовых команд ОС Linux	4	
<b>Тема 7: Модели безопасности основных операционных систем</b>			
7	Классификация защиты семейства ОС Windows	6	
<b>Итого за 3 семестр</b>		<b>18</b>	
4 семестр			
<b>Тема 11. Архитектура ОС Linux</b>			
11	Настройка ОС Linux	2	
11	Работа с файлами в Linux	2	
11	Работа с файлами в Linux	2	

<b>Тема 12. Архитектура ОС Windows</b>			
12	Работа с реестром Windows	2	
12	Работа в командной строке	2	
12	Пакетные файлы	2	
12	Оснастки консоли	2	
<b>Тема 14. Управление файлами и вводом-выводом в ОС</b>			
14	Файловая система Windows	2	
14	Файловая система UNIX и Linux	2	
<b>Итого за 4 семестр</b>		<b>18</b>	
<b>Итого</b>		<b>36</b>	

#### **4. Содержание практических работ**

##### **Практическая работа 1 «Конфигурирование и проверка IPsec VPN между двумя пунктами с помощью интерфейса командной строки**

В ходе практической работы: требуется построить соответствующий проект в Cisco Packet Tracer, топология сети включает три маршрутизатора. Ваша задача заключается в том, чтобы настроить маршрутизаторы R1 и R3 для поддержки сети site-to-site IPsec VPN (при передаче трафика между соответствующими локальными сетями. Туннель IPsec VPN проходит от маршрутизатора R1 к маршрутизатору R3 через R2. Маршрутизатор R2 играет роль транзитного узла и не имеет информации о VPN. IPSec обеспечивает безопасную передачу конфиденциальной информации по незащищенным сетям.

##### **Практическая работа 2 «Настройка межсетевого экрана Cisco ASA 5505»**

Целью практической работы является изучение работы межсетевого экрана Cisco ASA 5505 конфигурации. В ходе выполнения практической работы необходимо активировать пользовательские порты на R1 и S1. Настроить Ethernet интерфейсы персональных компьютеров. Настроить сетевые интерфейсы PC1, PC2, PC3, PC4. Создать VLAN'ы согласно топологии сети. Произвести базовую настройку межсетевого экрана Cisco ASA 5505 Результаты о проделанной работе показать преподавателю. Раздел «Итоговый контроль» представляет собой самостоятельное выполнения заданий по вариантам. Каждый вариант содержит по 3 задания, и дана общая топология сети для всех вариантов. В разделе «Литература» написаны учебные пособия и книги по дисциплине «Безопасность компьютерных сетей», которые помогут освоить дисциплину. Раздел «Оборудование». Описывается лаборатория, компьютерный класс, и весь технический парк на котором будет производиться работы по обучению студентов и подготовки их к WorldSkills Russia.

Технопарк оснащён следующим оборудованием:

- маршрутизаторы Cisco 2911 (количество 6);
- маршрутизаторы Cisco 2811 (количество 9);
- коммутаторы Cisco Catalyst 2960 (количество 9);
- межсетевые экраны Cisco ASA 5505 (количество 6);
- блейд-система HP c7000;
- сервер HP ProLiant BL 460c G7 2x 5675;
- консольный сервер;
- 12 рабочих мест;
- IP-телефоны Cisco CP-7961G (количество 6).

Маршрутизатор Cisco 2911/K9 — это устройство нового поколения, относящееся к семейству ISR G2. Модель позволяет создавать безопасное широкополосное подключение к сети, производить передачу мультимедийных данных, видео, осуществлять беспроводную связь, а также применять множество дополнительных функций при минимальном уровне затрат на приобретение и содержание. Маршрутизатор Cisco 2911/K9 со встроенными сетевыми сервисами на 3 порта идеально подходит для создания сетей малых и средних организаций и предприятий. Модель имеет 512 Мб встроенной и 526 Мб флеш-памяти, возможность поддержки PoE на портах 10/100/1000 Ethernet, а также поддержку протокола VPN и туннелирования для осуществления надежной и безопасной передачи при построении удаленных VPN сетей. Маршрутизатор Cisco 2811 — маршрутизатор с интеграцией сервисов, обеспечивающий все потребности небольших офисов и филиалов (до 36 рабочих мест) в современных коммуникациях.

Может выполнять функции:

- маршрутизатора доступа и маршрутизатора локальной сети;
- центра IP-телефонии и голосовой почты (в вариантах Voice bundle и Voice Security bundle);
- интегрированного решения для обеспечения безопасности (в вариантах Security bundle и Voice Security bundle);
- межсетевого экрана;
- систему предотвращения вторжений;
- шифрования и создания VPN-туннелей;
- система Cisco NAC и фильтрация по Uniform Resource Locator (URL);

Коммутатор Cisco Catalyst 2960 — линейка ведущих продуктов среди коммутаторов второго уровня. Эти коммутаторы позволяют снизить совокупную стоимость владения при развертываниях с использованием IP- телефонов Cisco, точек доступа к беспроводной сети (WLAN) Cisco Aironet или других устройств, совместимых с IEEE 802.3af. Технология PoE позволяет устранить потребность в обеспечении питания устройств с поддержкой PoE от сети электропитания и исключить затраты на дополнительные электрические кабели, которые в противном случае необходимы при развертываниях IP-телефонов и сетей WLAN. Их использование позволяет упростить эксплуатацию ИТ-инфраструктуры, повысить

уровень безопасности бизнес-процессов, обеспечить устойчивую работу сети, а также предоставить пользователям возможность работы в «сетях без границ». Каждый спаренный порт каскадирования состоит из одного порта Ethernet RJ-45 и одного порта GE для подключения SFP, при этом активен может быть только один из двух портов, возможность подключения настольных компьютеров к 24 портам Fast Ethernet, широкий набор программных средств, обеспечивающих простоту эксплуатации, защиту бизнес-процессов, устойчивость и работу в «сетях без границ». Межсетевой экран Cisco ASA 5505 — современное многофункциональное устройство для защиты локальных сетей от внешних атак и вторжений. Основная функция межсетевого экрана Cisco (файрвол) — защита сети от вторжений, вирусов, спама, шпионских программ, контентная фильтрация трафика пользователей. Построенные на базе аппаратной платформы сетевые

### **Практическая работа 3: Исследование процессов создания бюджетов пользователей в ОС Windows NT и Unix**

**Цель работы и содержание:** освоить процессы создания бюджетов пользователей в ОС Windows NT и Unix

#### **Теоретическая часть**

Организационные требования к системе защиты предусматривают реализацию совокупности административных и процедурных мероприятий. Требования по обеспечению сохранности должны выполняться прежде всего на административном уровне. Организационные мероприятия, проводимые с целью повышения эффективности защиты информации, должны предусматривать следующие процедуры:

- ограничение несопровождаемого доступа к вычислительной системе (регистрация и сопровождение посетителей);
- осуществление контроля за изменением в системе программного обеспечения;
- выполнение тестирования и верификации изменений в системе программного обеспечения и программах защиты;
- организацию и поддержку взаимного контроля за выполнением правил защиты данных;
- ограничение привилегии персонала, обслуживающего ИС;
- осуществление записи протокола о доступе к системе; ■ гарантию компетентности обслуживающего персонала;

- разработку последовательного подхода к обеспечению сохранности информации для всей организации;
  - организацию четкой работы службы ленточной и дисковой библиотек;
  - комплектование основного персонала на базе интегральных оценок и твердых знаний;
  - организацию системы обучения и повышения квалификации обслуживающего персонала.
- С точки зрения обеспечения доступа к ИС необходимо выполнить следующие процедурные мероприятия:
  - разработать и утвердить письменные инструкции на загрузку и остановку работы операционной системы;
  - контролировать использование магнитных лент, дисков, карт, листингов, порядок изменения программного обеспечения и доведение этих изменений до пользователя;

- разработать процедуру восстановления системы при отказах; • установить политику ограничений при разрешенных визитах в вычислительный центр и определить объем выдаваемой информации;
- разработать систему протоколирования использования ЭВМ, ввода данных и вывода результатов;
- обеспечить проведение периодической чистки архивов и хранилищ носителей информации для исключения и ликвидации неиспользуемых; • поддерживать документацию вычислительного центра в соответствии с установленными стандартами. Требования к подсистемам защиты информации (400) В общем случае СЗИ целесообразно условно разделить на подсистемы:
- управления доступом к ресурсам ИС (включает также функции управления системой защиты в целом);
- регистрации и учета действий пользователей (процессов);
- криптографическую;
- обеспечения целостности информационных ресурсов и конфигурации ИС. Для каждой из них определяются требования в виде:
  - перечня обеспечиваемых подсистемой функций защиты;
  - основных характеристик этих функций;
  - перечня средств, реализующих эти функции. Подсистема управления доступом должна обеспечивать:
    - идентификацию, аутентификацию и контроль за доступом пользователей (процессов) к системе, терминалам, узлам сети, каналам связи, внешним устройствам, программам, каталогам, файлам, записям и т.д.;
    - управление потоками информации;
    - очистку освобождаемых областей оперативной памяти и внешних накопителей.
- Подсистема регистрации и учета выполняет:
  - регистрацию и учет: доступа в ИС, выдачи выходных документов, запуска программ и процессов, доступа к защищаемым файлам; передачу данных по линиям
    - и каналам связи;
  - регистрацию изменения полномочий доступа, создание объектов доступа, подлежащих защите;
  - учет носителей информации;
- оповещение о попытках нарушения защиты. Криптографическая подсистема предусматривает:
  - шифрование конфиденциальной информации.
  - шифрование информации, принадлежащей разным субъектам доступа (группам субъектов), с использованием разных ключей.
  - использование аттестованных (сертифицированных) криптографических средств.
- Подсистема обеспечения целостности осуществляет:
  - обеспечение целостности программных средств и обрабатываемой информации,
  - физическую охрану средств вычислительной техники и носителей информации,
  - наличие администратора (службы) защиты информации в ИС,
  - периодическое тестирование СЗИ,
  - наличие средств восстановления СЗИ,
  - использование сертифицированных средств защиты,
  - контроль за целостностью:
    - программных средств защиты информации при загрузке операционной среды,
    - операционной среды перед выполнением процессов,
    - функционального ПО и данных,
    - конфигурации ИС,
    - оперативное восстановление функций СЗИ после сбоев,

- тестирование средств защиты информации,
- обнаружение и блокирование распространения вирусов,
- резервное копирование программного обеспечения и данных,
- контроль доступа к СВТ, дающий уверенность в том, что только авторизованный пользователь использует имеющиеся рабочие программы и информацию,
- контроль действий с персональной авторизацией, запрещающий операции, которые делают операционную среду уязвимой,
- защиту программного обеспечения, исключающую повреждение установленных программ,
- использование только лицензионного программного продукта с целью обеспечения защиты от встроенных модулей разрушения информационной среды и дискредитации систем защиты;
- защиту коммуникаций для обеспечения недоступности передаваемой информации. Требования к техническому обеспечению В этой группе формулируются требования к таким параметрам:
  - месту применения средств защиты;
  - способам их использования (например, реализация требований по защищенности должна достигаться без применения экранирования помещений, активные средства могут применяться только для защиты информации главного сервера и т.п.);
  - размерам контролируемой зоны безопасности информации;
  - требуемой величине показателей защищенности, учитывающей реальную обстановку на объектах ИС;
  - применению способов, методов и средств достижения необходимых показателей защищенности.
- проведению специсследования оборудования и технических средств, целью которого является измерение показателей ЭМИ;
- проведению спецпроверки технических объектов ИС, целью которой является
  - выявление специальных электронных (закладных) устройств. Требования к программному обеспечению Программные средства защиты информации должны обеспечивать контроль доступа, безопасность и целостность данных и защиту самой системы защиты. Для этого необходимо выполнить следующие условия:
    - объекты защиты должны идентифицироваться в явном виде при использовании паролей, пропусков и идентификации по голосу;
    - система контроля доступа должна быть достаточно гибкой для обеспечения многообразных ограничений и различных наборов объектов;
    - каждый доступ к файлу данных или устройству должен прослеживаться через систему контроля доступа для того, чтобы фиксировать и документировать любое обращение. Безопасность данных может обеспечиваться следующей системой мероприятий:
      - объекты данных идентифицируются и снабжаются информацией службы безопасности. Целесообразно эту информацию размещать не в отдельном каталоге, а вместе с информацией, имеющей метки;
      - кодовые слова защиты размещаются внутри файлов, что в значительной мере повышает эффективность защиты;
      - доступ к данным целесообразен с помощью косвенных ссылок, например списка пользователей, допущенных владельцем файла к размещенным в нем данным;
      - данные и программы могут преобразовываться (кодироваться) внутренним способом для хранения.
  - Система защиты информации должна быть защищена от воздействия окружающей среды. С этой целью выполняется следующая совокупность мероприятий:
    - информация по отрицательным запросам не выдается;
    - повторные попытки доступа после неудачных обращений должны иметь предел;

- при уменьшении конфигурации системы или при ее тестировании функции защиты сохраняются;
- никакие изменения таблиц безопасности, кроме изменения со специального устройства или пульта управления, не разрешаются. Требования по применению способов, методов и средств защиты Рекомендуются применение следующих способов, методов и средств, которые предполагают использование:
  - интерфейсов с передачей сигналов в виде последовательного кода и в режиме многократных повторений;
  - мультиплексных режимов обработки информации, а также СВТ и системного обеспечения, базирующихся на многоразрядных платформах, интерфейсов с передачей сигналов в виде многоразрядного параллельного кода;
  - рациональных способов монтажа, при которых обеспечивается минимальная протяженность электрических связей и коммуникаций;
  - технических средств, в состав которых входят устойчивые к самовозбуждению схемы, развязывающие и фильтрующие элементы, комплектующие с низкими уровнями ЭМИ;
  - сетевых фильтров для блокирования утечки информации по цепям электропитания, а также линейных (высокочастотных) фильтров для блокирования утечки информации по линиям связи;
  - технических средств в защищенном исполнении;
  - средств пространственного и линейного “зашумления”;
  - средств локального либо общего экранирования;
  - способов оптимального размещения технических средств с целью минимизации контролируемой зоны безопасности информации. Требования к документированию Можно выделить три группы требований к документированию системы защиты информации. Это протоколирование, тестирование программ и обработка угроз. При разработке системы протоколирования следует учитывать следующие специфические требования:
    - необходимость записей всех движений, защищаемых данных;
    - возможность воссоздания при необходимости ретроспективы использования защищаемого объекта, для реализации которой обеспечивается запоминание состояний программы и окружающей среды;
    - накопление статистики по протоколам использования информации в системе. Существенной особенностью тестирования программ системы защиты информации должно быть наличие специальной программы генерации ложных адресов, несанкционированных попыток доступа к данным, моделирования сбойных ситуаций и других специфических свойств. При тестировании системы защиты информации необходимо также обратить внимание на тщательную проверку таблиц безопасности, системы паролей и программ доступа.

**Аппаратура и материалы** Для выполнения практической работы необходим персональный компьютер со следующими характеристиками: процессор Intel Pentium-сов – месстимый с тактовой частотой 800 МГц и выше, оперативная память – не менее 512 Мбайт, свободное дисковое пространство – не менее 2 Гбайт, устройство для чтения компакт-дисков, монитор типа Super VGA (число цветов от 256) с диагональю не менее 15<sup>2</sup>. Программное обеспечение – операционная система Windows XP с пакетом обновлений SP2 и выше, ассемблер MASM версии 6.14 и выше, дизассемблер IDA Pro Free версии 4.9 и выше.

Указания по технике безопасности Техника безопасности при выполнении практической работы совпадает с общепринятой для пользователей персональных компьютеров, самостоятельно не производить ремонт персонального компьютера, установку и удаление программного обеспечения; в случае неисправности персонального компьютера сообщить об этом обслуживающему персоналу лаборатории (оператору, администратору); соблюдать правила техники безопасности при работе с электрооборудованием; не касаться электрических розеток металли-24 ческими предметами; рабочее место пользователя

персонального компьютера должно содержаться в чистоте; не разрешается возле персонального компьютера принимать пищу, напитки.

#### **Практическая работа 4: Исследование методов интеграции сетей Microsoft и Unix с использованием сервера Samba**

**Цель работы и содержание:** освоить методы интегрирования сетей Microsoft и Unix с использованием сервера Samba

**Теоретическая часть** При избирательном разграничении доступа определенные операции над конкретным ресурсом запрещаются или разрешаются субъектам или группам субъектов. Большинство ОС реализуют именно избирательное разграничение доступа (discretionary access control).

Полномочное разграничение доступа заключается в том, что все объекты могут иметь уровни секретности, а все субъекты делятся на группы, образующие иерархию в соответствии с уровнем допуска к информации. Иногда эту модель называют моделью многоуровневой безопасности, предназначенной для хранения секретов. Избирательное разграничение доступа Система правил избирательного разграничения доступа формулируется следующим образом.

1. Для любого объекта ОС существует владелец.
2. Владелец объекта может произвольно ограничивать доступ других субъектов к данному объекту.
3. Для каждой тройки субъект—объект—метод возможность доступа определена однозначно.
4. Существует хотя бы один привилегированный пользователь (администратор), имеющий возможность обратиться к любому объекту по любому методу доступа. Привилегированный пользователь не может игнорировать разграничение доступа к объектам. Например, в Windows NT администратор для обращения к чужому объекту (принадлежащему другому субъекту) должен сначала объявить себя владельцем этого объекта, использовав привилегию администратора объявлять себя владельцем любого объекта, затем дать себе необходимые права и только после этого может обратиться к объекту. Последнее требование введено для реализации механизма удаления потенциально недоступных объектов. При создании объекта его владельцем назначается субъект, создавший данный объект. В дальнейшем субъект, обладающий необходимыми правами, может назначить объекту нового владельца. При этом субъект, изменяющий владельца объекта, может назначить новым владельцем объекта только себя. Такое ограничение вводится для того, чтобы владелец объекта не мог отдать «владение» объектом другому

субъекту и тем самым снять с себя ответственность за некорректные действия с объектом. Для определения прав доступа субъектов к объектам при избирательном разграничении доступа используются такие понятия, как матрица доступа и домен безопасности. С концептуальной точки зрения текущее состояние прав доступа при избирательном разграничении доступа описывается матрицей, в строках которой перечислены субъекты доступа, в столбцах — объекты доступа, а в ячейках — операции, которые субъект может выполнить над объектом. Домен безопасности (protection domain) определяет набор объектов и типов операций, которые могут производиться над каждым объектом ОС. Возможность выполнять операции над объектом есть право доступа, каждое из которых есть упорядоченная пара. Таким образом, домен есть набор прав доступа. Например, если домен  $D$  имеет право доступа  $p$ , это означает, что процесс, выполняемый в домене  $D$ , может читать или писать в файл  $F$ , но не может выполнять других операций над этим объектом. Связь конкретных субъектов, функционирующих в ОС, может быть организована следующим образом:

- каждый пользователь может быть доменом. В этом случае набор объектов, к которым может быть организован доступ, зависит от идентификации пользователя;
- каждый процесс может быть доменом. В этом случае набор доступных объектов определяется идентификацией процесса;
- каждая процедура может быть доменом. В этом случае набор доступных объектов соответствует локальным переменным, определенным внутри процедуры. Заметим, что, когда процедура выполнена, происходит смена домена. Модель безопасности, специфицированная выше (см. рис. 8.1), имеет вид матрицы и называется матрицей доступа. Столбцы этой матрицы представляют собой объекты, строки — субъекты. В каждой ячейке матрицы хранится совокупность прав доступа, предоставленных данному субъекту на данный объект. Поскольку реальная матрица доступа очень велика (типичный объем для современной ОС составляет несколько десятков мегабайтов), матрицу доступа никогда не хранят в системе в явном виде. В общем случае эта матрица будет разреженной, т. е. большинство ее клеток будут пустыми. Матрицу доступа можно разложить по столбцам, в результате чего получаются списки прав доступа ACL (access control list). В результате разложения матрицы по строкам получаются мандаты возможностей (capability list, или capability tickets). Список прав доступа ACL. Каждая колонка в матрице может быть реализована как список доступа для одного объекта. Очевидно, что пустые клетки могут не учитываться. В результате для каждого объекта имеем список упорядоченных пар, который определяет все домены с непустыми наборами прав для данного объекта. Элементами

списка прав доступа ACL могут быть процессы, пользователи или группы пользователей. При реализации широко применяется предоставление доступа по умолчанию для пользователей, права которых не указаны. Например, в ОС Unix все субъекты-пользователи разделены на три группы (владелец, группа и остальные), и для членов каждой группы контролируются операции чтения, записи и исполнения (rwx). В итоге имеем ACL — 9-битный код, который является атрибутом разнообразных объектов Unix. Мандаты возможностей. Как отмечалось выше, если матрицу доступа хранить по строкам, т. е. если каждый субъект хранит список объектов и для каждого объекта — список допустимых операций, то такой способ хранения называется «мандаты возможностей» или «перечни возможностей» (capability list). Каждый пользователь обладает несколькими мандатами и может иметь право передавать их другим. Мандаты могут быть рассеяны по системе и вследствие этого представлять большую угрозу для безопасности, чем списки контроля доступа. Их хранение должно быть тщательно продумано. Избирательное разграничение доступа — наиболее распространенный способ разграничения доступа. Это обусловлено сравнительной простотой его реализации и необременительностью правил такого разграничения доступа для пользователей. Главное достоинство избирательного разграничения доступа — гибкость; основные недостатки — рассредоточенность управления и сложность централизованного контроля. Вместе с тем, защищенность ОС, подсистема защиты которой реализует только избирательное разграничение доступа, в некоторых случаях может оказаться недостаточной. В частности, в США запрещено хранить информацию, содержащую государственную тайну, в компьютерных системах, поддерживающих только избирательное разграничение доступа. Расширением модели избирательного разграничения доступа является изолированная (или замкнутая) программная среда. При использовании изолированной программной среды права субъекта на доступ к объекту определяются не только правами и привилегиями субъекта, но и процессом, с помощью которого субъект обращается к объекту. Можно, например, разрешить обращаться к файлам с расширением .doc только программам Word, Word Viewer и WPview. Изолированная программная среда существенно повышает защищенность операционной системы от разрушающих программных воздействий, включая программные закладки и компьютерные вирусы. Кроме того, при использовании данной модели повышается защищенность целостности данных, хранящихся в системе. Полномочное разграничение доступа с контролем информационных потоков Полномочное, или мандатное, разграничение доступа (mandatory access control) обычно применяется в совокупности с избирательным разграничением доступа. Рассмотрим именно такой случай.

Правила разграничения доступа в данной модели формулируются следующим образом.

1. Для любого объекта ОС существует владелец.
2. Владелец объекта может произвольно ограничивать доступ других субъектов к данному объекту.
3. Для каждой четверки субъект—объект—метод—процесс возможность доступа определена однозначно в каждый момент времени. При изменении состояния процесса со временем возможность предоставления доступа также может измениться. Вместе с тем, в каждый момент времени возможность доступа определена однозначно. Поскольку права процесса на доступ к объекту меняются с течением времени, они должны проверяться не только при открытии объекта, но и перед выполнением над объектом таких операций, как чтение и запись.
4. Существует хотя бы один привилегированный пользователь (администратор), имеющий возможность удалить любой объект.
5. В множестве объектов выделяется множество объектов полномочного разграничения доступа. Каждый объект полномочного разграничения доступа имеет гриф секретности. Чем выше числовое значение грифа секретности, тем секретнее объект. Нулевое значение грифа секретности означает, что объект несекретен. Если объект не является объектом полномочного разграничения доступа или если объект несекретен, администратор может обратиться к нему по любому методу, как и в предыдущей модели разграничения доступа.
6. Каждый субъект доступа имеет уровень допуска. Чем выше числовое значение уровня допуска, тем больший допуск имеет субъект. Нулевое значение уровня допуска означает, что субъект не имеет допуска. Обычно ненулевое значение допуска назначается только субъектам-пользователям и не назначается субъектам, от имени которых выполняются системные процессы.
7. Доступ субъекта к объекту должен быть запрещен независимо от состояния матрицы доступа, если:
  - объект является объектом полномочного разграничения доступа;
  - гриф секретности объекта строго выше уровня допуска субъекта, обращающегося к нему;
  - субъект открывает объект в режиме, допускающем чтение информации. Это правило называют правилом NRU (Not Read Up — не читать выше).
8. Каждый процесс ОС имеет уровень конфиденциальности, равный максимуму из грифов секретности объектов, открытых процессом на протяжении своего существования. Уровень конфиденциальности фактически представляет собой гриф секретности информации, хранящейся в оперативной памяти процесса.

9. Доступ субъекта к объекту должен быть запрещен независимо от состояния матрицы доступа, если:

- объект является объектом полномочного разграничения доступа;
- гриф секретности объекта строго ниже уровня конфиденциальности процесса, обращаемого к нему;
- субъект собирается записывать в объект информацию, Это правило предотвращает утечку секретной информации; его называют правилом NWD (Not Write Down — не записывать ниже).

10. Понизить гриф секретности объекта полномочного разграничения доступа может только субъект, который:

- имеет доступ к объекту согласно правилу 7;
- обладает специальной привилегией, позволяющей ему понижать грифы секретности объектов.

**Аппаратура и материалы** Для выполнения практической работы необходим персональный компьютер со следующими характеристиками: процессор Intel Pentium-совместимый с тактовой частотой 800 МГц и выше, оперативная память – не менее 512 Мбайт, свободное дисковое пространство – не менее 2 Гбайт, устройство для чтения компакт-дисков, монитор типа Super VGA (число цветов от 256) с диагональю не менее 15".  
**Программное обеспечение** – операционная система Windows XP с пакетом обновлений SP2 и выше, ассемблер MASM версии 6.14 и выше, дизассемблер IDA Pro Free версии 4.9 и выше. Указания по технике безопасности Техника безопасности при выполнении практической работы совпадает с общепринятой для пользователей персональных компьютеров, самостоятельно не производить ремонт персонального компьютера, установку и удаление программного обеспечения; в случае неисправности персонального компьютера сообщить об этом обслуживающему персоналу лаборатории (оператору, администратору); соблюдать правила техники безопасности при работе с электрооборудованием; не касаться электрических розеток металлическими предметами; рабочее место пользователя персонального компьютера должно содержаться в чистоте; не разрешается возле персонального компьютера принимать пищу, напитки.

## **Практическая работа 5: Изучение базовых команд ОС Linux**

**Цель:** Первичное знакомство с командным интерпретатором. Изучение базовых команд операционной системы Linux.

### **Теоретическая часть:**

Среди всех элементов операционной системы Linux самым важным, является командная строка (Терминал). Оболочка во многом определяет богатые возможности и гибкость операционной системы Linux. С помощью командной строки можно выполнять действия, которые были бы немыслимы при работе с графическим пользовательским интерфейсом. Независимо от того, KDE или GNOME, оказывается, что многие действия гораздо быстрее и эффективнее выполнить, пользуясь только командной строкой. Освоение Linux стоит начинать с изучения средств командной оболочки.

#### **Файлы и ничего кроме файлов**

Все, с чем Вы встретитесь в операционной системе Linux, - это файлы. Абсолютно все! Очевидно, что текстовый документ - это файл. Изображения, аудиоданные в формате MP3 и видеофрагменты - это несомненно файлы. Каталоги - это тоже файлы, содержащие информацию о других файлах. Дисковые устройства - это большие файлы. Сетевые соединения тоже файлы. Даже исполняемый процесс - это файл. С точки зрения операционной системы Linux файл представляет собой поток битов или байтов. Система не интересуется тем, что означает каждый байт. Это забота конкретных программ, выполняющихся в операционной системе Linux. Для операционной системы Linux и документ, и сетевое соединение всего лишь файлы. Как обрабатывать текстовый документ, знает редактор, а сетевое приложение умеет работать с сетевым соединением.

В отличие от Windows и MacOS в операционной системе Linux имена файлов чувствительны к регистру символов. В частности, Вы можете встретить в одном каталоге все три файла которые приведены ниже в качестве примера:

Sit.txt  
sIt.txt  
SIT.txt

С точки зрения файловой операционной системы Linux - это различные имена файлов. Если вы попытаетесь создать файлы с этими же именами в Windows или MacOS, то вероятнее всего попытка увенчается провалом, и система предложит Вам выбрать другое имя для файла.

Чувствительность к регистру символов также означает, что при вводе команд они должны в точности совпадать с именами файлов, поддерживающих их. Так, например, удаляя файл с помощью команды `rm`, нельзя вводить `RM`, `Rm` или `rM`. Надо также следить за написанием имен, задаваемых в качестве параметров. Если вы захотите удалить файл «SIT.txt», а укажете имя `Sit.txt`, вы лишитесь совсем не того файла, с которым предполагали расстаться.

## ⓘ Предупреждение

Список специальных символов которые не рекомендуется использовать в названиях файлов.

```
/ - Нельзя использовать ни при каких обстоятельствах
\ - Должен быть предварен таким же символом. Применять не рекомендуется
- - Нельзя использовать в начале имени файла или каталога
[] - Каждый из этих символов должен быть предварен обратной косой чертой. Применять не рекомендуется
{} - Каждый из этих символов должен быть предварен обратной косой чертой. Применять не рекомендуется
* - Должен быть предварен обратной косой чертой. Применять не рекомендуется
? - Должен быть предварен обратной косой чертой. Применять не рекомендуется
' - Должен быть предварен обратной косой чертой. Применять не рекомендуется
" - Должен быть предварен обратной косой чертой. Применять не рекомендуется
```

## Групповые операции:

Предположим, что в одном из каталогов на вашем компьютере содержатся сто файлов с изображениями и два текстовых файла. Ваша задача удалить все файлы с изображениями за исключением двух текстовых файлов. Удалять файлы по одному - это утомительное занятие. В операционных системах Linux для автоматизации данного процесса можно применять символы групповых операций. Групповые операции задаются посредством звездочки (\*), знака вопроса (?) и квадратных скобок ([ ]).

## Пример использования групповых операций:

Групповая операция с применением « \* » - отмечает любое (в том числе нулевое) количество любых символов.

```
rm sit1*.*      Удаляются файлы : sit1.txt, sit1.jpg, sit11.jpg, sit123123.txt
rm sit*.jpg     Удаляются файлы : sit1.jpg, sit11.jpg
rm *txt        Удаляются файлы : sit1.txt, sit123123.txt
rm sit*        Удаляются файлы : sit1.txt, sit1.jpg, sit11.jpg, sit123123.txt
rm *           Удаляются все файлы в каталоге
```

Групповая операция с применением « ? ». Символ « ? » - соответствует одному произвольному символу.

```
rm sit1?.jpg   Удалится файл : sit11.jpg, но не sit1.txt, sit1.jpg, sit123123.txt
rm sit?.jpg    Удалится файл : sit1.jpg, но не sit1.txt, sit11.jpg, sit123123.txt
rm sit?.*     Удаляются файлы : sit1.txt, sit1.jpg, но не sit11.jpg, sit123123.txt
```

Групповая операция с применением « [] ». Квадратные скобки позволяют задавать один символ из набора или символ, принадлежащий определенному диапазону.

```
rm sit[0-1].txt      Удалится файл : sit1.txt, но не sit1.jpg, sit11.jpg, sit123123.txt
rm sit1[0-2].jpg    Удалится файл : sit11.jpg, но не sit1.txt, sit1.jpg, sit123123.txt
```

## Практическая работа 6 Настройка ОС Linux

### 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 этом режиме использование 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. Системараспространяется 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 монополизации.

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.

### **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.

### **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.

### **Выход из системы**

**exit** - окончание сеанса пользователя.

### **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.

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

### Порядок выполнения работы

1. Ознакомиться с теоретической частью к практической 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.

### 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*?

## Практическая работа 7. Работа с файлами в Linux

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 в этом режиме использование 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. Системараспространяется 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 монополизации.

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

- 6) использование UNIX-сервера, как файл - сервера;
- 7) эмуляция на ПК удаленного терминала (режим TELNET);
- 8) организация системы клиент - сервер (рабочая станция формирует SQL - запросы, сервер их обрабатывает);
- 9) непосредственный обмен файлами между ПК по протоколу FTP;
- 10) 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.

### 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 экранный 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.

**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.

#### **Выход из системы**

**exit** - окончание сеанса пользователя.

#### **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.

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

### Порядок выполнения работы

13. Ознакомиться с теоретической частью к практической работе.
14. Зарегистрироваться в системе LINUX.
15. Определить день недели, в который Вы родились.
16. Получить подробную информацию обо всех активных процессах.
17. Используя редактор VI (см. приложение), создать два текстовых файла (с расширением TXT) и командой CAT просмотреть их на экране.
18. Получить информацию о работающих пользователях, подсчитать их количество и запомнить в файле.
19. Объединить текстовые файлы в единый файл и посмотреть его на экране.
20. Посмотреть приоритет своего процесса и уменьшить скорость его выполнения за счет повышения номера приоритета.
21. Используя редактор VI, написать программу на языке СИ и запустить ее в трансляцию в фоновом режиме.
22. Показать преподавателю исходный текст программы на языке СИ, текстовый файл, файл с сохранением количества пользователей.
23. Продемонстрировать выполнение СИ - программы.
24. Удалить свои файлы и выйти из системы.

## Практическая работа 8 Работа с файлами в Linux

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.

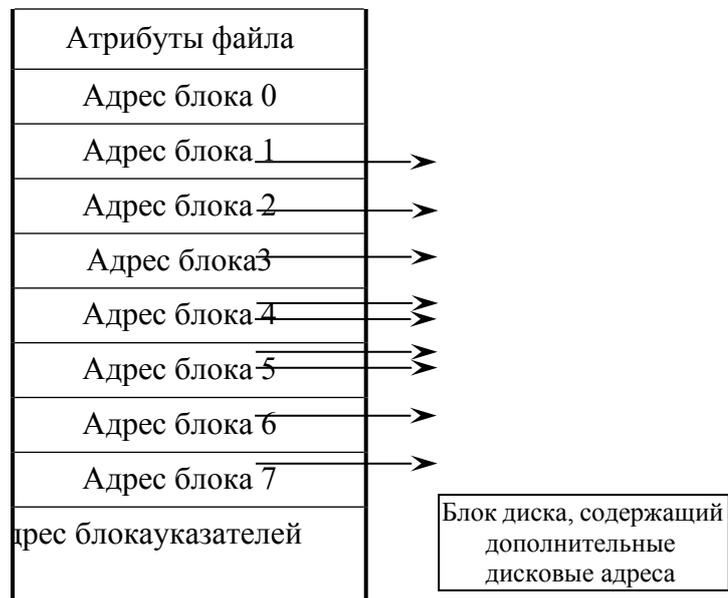


Рис. 5. Пример *i*-того узла

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.). Каталоговая 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 полным именем файла. Если обращение к файлу начинается с символа

«/», то считается, что указано полное имя файла и его поиск начинается с корневого каталога, в любом другом случае поиск файла начинается с текущего каталога. У любого файла может быть несколько имен. Фактически имя файла является ссылкой на файл, специфицированный номером описателя.

Номер i-узла	Имя файла
--------------	-----------

Рис.6. Каталоговая запись файловой системы UNIX V7

I-узлы системы UNIX (рис.7.) содержат атрибуты: размер файла, три указателя времени (создания, последнего доступа и последнего изменения), идентификатор владельца, номер группы, информацию о защите и счетчик каталоговых записей, указывающих на данный i-узел. При добавлении новой связи к i-узлу счетчик в i-узле увеличивается на единицу. При удалении связи счетчик в i-узле уменьшается на единицу. Когда значение счетчика достигает нуля, i-узел освобождается, а блоки диска, которые занимал файл, возвращаются в список свободных блоков. Для учета дисковых блоков файла используется обобщение схемы, позволяющее работать с очень большими файлами. Первые 10 дисковых адресов хранятся в самом i-узле. Для небольших файлов необходимая информация содержится прямо в i-узле, считываемом с диска при открытии файла. Для файлов большего размера один из адресов в i-узле представляет собой адрес блока диска, называемого одинарным косвенным блоком. Этот блок содержит дополнительные дисковые адреса. Если и этого недостаточно используется другой адрес в i-узле, называемый двойным косвенным блоком и содержащий адрес блока, в котором хранятся адреса однократных косвенных блоков. Если и этого мало, используется тройной косвенный блок.

При открытии файла файловая система по имени файла находит его блоки на диске. Вначале файловая система открывает корневой каталог, i-узел которого располагается в фиксированном месте диска. По этому i-узлу система определяет положение корневого каталога, который может находиться в любом месте диска, например, в блоке 1. Затем файловая система считывает корневой каталог и ищет в нем первый компонент пути, чтобы определить номер i-узла файла. По этому i-узлу файловая система находит следующий каталог и находит в нем следующий компонент и т.д. Относительные пути файлов обрабатываются так же как и абсолютные, с той 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:

- 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 (номерами 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.

## Практическая работа 9

### *Работа с реестром Windows*

#### **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.

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\SystemSystem, 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.

HKKEY\_USERS      Данный раздел содержит все активные загруженные профили пользователей компьютера.

**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.

## Практическая работа 10

### *Работа в командной строке*

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 спомощью 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.

#### ***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?

## Практическая работа 11

### Пакетные файлы

#### 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 в символы процентов (%), например **Set MyPath=%CD%**. Символы процентов указывают на то, что командный интерпретатор должен обратиться к значению переменной безпосимвольного 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		%~nx1 замена %1 именем файла и расширением
3	%~dp\$PATH:1	поиск в каталогах, перечисленных в переменной средыPATH, 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 практических 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 спрограмным 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 «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 практических 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 практической 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. Практическая работа выполняется на виртуальной машине в среде ОС Windows XP.

## **5. Учебно-методическое и информационное обеспечение дисциплины**

### **5.1. Перечень основной и дополнительной литературы, необходимой для освоения дисциплины**

#### **5.1.1. Перечень основной литературы:**

- 4 Воронцова, В. Л. Лекции по системному и прикладному программному обеспечению: Учебное пособие / В. Л. Воронцова, И. А. Журавлева; Ставроп. гос. ун-т. – Ставрополь: СГУ, 2004. – 208 с. – Библиогр: с. 205-206
- 5 Защита информации в операционных системах: лабораторные работы: учеб.- метод. пособие / М. В. Романкова / сост.: И. В. Зайцева / Федеральное агентство по образованию Ставроп. гос. ун-т. – Ставрополь : Изд-во СГУ, 2008. – 99 с. – Библиогр.: с. 96-97
- 6 Программно-аппаратные средства обеспечения информационной безопасности: учеб. пособие / сост. В. И. Петренко, Н. В. Куклева ; Федер. агентство по образованию, Ставроп. гос. ун-т, Ч. 2, Защита в операционных системах. – Ставрополь: Изд-во СГУ, 2007. – 154 с. – Библиогр: с. 154

#### **5.1.2. Перечень дополнительной литературы:**

- 3 Безручко, В. Т. Практикум по курсу "Информатика". Работа в Windows, Word, Excel [Учеб. пособие для вузов\*]. – М.: Финансы и статистика, 2004. – 272с.: ил. – Библиогр.: 265. – ISBN 5-279-02436-8
- 4 Гриценко, Ю.Б. Операционные системы: в 2-х ч. / Ю.Б. Гриценко; Федеральное агентст по образованию, Томский межвузовский центр дистанционного образования (ТУСУРКафедра автоматизации обработки информации (АОИ). Томск: Томский государственный университет систем управления и радиоэлектроники, 2009. – Ч. 2. – 2 с.– Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=208655>

### **5.2. Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине (модулю)**

1. Методические рекомендации по выполнению практических работ по дисциплине «Защищенные локальные вычислительные сети».

2. Методические рекомендации по организации самостоятельной работы студентов по дисциплине «Защищенные локальные вычислительные сети».

### **5.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)**

1. <http://el.ncfu.ru/> – система управления обучением ФГАОУ ВО СКФУ. Дистанционная поддержка дисциплины «Цифровая грамотность и обработка данных»
2. <http://www.un.org> - Сайт ООН Информационно-коммуникационные технологии
3. <http://www.intuit.ru> – Интернет-Университет Компьютерных технологий.