Индивидуальный предприниматель
Апалин Владимир Александрович

УТВЕРЖДАЮ
Апалин В.А.



22.11.2024 г.

Дополнительная общеобразовательная программа – дополнительная общеразвивающая программа «Системный администратор Linux»

Возраст обучающихся: от 18 лет
Срок реализации: 81 ак.ч.

Разработчик:
Апалин В.А.

Санкт-Петербург
2024 г.

I. Общая характеристика программы

Введение. Дополнительная общеразвивающая программа «Системный администратор Linux» (далее – «Программа») предназначена для лиц, сферу интересов которых составляют: системные администрирование и смежные области, связанные с этой сферой. Усвоение слушателями данной Программы позволит использовать полученные знания в области системного администрирования. Программа предполагает:

Нормативно-правовая основа реализации Программы. Программа основана на следующих положениях законодательных и нормативных актов Российской Федерации: Федеральный закон от 29.12.2012 №273-ФЗ «Об образовании в Российской Федерации».

Актуальность Программы определяется ростом использования Linux, развитием облачных технологий, постоянным изменением в мире информационных технологий и расширением баз знаний, применение которых вызывает затруднения у системных администраторов.

Новизна Программы и ее отличительная особенность заключается в формировании у обучающихся навыков, необходимых для успешного применения современных технологий. Программа «Системный администратор Linux» предлагает современное обучение с акцентом на актуальные технологии. Она также включает аспекты администрирования Linux и адаптирована под требования рынка труда, что делает выпускников востребованными специалистами.

Цель программы — формирование и развитие творческих способностей обучающихся, удовлетворение их индивидуальных потребностей в интеллектуальном, нравственном и физическом совершенствовании, формирование культуры здорового и безопасного образа жизни, укрепление здоровья, а также на организацию их свободного времени.

Данная цель реализуется решением следующих **задач:**

- адаптация к жизни в обществе,
- профессиональная ориентация,
- выявление и поддержка обучающихся, проявивших выдающиеся способности

II. Условия реализации Программы

Организационно-педагогические условия Программы

В реализации Программы участвуют обучающиеся от 18 лет.

Кадровое обеспечение: Учебный процесс обеспечивают высококвалифицированные специалисты в сфере IT и системного администрирования.

Материально-техническое обеспечение. Для реализации образовательной программы используется следующее материально-техническое обеспечение:

1. Персональный компьютер (ноутбук/моноблок/стационарный системный блок)

Минимальные технические характеристики:
Операционная система: Microsoft Windows 7/8/8.1/10/11, разрядность: x86 или x64
ЦП: Процессор Intel или AMD с частотой не менее 1,5 ГГц
ОЗУ: 2 ГБ RAM
Жесткий диск/SSD-накопитель: 120 Гб
Графический адаптер: встроенная графика Intel/AMD или любой дискретный графический адаптер

2. Периферийное оборудование:
Монитор: диагональ не менее 19”, разрешение не менее 1024x768
Мышь и клавиатура (комплект): Logitech Desktop MK120 или аналоги.

Доступ в сеть Интернет: беспроводной доступ на скорости не менее 30 Мбит/с

Сроки реализации и объем Программы: Срок реализации — 10 недель. Объем Программы — 81 академический час.

Документ о квалификации. По окончании курса слушателям выдается документ об обучении (сертификат о прохождении обучения).

Форма и режим занятий.

Форма обучения: настоящая образовательная программа реализуется с применением исключительно электронного обучения, дистанционных образовательных технологий и асинхронного метода обучения на электронной площадке Антитренинги.

Программа состоит из восьми модулей. Каждый модуль включает в себя темы. Каждая тема включает в себя текстовое и графическое изложение материала, промежуточный тест по теме, либо задание, тексты по модулям и итоговый тест по всем темам. При изучении материала по определенной теме слушателям в качестве учебного материала предоставляются учебно-методические, информационные материалы по тематике обучения для более детального усвоения изученного материала.

Доступ к каждому новому модулю (теме) обучения открывается для слушателей Программы в соответствии с календарным учебным графиком Программы согласно расписанию, представленному на образовательной площадке или после выполнения слушателем задания по теме.

III. Планируемые результаты освоения Программы

Планируемые результаты обучения заключаются в том, что обучающийся должен:

знать:

- краткую историю Linux, где применяются операционные системы на базе ядра Linux;
- основную информацию о дистрибутиве Ubuntu;
- основные команды Linux;
- знать основы Bash

уметь:

- работать с терминалом Linux;
- устанавливать Linux на VirtualBox;
- администрировать операционные системы на базе ядра Linux;
- работать с форматами MS Word и Excel в LibreOffice и печатными устройствами

IV. Перечень компетенций Программы

1. Работа с программными продуктами виртуализации
2. Установка и настройка Linux-системы
3. Базовые навыки администрирования ОС на базе ядра Linux
4. Основы программирования на Bash
5. Работа с LibreOffice
6. Работа с печатными устройствами в Linux

V.Содержание Программы

Учебный план Программы

№ п/п	Название модуля	Количество часов			Форма контроля
		Всего	Лекции	Самостоятельная работа	
1.	Историческая справка. Обоснование выбора дистрибутива Ubuntu	5	4	1	тесты
2.	Установка Linux на VirtualBox. Работа с виртуальными машинами	8,5	7	1,5	тесты, практические задания
3.	Первые шаги в Linux. Знакомство с терминалом и пользовательским интерфейсом Ubuntu 24.04	10	8	2	тесты, практические задания
4.	Необходимый минимум. Изучение основных команд, работа с редакторами Nano и Vim	8	6	2	тесты, практические

					задания
5.	Погружение в администрирование Linux	32	25	7	тесты, практические задания
6.	Основы программирования на Bash	10	8	2	тесты, практические задания
7.	Работа с документами форматов Word и Excel в LibreOffice	4,5	3	1,5	тесты, практические задания
8.	Работа с печатными устройствами в Ubuntu 24.04	3	2	1	тесты
	Итого:	**81**	63	18	

Календарный учебный график Программы

№ модуля	**Период обучения (недели)**	**Название раздела**	**Количество часов**	**Форма контроля**
1.	Неделя 1	Историческая справка. Обоснование выбора дистрибутива Ubuntu	5	тесты
2.	Неделя 2	Установка Linux на VirtualBox. Работа с виртуальными машинами	8,5	тесты, практические задания
3.	Неделя 3	Первые шаги в Linux. Знакомство с терминалом и пользовательским интерфейсом Ubuntu 24.04	10	тесты, практические задания
4.	Неделя 4	Необходимый минимум. Изучение основных команд, работа с редакторами Nano и Vim	8	тесты, практические задания
5.	Неделя 5	Погружение в администрирование Linux	8	тесты, практические задания
6.	Неделя 6	Погружение в администрирование Linux	8	тесты, практические задания
7.	Неделя 7	Погружение в администрирование Linux	8	тесты, практические задания
8.	Неделя 8	Погружение в администрирование Linux	8	тесты, практические задания
9.	Неделя 9	Основы программирования на Bash	10	тесты, практические задания
10.	Неделя 10	Работа с документами форматов Word и Excel в LibreOffice	4,5	тесты, практические

				задания
11.	Неделя 10	Работа с печатными устройствами в Ubuntu 24.04	3	тесты

VI. Формы и оценочные критерии результатов

Формы и оценочные критерии текущего контроля

Промежуточная аттестация проводится в форме тестирования. Пример теста приведен в приложении А. Тест состоит из 3 вопросов по итогам каждого урока (и 5 вопросов по итогам каждого модуля). Критерии оценки уровня освоения урока:

- оценки «зачтено» заслуживает обучающийся, ответивший верно на 2 и более из 3 вопросов теста;
- оценки «не зачтено» заслуживает обучающийся, ответивший верно менее, чем на 2 из 3 вопросов теста.

VII. Методическое обеспечение образовательного процесса

Методическое обеспечение образовательного процесса по Программе включает в себя:

1. Авторские разработки на электронной площадке:
   - модульная структура программы;
   - описание отдельных занятий;
   - видеолекции;
   - тесты.
2. Учебно-иллюстративный материал на электронной площадке:
   - презентации по темам.
3. Методические материалы:
   - литература для обучающихся (из электронного библиотечного фонда).

Учебно-методическое и информационное обеспечение

1. List of releases [Электронный ресурс] URL: https://wiki.ubuntu.com/Releases#Release_cadence
2. ArchWiki [Электронный ресурс] URL: https://wiki.archlinux.org
3. DebianWiki [Электронный ресурс] URL: https://wiki.debian.org/
4. Официальный сайт Ubuntu [Электронный ресурс] URL: https://ubuntu.com/
5. Грузин, Н. А. Сравнение Linux дистрибутивов: Arch Linux и Ubuntu / Н. А. Грузин // Modern Science. – 2021. – № 1–1. – С. 444–447. – EDN ZHNMBL.
6. Обзор и анализ операционных систем / Р. П. Христофоров, В. В. Гусев, И. В. Гусев [и др.] // Научное сообщество студентов XXI столетия. Технические науки: Электронный сборник статей по материалам LIX студенческой международной научно-практической конференции. Том 11(58). Часть 2: Ассоциация научных сотрудников "Сибирская академическая книга", 2017. – С. 24–26. – EDN YPWHYU.

Приложение А

Пример теста для промежуточной аттестации

1. Кто такой Tux и какое отношение он имеет к Linux?

- Талисман Linux
- Tux – это прозвище Линуса Торвальдса
- Разработчик первого дистрибутива Linux
- Автор лицензии GPL

2. Как называется организация, занимающаяся поддержкой, продвижением и развитием Linux?

- IBM
- Linux Foundation
- GNU Project
- Microsoft

3. Какой проект послужил основой для создания ядра Linux?

- Unix
- MS-DOS
- BSD
- Minix

Приложение № 1

РАБОЧАЯ ПРОГРАММА МОДУЛЯ 1
(Историческая справка. Обоснование выбора дистрибутива Ubuntu)

По дополнительной общеобразовательной программе –
дополнительной общеразвивающей программе
«Системный администратор Linux»

1. Цель и задачи модуля

Целью освоения модуля «Историческая справка. Обоснование выбора дистрибутива Ubuntu» является приобретение знаний о ключевых этапах развития Linux и понимание причин, почему Ubuntu является популярным выбором среди пользователей. Модуль также направлен на развитие навыков анализа и сравнения дистрибутивов, что поможет сделать обоснованный выбор операционной системы в зависимости от потребностей и целей.

Задачи:

1. Исследовать историческое развитие и ключевые этапы в истории Linux.

2. Рассмотреть концепцию дистрибутивов Linux и понять основные причины их выбора и использования

3. Рассмотреть особенности дистрибутива Ubuntu, а также его области применения и сравнение с Debian.

2. Планируемые результаты освоения модуля

Знать	Уметь
Историю развития Linux и его ключевые этапы Причины, объясняющие, почему системы Linux называют дистрибутивами Особенности дистрибутива Ubuntu, его области применения и отличие от Debian	Изложить основные этапы истории Linux Объяснить термин «дистрибутив» и его применение в контексте Linux Сравнивать Ubuntu с другими дистрибутивами, в первую очередь с Debian, и определять сферы применения Ubuntu

3. Объем модуля

Вид учебной работы	Количество часов
Лекции	4
Самостоятельная работа	1
Форма контроля	Тест
Общая трудоемкость по учебному плану	5

4. Структура и содержание модуля

4.1. Структура модуля

№ п/п	Название темы модуля	Количество часов			Форма контроля
		Всего	Лекции	Самостоятельная работа	
1.	Историческая справка. Обоснование выбора дистрибутива Ubuntu	5	4	1	тест

4.2. Содержание модуля

Номер темы	Содержание темы
1	Кратко о курсе. Как добиться максимальных результатов? История Linux – с чего все началось? Почему системы семейства Linux принято называть дистрибутивами? Почему Ubuntu: Погружение в самый популярный дистрибутив Linux: кто разрабатывает, где применяется, сравнение с Debian

Содержание лекции

Почему Ubuntu: Погружение в самый популярный дистрибутив Linux: кто разрабатывает, где применяется, сравнение с Debian

Основные блоки
Основные редакции Ubuntu
Сравнение с Debian

Ubuntu можно с уверенностью считать самым популярным дистрибутивом Linux. Этот дистрибутив используется на домашних компьютерах и ноутбуках, в офисных средах, но чаще всего на серверах. Так, Ubuntu Server является одним из наиболее часто используемых дистрибутивов Linux, применяемых в качестве серверных решений. Согласно открытой статистике, 25% серверов на базе Linux работают именно на Ubuntu.

Основные редакции Ubuntu

Ubuntu предлагается в нескольких редакциях, из которых имеет смысл отметить три: Desktop для настольных компьютеров и ноутбуков, Server для серверов и Core для интернета вещей. Разработчиком дистрибутива является британская компания Canonical.

Преимущества Ubuntu:

- **Популярность.** Самый популярный дистрибутив позволяет использовать более широкий круг программного обеспечения, поскольку многие проекты ориентируются сугубо на Ubuntu ввиду ее популярности. Куда более важно то, что информацию о том, что и как установить и настроить, легче всего найти именно

для Ubuntu. Множество дистрибутивов Linux, таких как Linux Mint и Trisquel, основаны на Ubuntu.

- **Удобство и легкость в освоении.** «Убунта» основана на Debian, но в отличие от своего «прародителя», стремится быть дружелюбнее к неподготовленным пользователям. Большинство действий в системе пользователь может выполнить, не обращаясь к командной строке. С точки зрения среднего пользователя этот критерий рассматривается как преимущество. Кроме того, дистрибутив относительно прост в установке и русифицирован.

- **Регулярные релизы.**Ubuntu имеет четкий график релизов: каждые шесть месяцев происходит новый релиз. Отдельного внимания заслуживают LTS-версии дистрибутива. LTS или Long Term Support – это долгосрочная поддержка обновлениями, которая в данном случае составляет пять лет. Так, в апреле нынешнего года увидела свет LTS-версия Ubuntu 24.04, которая будет поддерживаться обновлениями до 2029 года.

Сравнение с Debian

Несмотря на то, что Ubuntu основана на Debian, она предлагает пользователю более дружественный интерфейс и упрощенную настройку, что делает ее более подходящей для начинающих пользователей.

Debian, в свою очередь, известен своей стабильностью и гибкостью, что делает его отличным выбором для более опытных пользователей и серверных решений, требующих высокой степени кастомизации.

Хочется добавить, что мы не «топим» за Ubuntu и не призываем вас использовать этот дистрибутив в качестве основной операционной системы всю оставшуюся жизнь. Для каждой задачи свое решение, и для учебных задач Ubuntu можно считать оптимальным выбором. В дальнейшем, когда вы наберетесь опыта, вы всегда сможете «пересесть» на другие дистрибутивы, которые лучше отвечают задачам, стоящим перед вами.

5. Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации по модулю

Промежуточная аттестация проводится в форме тестирования по каждой теме модуля. Критерии оценки уровня освоения темы:

- оценки «зачтено» заслуживает обучающийся, ответивший верно на 2 и более из 3 вопросов теста;
- оценки «не зачтено» заслуживает обучающийся, ответивший верно менее, чем на 2 из 3 вопросов теста.

Критерии оценки уровня освоения модуля:

- все тесты по темам сданы на оценку «зачтено».

Пример теста для промежуточной аттестации

1. Какая компания разрабатывает дистрибутив Ubuntu?

- Red Hat

- Canonical

- The Linux Foundation

- The Debian Project

2. С какой периодичностью выпускаются новые версии Ubuntu?

- Каждые шесть месяцев

- Каждые три месяца

- Каждый год

- Каждый месяц

3. Какие версии Ubuntu имеют долгосрочную поддержку?

- Версии LTS

- Версии Ubuntu Pro

- Все версии

- Версии non-LTS

6. Учебно-методическое и информационное обеспечение модуля

1. List of releases [Электронныйресурс] URL: https://wiki.ubuntu.com
2. ArchWiki [Электронный ресурс] URL: https://wiki.archlinux.org
3. DebianWiki [Электронный ресурс] URL: https://wiki.debian.org

Приложение № 2

РАБОЧАЯ ПРОГРАММА МОДУЛЯ 2
(Установка Linux на VirtualBox. Работа с виртуальными машинами)

По дополнительной общеобразовательной программе –
дополнительной общеразвивающей программе
«Системный администратор Linux»

1. Цель и задачи модуля

Целью освоения модуля «Установка Linux на VirtualBox. Работа с виртуальными машинами» является изучение процесса установки операционной системы Ubuntu в виртуальной среде, а также приобретение навыков работы с программными продуктами виртуализации на примере программы VirtualBox.

Задачи:

1. Изучить основы работы с VirtualBox, научиться устанавливать и настраивать эту программу.
2. Изучить основные параметры настройки виртуальных машин, такие как выделение ресурсов (память, процессоры, дисковое пространство) и сетевые настройки.
3. Изучить процесс установки операционной системы Ubuntu в VirtualBox.
4. Изучить способы создания и управления снимками состояния виртуальных машин и научиться использовать их для восстановления системы и тестирования.
5. Изучить типичные проблемы и ошибки, возникающие при работе с VirtualBox, научиться их диагностировать и устранять.

2. Планируемые результаты освоения модуля

Знания	**Умения**
1. Знать базовую теоретическую информацию о работе с программными продуктами виртуализации. 2. Знать основные настройки и конфигурации виртуальных машин, необходимые для эффективной работы.	1. Уметь устанавливать ОС Ubuntu в VirtualBox, включая создание и настройку виртуальной машины. 2. Уметь настраивать параметры виртуальной машины для оптимальной производительности и совместимости. 3. Уметь решать распространенные проблемы, возникающие при работе с VirtualBox и установке Ubuntu. 4. Уметь создавать и использовать снимки состояния виртуальной машины для обеспечения безопасной работы и восстановления системы.

3. Объем модуля

Вид учебной работы	**Количество часов**
Лекции	7
Самостоятельная работа	1,5
Форма контроля	тест, практические задания
Общая трудоемкость по учебному плану	8,5

4. Структура и содержание модуля

4.1. Структура модуля

№ п/п	Название темы модуля	Количество часов			Форма контроля
		Всего	Лекции	Самостоятельная работа	
1.	Установка Linux на VirtualBox. Работа с виртуальными машинами	8,5	7	1,5	тест, практические задания

4.2. Содержание модуля

Номер темы	Содержание темы
1	Что такое виртуальная машина? Установка VirtualBox Где достать ISO-образ Ubuntu и что такое свободное ПО? Создание и настройка виртуальной машины в VirtualBox Практика. Создание и настройка виртуальной машины в VirtualBox Видеоурок. Установка ОС Ubuntu в VirtualBox Практика. Установка ОС Ubuntu в VirtualBox Как сделать снимок виртуальной машины в VirtualBox?

Содержание лекции

Где достать ISO-образ Ubuntu и что такое свободное ПО?

Основные блоки
Что такое образ ОС?
Где достать образ Ubuntu?
Легальность свободного ПО: почему Ubuntu безопасен для скачивания?
Системные требования для установки Ubuntu

Что такое образ ОС?

Образ ОС – это файл, в котором запакованы самые разнообразные данные, от картинок до приложений. Образы дисков часто используются для записи и хранения операционных систем.
Наиболее распространенным форматом данных для образов является формат ISO. Легко заметить, что большинство дистрибутивов Linux распространяется именно в ISO-формате. Отсюда вывод: востребованность ISO-образов достаточно высока и знание методов работы с данным форматом определенно пригодится системному администратору.

Где достать образ Ubuntu?

Информация, описанная в этом уроке, носит теоретический характер. Важно, чтобы вы имели представление о том, где достать образ с Linux. Однако для успешного прохождения курса вам не нужно идти на сайт Ubuntu и скачивать образ операционной системы – мы уже сделали это за вас. Образ Ubuntu 24.04 ждет вас в нашей виртуальной лаборатории, где вы будете выполнять практические задания в рамках курса.

Идем на официальный сайт Ubuntu, находим раздел Get Ubuntu и выбираем Download Ubuntu Desktop.

Кликаем на Download 24.04 LTS. Начинается загрузка и остается дождаться ее окончания.

Легальность свободного ПО: почему Ubuntu безопасен для скачивания?

Когда вы скачиваете образ свежей версии Ubuntu с официального сайта разработчика, вы не становитесь «пиратом» и не нарушаете закон никоим образом. Ubuntu – это свободное программное обеспечение (также называют свободный софт, СПО), что означает, что его можно свободно скачивать, использовать и распространять без каких-либо ограничений. В отличие от проприетарного ПО, которое требует покупки лицензии и может быть связано с юридическими ограничениями, свободное ПО, например Ubuntu, доступно для всех без платы и позволяет каждому иметь доступ к последним обновлениям и улучшениям. То же самое можно сказать про большинство дистрибутивов Linux (но не про все!).

Системные требования для установки Ubuntu

Согласно информации с официального сайта разработчика, для установки Ubuntu 24.04 потребуется минимум 4 ГБ оперативной памяти, двухъядерный процессор с тактовой частотой от 2,0 ГГц и 25 ГБ свободного места на жестком диске. Не рекомендуем ориентироваться на минимальные системные требования: для установки Ubuntu 24.04 лучше иметь «на борту» от 6 ГБ оперативной памяти.

Что, если компьютер слабый и не соответствует рекомендуемым системным требованиям? Можно попробовать скачать и установить одну из предыдущих версий «Убунты», например 20.04. Совершенно естественно, что с каждой новой версией Ubuntu «утяжеляют», требования к «железу» медленно, но верно растут. Но все-таки лучшим решением будет выбрать менее тяжеловесный дистрибутив, например Lubuntu или Xubuntu.

5. Учебно-методическое обеспечение самостоятельной работы слушателей по модулю

Методические указания к самостоятельной работе по освоению модуля

Самостоятельная работа слушателей заключается в изучении основной литературы, изучения дополнительных (тематических) материалов и подготовке к прохождению обязательного тестирования по модулю.

6. Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации по модулю

Промежуточная аттестация проводится в форме тестирования по каждой теме модуля. Критерии оценки уровня освоения темы:

- оценки «зачтено» заслуживает обучающийся, ответивший верно на 2 и более из 3 вопросов теста;

- оценки «не зачтено» заслуживает обучающийся, ответивший верно менее, чем на 2 из 3 вопросов теста.

Критерии оценки уровня освоения модуля:

- все тесты по темам сданы на оценку «зачтено».

Пример теста для промежуточной аттестации

1. Какой из следующих инструментов можно использовать для создания виртуальной машины?

- Oracle VM VirtualBox
- Wine
- Docker
- Linux-подсистема в Windows 10

2. Параллельная работа двух операционных систем требует значительных ресурсов компьютера. Прежде всего имеет значение...

- Объем свободного пространства на жестком диске
- Семейство, к которому принадлежит основная операционная система
- Количество оперативной памяти
- Архитектура процессора

3. Какое из следующих утверждений верно?

- Изменения и ошибки в виртуальной машине остаются ограниченными в пределах ее виртуальной среды
- Изменения и ошибки в виртуальной машине влияют на основную систему
- Современные виртуальные машины работают только на ОС семейства Windows

7. Учебно-методическое и информационное обеспечение модуля

1. List of releases [Электронныйресурс] URL: https://wiki.ubuntu.com
2. ArchWiki [Электронный ресурс] URL: https://wiki.archlinux.org
3. VirtualBox User Manual [Электронный ресурс] URL: https://www.virtualbox.org/manual/ch01.html

Приложение № 3

РАБОЧАЯ ПРОГРАММА МОДУЛЯ 3
(Первые шаги в Linux. Знакомство с терминалом и пользовательским интерфейсом Ubuntu 24.04)

По дополнительной общеобразовательной программе – дополнительной общеразвивающей программе «Системный администратор Linux»

1. Цель и задачи модуля

Целью освоения модуля «Первые шаги в Linux» является приобретение базовых навыков работы в командной строке, понимание роли интерпретаторов командной строки, освоение методов управления системой с помощью пакетного менеджера APT, а также получение практических навыков по обновлению системы и настройке среды.

Задачи:

1. Ознакомиться с основами работы в командной строке и командными интерпретаторами
2. Изучить работу с пакетным менеджером APT
3. Выполнить первоначальную настройку Ubuntu, включая настройку системных параметров, оптимизацию рабочего пространства и конфигурацию пользовательских предпочтений
4. Установить гостевые дополнения VirtualBox

2. Планируемые результаты освоения модуля

Знания	**Умения**
1. Иметь представление о роли интерпретаторов командной строки 2. Иметь представление о пакетном менеджере APT 3. Знать основные шаги первоначальной настройки Ubuntu 4. Иметь представление о роли гостевых дополнений VirtualBox 5. Знать функциональные возможности пользовательского интерфейса Ubuntu 24.04. иметь представление о популярных окружениях рабочего стола	1. Уметь эффективно использовать командную строку для выполнения простых задач 2. Уметь обновлять систему, используя пакетный менеджер APT 3. Уметь производить первоначальную настройку Ubuntu 4. Уметь устанавливать гостевые дополнения VirtualBox для оптимальной работы в виртуальной среде

3. Объем модуля

Вид учебной работы	**Количество часов**
Лекции	8
Самостоятельная работа	2
Форма контроля	тест, практические задания
Общая трудоемкость по учебному плану	10

4. Структура и содержание модуля

4.1. Структура модуля

№ п/п	Название темы модуля	Количество часов			Форма контроля
		Всего	Лекции	Самостоятельная работа	
1.	Первые шаги в Linux. Знакомство с терминалом и пользовательским интерфейсом Ubuntu 24.04	10	8	2	тест, практические задания

4.2. Содержание модуля

Номер темы	Содержание темы
1	Первое знакомство с терминалом. Интерпретаторы командной строки Почему следует отдавать предпочтение работе в командной строке? APT: Основы и возможности для пользователей Ubuntu Готовим Ubuntu к работе: Обновление системы Практика. Обновление системы Первоначальная настройка Ubuntu: Раскладка клавиатуры, режим полного экрана, системное время Практика. Установка гостевых дополнений VirtualBox Обзор пользовательского интерфейса и встроенного ПО Ubuntu 24.04. Среды рабочего стола

Содержание лекции

Почему следует отдавать предпочтение работе в командной строке?

Командную строку можно назвать основным способом управления операционной системой Linux. Особенно, если речь идет о сервере – на сервера редко ставят графический интерфейс, и вся работа выполняется исключительно через командную строку. Устранение неполадок становится намного проще, если вы знаете, как отслеживать и контролировать службы, работающие в системе, используя командную строку.

Для новичка в Linux предпочтение графического пользовательского интерфейса (GUI) естественно. Необходимость запоминать многочисленные команды поначалу может напугать, но, как мы отмечали ранее, умение работать в командной строке – это «маст-хэв» для системного администратора.

Преимущества командной строки:

- Большинство задач, так же как и проблем, возникающих на компьютере с Linux, решается через командную строку. Так, всего одной командой можно скопировать тысячи файлов или найти и заменить текст во множестве документов. Например

следующая команда позволит найти все файлы, принадлежащие пользователю admin, при этом поиск будет производиться по всей операционной системе:

```
sudofind / -useradmin -print
```

Один из наших студентов выразил желание, чтобы мы подробно разобрали данную команду:

sudo*: позволяет выполнить команду, требующую для своего запуска повышенных привилегий, что необходимо для доступа к файлам и каталогам, к которым пользователь, от имени которого выполняется команда, не имеет доступа.*

find*: это команда поиска файлов и директорий. Она рекурсивно «просматривает» каталоги для поиска файлов, соответствующих заданным критериям.*

/: *Это корневой каталог файловой системы. Указание / в команде find означает, что поиск будет выполнен начиная с корневого каталога и охватит всю файловую систему.*

-useradmin*: это критерий поиска. Опция -user указывает, что нужно искать файлы, принадлежащие пользователю с именем admin.*

-print*: это действие, которое команда find должна выполнить. Флаг -print указывает find выводить путь к каждому найденному файлу или директории, соответствующему критерию поиска.*

- Командная строка позволяет выполнять сложные задачи быстрее, чем инструменты графического интерфейса. Кроме того, графический интерфейс не всегда поставляется со всеми возможностями, доступными в консольном приложении (приложении, способном работать исключительно в командной строке).

- С помощью командной строки можно создавать и запускать скрипты для автоматизации рутинных задач. Это повышает продуктивность и снижает вероятность ошибок. Мы подробно рассмотрим эту тему в модуле, посвященном созданию и использованию скриптов на Bash.

- Большинство команд, которые вы изучите, будут работать в любом дистрибутиве Linux, а также в других Unix-подобных системах, включая Android.

- Для работы командной строки не требуется мощная графика или большое количество оперативной памяти, что позволяет эффективно работать в командной строке на старых или слабых компьютерах.

5. Учебно-методическое обеспечение самостоятельной работы слушателей по модулю

Методические указания к самостоятельной работе по освоению модуля

Самостоятельная работа слушателей заключается в изучении основной литературы, изучения дополнительных (тематических) материалов и подготовке к прохождению обязательного тестирования по модулю.

6. Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации по модулю

Промежуточная аттестация проводится в форме тестирования по каждой теме модуля. Критерии оценки уровня освоения темы:

- оценки «зачтено» заслуживает обучающийся, ответивший верно на 2 и более из 3 вопросов теста;
- оценки «не зачтено» заслуживает обучающийся, ответивший верно менее, чем на 2 из 3 вопросов теста.

Критерии оценки уровня освоения модуля:

- все тесты по темам сданы на оценку «зачтено».

Пример теста для промежуточной аттестации

1. Какой основной способ администрирования операционной системы Linux на серверах?

- Использование скриптов автоматизации для управления системными задачами
- Работа в графическом интерфейсе пользователя (GUI)
- Работа в командной строке
- Использование мобильного приложения

2. В каких операционных системах, кроме Linux, работает большинство терминальных команд, которые вы изучите?

- В MS-DOS
- В большинстве Unix-подобных операционных систем, включая Android
- Только в Ubuntu
- Только в операционных системах семействах Linux

3. Каковы преимущества использования командной строки на старых или слабых компьютерах?

- Командная строка не требует мощной графики или большого количества оперативной памяти
- Командная строка существенно повышает скорость интернет-соединения
- Командная строка предоставляет аппаратное ускорение
- Командная строка в принципе ориентирована на старые и слабые компьютеры

7. Учебно-методическое и информационное обеспечение модуля

1. List of releases [Электронныйресурс] URL: https://wiki.ubuntu.com
2. ArchWiki [Электронный ресурс] URL: https://wiki.archlinux.org
3. VirtualBox User Manual [Электронный ресурс] URL: https://www.virtualbox.org/manual/ch01.html

Приложение № 4

РАБОЧАЯ ПРОГРАММА МОДУЛЯ 4
(Необходимый минимум. Изучение основных команд, работа с редакторами Nano и Vim)

По дополнительной общеобразовательной программе –
дополнительной общеразвивающей программе
«Системный администратор Linux»

1. Цель и задачи модуля

Целью освоения модуля «Необходимый минимум. Изучение основных команд, работа с редакторами Nano и Vim» является предоставление студентам фундаментальных знаний и навыков, необходимых для эффективного управления системами на базе ядра Linux. Это включает умение пользоваться базовыми командами для навигации и управления файлами в командной строке, а также освоение работы с текстовыми редакторами Nano и Vim для редактирования конфигурационных файлов и создания скриптов.

Задачи:

1. Изучить основные команды командной строки. Изучение команд для навигации по файловой системе, управления файлами и каталогами, а также выполнения основных операций с текстовыми файлами.
2. Приобрести навыки работы с текстовым редактором Nano. Обучение использованию редактора Nano для создания, редактирования и сохранения текстовых файлов, а также освоение его основных команд и функций.
3. Приобрести навыки работы с текстовым редактором Vim. Понимание особенностей работы в редакторе Vim, включая режимы редактирования, команды для навигации, поиска и замены текста, а также использование расширенных функций для эффективного редактирования.

2. Планируемые результаты освоения модуля

Знать	Уметь
1. Основные команды командной строки 2. Основные функции и команды текстового редактора Nano. 3. Основные режимы и команды текстового редактора Vim.	1. Применять и комбинировать основные команды командной строки 2. Работать в редакторе Nano, эффективно использовать Nano для создания и редактирования текстовых файлов. 3. Работать в редакторе Vim, переключаться между режимами редактирования, выполнять навигацию по документу, искать и заменять текст.

3. Объем модуля

Вид учебной работы	Количество часов
Лекции	6
Самостоятельная работа	2
Форма контроля	тест, практические задания
Общая трудоемкость по учебному плану	8

4. Структура и содержание модуля

4.1. Структура модуля

№ п/п	Название темы модуля	Количество часов			Форма контроля
		Всего	Лекции	Самостоятельная работа	
1.	Необходимый минимум. Изучение основных команд, работа с редакторами Nano и Vim	8	6	2	тест, практические задания

4.2. Содержание модуля

Номер темы	Содержание темы
1	Команды Linux: с чего начать? Основные команды Linux Практика. Основные команды Linux Nano в арсенале системного администратора. Как копировать текст в терминале Linux? Практика. Работа с Nano Vim: Как эффективно работать с конфигурационными файлами? Практика. Работа с Vim Практическая тренировка: Изучаем Vim в формате увлекательной игры

Содержание лекции

Vim: Как эффективно работать с конфигурационными файлами?

Основные блоки
Установка Vim
Основы работы с Vim
Интерфейс Vim
Немного о Neovim
Минутка юмора
Заключение

Vim (ViIMproved) – это мощный текстовый редактор, который представляет собой улучшенную версию редактора Vi. Благодаря обширному функционалу и гибкой настройке Vim широко популярен среди пользователей Unix-подобных операционных систем и считается одним из лучших текстовых редакторов на сегодняшний день.

Так как курс не посвящен Vim и возможно, что вам вообще не придется использовать этот редактор в своей работе, в этом уроке мы ограничимся рассмотрением его режимов работы и основными командами, которые обеспечат комфортное использование редактора. Кроме того, мы научим вас выходить из Vim!

Установка Vim

Не во всех дистрибутивах Linux предустановлен Vim. Если в вашем дистрибутиве нет этого редактора, перейдите в терминал и выполните следующую команду:

sudoaptinstallvim -y

Система запросит пароль от вашей текущей учетной записи. Введите пароль и нажмите Enter. После ввода пароля начнется процесс установки текстового редактора Vim.

Основы работы с Vim

Чтобы начать работу с Vim, перейдите в терминал и введите команду:

vim

Это откроет редактор.

Если вы хотите открыть конкретный файл, укажите имя этого файла в качестве аргумента – по аналогии с nano:

vim /etc/apt/sources.list

Vim работает в четырех специальных режимах, основными из которых являются режим команд (Command mode) и режим вставки (Insertmode). В режиме команд пользователь имеет возможность выполнять команды. Например если пользователь желает сохранить редактируемый файл, ему нужно перейти в командный режим и выполнить команду :w. Кстати, при запуске редактора вы оказываетесь именно в командном режиме.

В режиме вставки пользователь имеет возможность вводить или редактировать текст, другими словами реализуется именно тот функционал, который ожидается от текстового редактора. Чтобы перейти в режим вставки, достаточно нажать клавишу i. Попробуйте сделать это сами. Вы сразу поймете, что теперь находитесь в режиме вставки, потому что сможете печатать текст. Кроме того, о переходе в режим вставки сообщит строка состояния, на которой отобразится «`-- INSERT --`».

После переключения в режим вставки процесс редактирования происходит так же, как и в обыкновенном текстовом редакторе, например за удаление текста отвечают клавиши Delete или Backspace, за переход в начало строки – клавиша Home, за переход в конец строки – клавиша End.

Режим визуального выделения (Visual mode) используется для выделения текста для последующего копирования или вырезания. Для переключения в этот режим нужно нажать клавишу v. Если такой способ копирования кажется вам не слишком удобным, рекомендуем обратиться к уроку, посвященному редактору Nano, где мы предложили универсальный метод копирования текста в терминале Linux.

Режим замены (Replacemode) используется для замены существующего текста. Для перехода в этот режим нужно нажать клавишу R (именно прописная R, а не строчная).

В Vim доступен целый ряд команд, однако мы ограничимся только наиболее важными.

Перечислим основные команды Vim:

- :w – сохранить редактируемый файл
- :q – выйти из редактора. Если в файл были внесены какие-нибудь изменения, то при попытке ввода этой команды Vim откажется выходить и предложит либо сохранить файл, либо воспользоваться одной из следующий команд
- :wq – сохранить редактируемый файл и выйти из редактора. В этом есть своя логика: w – это сокращение от слова write (сохранить изменения), q – это сокращение от слова quit (выйти из редактора).
- :q! – выйти из редактора без сохранения изменений
- :!*имя_команды* – выполнить команду оболочки *имя_команды*, находясь в редакторе. Чтобы вернуться в Vim, следует нажать клавишу Enter. Этот трюк позволяет выполнять команды оболочки и возвращаться к редактированию текста без необходимости выхода из Vim.

Напомним, перед тем, как вводить команды, убедитесь, что вы находитесь в командном режиме. Если это не так, нажмите клавишу Esc. Другими словами, если вы, редактируя какой-нибудь файл, решили его сохранить и сразу после этого выйти из редактора, вам нужно сначала нажать на клавишу Esc и только затем ввести команду :wq.

Интерфейс Vim

Перейдем к интерфейсу. Интерфейс Vim можно разделить на три части: главное окно, строка команд и строка состояния. Главное окно – это основная часть экрана, где отображается редактируемый текст. В этом окне вы можете перемещаться по тексту, вносить изменения и выполнять команды.

Строка команд(не путать с командной строкой!) находится в нижней части экрана. Здесь отображаются команды, которые вы вводите в режиме команд. Например, когда вы вводите `:w` для сохранения файла или `:q` для выхода из редактора, эти команды отображаются на строке команд.

Строка состояния, она же statusbar располагается там же, где и строка команд. Она показывает текущую информацию о файле, такую как имя файла, номер строки и столбца, а также режим редактирования (например, «`-- INSERT --`» для режима вставки).

Стоит отметить, что Vim может работать не только в командной строке, но и посредством графического интерфейса (gVim).

Немного о Neovim

Многие специалисты советуют новичкам начинать работу не с Vim, а с Neovim. Этот редактор является современным форком Vim.

Установить Neovim можно с помощью следующей команды:

```
sudo apt install neovim -y
```

Чтобы начать работу с Neovim, введите команду:

```
nvim
```

Как можно заметить, интерфейс Neovim практически не отличается от оригинального Vim. Однако Neovim предлагает новичкам дополнительные возможности для обучения. Чтобы начать, убедитесь, что вы находитесь в командном режиме, а затем введите команду `:Tutor`. Пройдите все уроки, на это уйдет примерно полтора часа, но это стоит того. В интернете можно найти русскую версию данного туториала.

Минутка юмора

Vim также известен своей крутой кривой обучения и многочисленными шутками среди пользователей Linux на эту тему. Одна из самых популярных шуток: «Я очень много пользуюсь редактором Vim. Главным образом потому, что не знаю, как выйти из него». Эта шутка отражает реальную проблему, с которой сталкиваются новички, пытаясь безуспешно закрыть текстовый редактор. Другой вариант той же шутки: «Vim – это как отель «Калифорния»: ты можешь войти, но не сможешь выйти.» Это прямой намек на культовую песню рок-группы The Eagles. Текст песни описывает путешествие, которое начинается с невинного любопытства и заканчивается осознанием того, что лирический герой застрял в отеле, из которого нет выхода.

Кроме шуток, как только вы освоите основные команды, работа с Vim покажется простой и возможно, даже удобной.

Заключение

Мы рассмотрели два консольных редактора – Nano и Vim. Какой из них выбрать для повседневного использования или работы, зависит от ваших предпочтений и потребностей. Тем не менее, рекомендуется овладеть навыками работы с обоими редакторами, чтобы быть готовым к различным задачам.

5. Учебно-методическое обеспечение самостоятельной работы слушателей по модулю

Методические указания к самостоятельной работе по освоению модуля

Самостоятельная работа слушателей заключается в изучении основной литературы, изучения дополнительных (тематических) материалов и подготовке к прохождению обязательного тестирования по модулю.

6. Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации по модулю

Промежуточная аттестация проводится в форме тестирования по каждой теме модуля.
Критерии оценки уровня освоения темы:

- оценки «зачтено» заслуживает обучающийся, ответивший верно на 2 и более из 3 вопросов теста;
- оценки «не зачтено» заслуживает обучающийся, ответивший верно менее, чем на 2 из 3 вопросов теста.

Критерии оценки уровня освоения модуля:

- все тесты по темам сданы на оценку «зачтено».

Пример теста для промежуточной аттестации

1. В каком режиме вы оказываетесь при запуске Vim?

- Режим команд
- Режим замены
- Режим вставки
- Режим визуального выделения

2. Какая клавиша используется для перехода в режим вставки (Insertmode) в Vim?
   - v
   - Super
   - i
   - Esc
3. Какая команда используется для выхода из Vim без сохранения изменений, внесенных в редактируемый файл?
   - :w
   - :wq
   - :q!
   - :q

7. Учебно-методическое и информационное обеспечение модуля

1. ArchWiki [Электронный ресурс] URL:https://wiki.archlinux.org
2. DebianWiki [Электронный ресурс] URL:https://wiki.debian.org/
3. Linux по-русски [Электронный ресурс] URL:https://rus-linux.net/

Приложение № 5

РАБОЧАЯ ПРОГРАММА МОДУЛЯ 5
(Погружение в администрирование Linux)

По дополнительной общеобразовательной программе –
дополнительной общеразвивающей программе
«Системный администратор Linux»

1. Цель и задачи модуля

Целью освоения модуля «Погружение в администрирование Linux» является предоставление учащимся необходимых знаний и навыков для эффективного управления и настройки систем на базе Linux. Этот модуль направлен на понимание ключевых аспектов системного администрирования, включая установку и конфигурацию программного обеспечения, управление пользовательскими учетными записями, выполнение резервного копирования и восстановление данных, а также обеспечение безопасности и оптимизацию работы системы. Овладев этими навыками, учащиеся смогут уверенно и эффективно справляться с задачами администрирования Linux-систем, что значительно повысит их квалификацию и профессиональные компетенции в этой области.

Задачи:

1. Овладеть навыками управления пользователями и группами, включая создание, удаление и модификацию учетных записей пользователей, а также настройку прав доступа и привилегий.
2. Освоить использование командной строки и основных утилит Linux для выполнения системных задач, таких как управление процессами, мониторинг ресурсов системы и выполнение скриптов.
3. Познакомиться с методами резервного копирования и восстановления данных, включая восстановление файлов из резервных копий.
4. Научиться работать с системами пакетного управления для установки, обновления и удаления программного обеспечения.
5. Исследовать структуру каталогов в Linux, изучить иерархию файловой системы, понять основные каталоги и их назначение. Овладеть знанием о том, где расположены системные файлы, конфигурационные файлы и пользовательские данные, чтобы уверенно ориентироваться в файловой системе Linux.
6. Овладеть утилитами find и grep для эффективного поиска и фильтрации данных в Linux.
7. Освоить методы получения системной информации через командную строку Linux. Изучить команды, такие как dmidecode, free, uname, и другие, для сбора данных об аппаратном обеспечении и конфигурации системы.
8. Познакомиться с философией Unix, изучить основные принципы и концепции, такие как принцип "делай одну вещь и делай это хорошо", чтобы глубже понять подход к разработке и администрированию систем, основанных на Unix.

2. Планируемые результаты освоения модуля

Знать	Уметь

1. Принципы управления пакетами в Debian/Ubuntu 2. Команды и утилиты для загрузки файлов из интернета 3. Структуру каталогов в Linux и назначения ключевых каталогов 4. Команды и утилиты для поиска и фильтрации данных 5. Команды для получения системной информации 6. Основы философии Unix и принципы разработки программ для Unix-подобных систем	1. Создавать и управлять пользователями и группами 2. Устанавливать, обновлять и удалять пакеты с использованием утилит пакетного менеджера 3. Загружать файлы из интернета и управлять загрузками 4. Ориентироваться в структуре каталогов и эффективно управлять файлами и каталогами 5. Настраивать права доступа к файлам и каталогам 6. Применять утилиты для поиска и фильтрации данных в файловой системе 7. Получать и интерпретировать системную информацию для диагностики и администрирования

3. Объем модуля

Вид учебной работы	Количество часов
Лекции	25
Самостоятельная работа	7
Форма контроля	тест, практические задания
Общая трудоемкость по учебному плану	32

4. Структура и содержание модуля

4.1. Структура модуля

№ п/п	Название темы модуля	Количество часов			Форма контроля
		Всего	Лекции	Самостоятельная работа	
1.	Погружение в администрирование Linux	32	25	7	тесты, практические задания

4.2. Содержание модуля

Номер темы	Содержание темы
1	Пользователи в Linux: Понимание Sudo и Su Пользователи в Linux: Создание и управление Пользователи в Linux: Назначение прав доступа к файлам и директориям. Понимание chmod Пользователи в Linux: Назначение прав доступа к файлам и директориям. Как дать права на исполнение скрипту для его запуска? Группы пользователей в Linux: Создаем группы, добавляем и удаляем из

	них пользователей Пользователи в Linux: Передача прав собственности на файлы или директории Практика. Работа с файловыми атрибутами в Linux Эффективные способы установки ПО в Ubuntu: погружение в APT Эффективные способы установки ПО в Ubuntu: Установка из DEB-пакетов Эффективные способы установки ПО в Ubuntu: Использование wget для загрузки установочных файлов Эффективные способы установки ПО в Ubuntu: Клонирование удаленного репозитория. Работа с Git Создание и распаковка архивов в Linux Как создать резервную копию домашней директории: Учимся делать бэкапы в Linux Практика. Создание бэкапа данных Структура каталогов Linux. Абсолютные и относительные пути Скрытые файлы и каталоги в Linux Основы работы с SSH: Безопасное управление удаленными серверами Основы работы с SSH: создание ключей, изменение порта по умолчанию, отладка Практика. Подключение к серверу по SSH Управление службами в Linux Практика. Управление службами в Linux Управление процессами в Linux Практика. Управление процессами в Linux Эффективный поиск файлов и текста в Linux: Используем find и grep Как использовать команды find и grep в Linux: Практические примеры для реальных сценариев Практика. Применение команд find и grep в Linux Получение системной информации в командной строке Linux Практика. Получение системной информации в командной строке Linux Философия Unix. Концепция «все есть файл»

Содержание лекции

Философия Unix. Концепция «все есть файл»

Основные блоки
Три принципа Макилроя
Все есть файл
Формат времени Unix

Как системный администратор, вы должны иметь хотя бы базовое представление о философии Linux (Unix). Это знание станет основой для более глубокого понимания Unix-подобных операционных систем. Мы начнем с рассмотрения трех принципов, сформулированных Макилроем.

Три принципа Макилроя

Американский программист Дуг Макилрой, который наиболее известен первоначальной разработкой конвейера в семействе операционных систем Unix, обобщил философию Unix следующим образом:

- **Пиши программы, каждая из которых делает ровно одну вещь и делает ее хорошо.** Этот принцип часто выражается кратко: «Делай одну вещь, и делай ее хорошо». Проще говоря, программа соответствует философии Unix, когда имеет простое и соответствующее поведение, с понятными пользователю и другим программам параметрами.
- **Пиши программы так, чтобы было удобно использовать их вместе.** Этот принцип подразумевает, что программы должны разрабатываться с учетом возможности их совместного использования, что облегчает создание более сложных решений путем объединения простых инструментов.
- **Пиши программы для обработки текстовых потоков, так как это универсальный интерфейс.** Текстовые потоки – это концепция, основанная на представлении данных в виде последовательности символов. В Unix-подобных системах текстовые потоки играют ключевую роль в обработке данных и взаимодействии между программами.

Стоит отметить, что в сообществе поклонников *nix такую философию называют Unix-way. В данном случае под Unix-way понимается философия, в соответствии с которой приветствуются небольшие, но отлаженные программы, решающие небольшие задачи, но которые можно объединять в конвейеры и использовать с помощью командной строки или в скриптах.

Все есть файл

В семействе операционных систем Unix есть идея, которая описывается словами «everything is a file» – «все есть файл». Это означает, что вся работа с операционной системой сводится к файлам. Поскольку Linux – это Unix-подобная операционная система, то данная концепция применима и здесь.

Почти все ресурсы операционной системы представлены в виде файлов. Другими словами, файлы – это не только объекты для хранения информации, но и устройства, сокеты и многое другое. Дисковые накопители – это тоже определенный тип файла. Более того, директория – это тоже файл. Вероятно, вы удивитесь, но директорию можно прочитать утилитой cat. Правда, в современных операционных системах это сделать не получится. Давайте попробуем прочитать директорию /tmp:

```
cat /tmp
```

Выполнив эту операцию, мы получим ожидаемый результат: система проинформирует нас о том, что команда `cat` предназначена для вывода содержимого файлов, а не каталогов.

```
cat: /tmp: Is a directory
```

Если вы попытаетесь выполнить команду «`cat /tmp`», вы получите ошибку, так как `cat` ожидает файл, а не директорию. Однако в операционных системах FreeBSD до версии 12 можно было использовать `cat` для вывода содержимого каталогов.

А вот если к этой команде дописать слэш и звездочку…

```
cat /tmp/*
```

...то будет выведено содержимое всех доступных для чтения файлов, находящихся в каталоге /tmp.

Философия «все есть файл» пронизывает операционную систему Linux и разработанные для нее инструменты. Эти принципы лежат в основе проектирования и работы операционных систем. Понимание этих принципов поможет вам в эффективном администрировании систем.

Формат времени Unix

Важно также упомянуть, что Unix-системы используют специфический формат времени, который называется **Unix-время** или **эпоха Unix**. Этот формат отсчитывает время в секундах с 00:00:00 1 января 1970 года по всемирному координированному времени (UTC). Таким образом, каждый момент времени представлен в виде числа, которое соответствует количеству секунд, прошедших с этой даты. Это упрощает работу с временными метками в программировании и администрировании, поскольку позволяет легко производить вычисления и сравнения.

5. Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации по модулю

Промежуточная аттестация проводится в форме тестирования по каждой теме модуля. Критерии оценки уровня освоения темы:

- оценки «зачтено» заслуживает обучающийся, ответивший верно на 2 и более из 3 вопросов теста;
- оценки «не зачтено» заслуживает обучающийся, ответивший верно менее, чем на 2 из 3 вопросов теста.

Критерии оценки уровня освоения модуля:

- все тесты по темам сданы на оценку «зачтено».

Пример теста для промежуточной аттестации

1. Какая из следующих программных философий не является частью принципов, описанных Макилроем?
   - Пиши программы, каждая из которых делает ровно одну вещь и делает ее хорошо.
   - Пиши программы, которые предназначены только для работы в одной операционной системе.
   - Пиши программы для обработки текстовых потоков, так как это универсальный интерфейс.
   - Пиши программы так, чтобы было удобно использовать их вместе.
2. Что понимается под концепцией «все есть файл»?
   - Файлы – это не только объекты для хранения информации, но и устройства, дисковые накопители, сокеты и многое другое.
   - Все данные системы хранятся только в бинарном формате.
   - Директории не являются файлами и их нельзя обрабатывать как файлы.
   - Только текстовые файлы представлены в виде файлов.
3. Что понимается под термином Unix-way?
   - Подход, при котором программы должны быть многофункциональными и универсальными, чтобы решать широкий спектр задач.

- Подход, требующий использования только встроенных в систему утилит для решения всех задач без стороннего ПО.
- Философия, в соответствии с которой приветствуются небольшие, но отлаженные программы, решающие небольшие задачи, но которые можно объединять в конвейеры и использовать с помощью командной строки или в скриптах.
- Метод разработки, ориентированный на создание графических пользовательских интерфейсов для улучшения взаимодействия с пользователем.

6. Учебно-методическое и информационное обеспечение модуля

1. ArchWiki [Электронный ресурс] URL:https://wiki.archlinux.org
2. DebianWiki [Электронный ресурс] URL:https://wiki.debian.org/
3. Linux по-русски [Электронный ресурс] URL:https://rus-linux.net/

Приложение № 6

РАБОЧАЯ ПРОГРАММА МОДУЛЯ 6
(Основы программирования на Bash)

По дополнительной общеобразовательной программе –
дополнительной общеразвивающей программе
«Системный администратор Linux»

1. Цель и задачи модуля

Целью освоения модуля «Основы программирования на Bash» является приобретение базовых навыков для написания и эффективного использования Bash-скриптов, которые автоматизируют задачи в операционной системе и упрощают управление системными ресурсами.

Задачи:

1. Ознакомиться с основными принципами и синтаксисом Bash.
2. Научиться создавать и редактировать простые Bash-скрипты.
3. Изучить методы обработки аргументов командной строки в скриптах.
4. Овладеть навыками использования управляющих конструкций (циклы, условные операторы) для автоматизации более сложных сценариев и задач в скриптах.
5. Развить навыки написания скриптов для автоматизации административных задач и управления системными процессами.
6. Предоставить студенту возможность практиковаться в написании и запуске скриптов в различных условиях, чтобы закрепить полученные знания и навыки.

2. Планируемые результаты освоения модуля

Знать	Уметь
1. Основные принципы и синтаксис Bash 2. Способы объявления и использования переменных 3. Управляющие конструкции (циклы, условные операторы)	1. Создавать и редактировать простые Bash-скрипты 2. Обрабатывать аргументы командной строки в скриптах 3. Использовать управляющие конструкции для автоматизации задач 4. Работать с файлами и директориями в скриптах

3. Объем модуля

Вид учебной работы	Количество часов
Лекции	8
Самостоятельная работа	2
Форма контроля	тест, практические задания
Общая трудоемкость по учебному плану	10

4. Структура и содержание модуля

4.1. Структура модуля

№ п/п	Название темы модуля	Количество часов			Форма контроля
		Всего	Лекции	Самостоятельная работа	
1.	Основы программирования на Bash	10	8	2	тесты, практические задания

4.2. Содержание модуля

Номер темы	Содержание темы
1	Введение в Bash-скриптинг. Отличия Bash от других языков программирования Переменные в Bash. Переменные окружения Условные операторы: как использовать конструкцию if-else в Bash? Циклы в Bash: как использовать циклы for, while и until для автоматизации задач? Работа с аргументами в Bash Практика. Основы программирования на Bash Практика. Где прокачать навыки работы в терминале Linux? Начинаем с Bandit! Практическая тренировка: используем CTF-платформы для углубленного изучения Linux

Содержание лекции

Работа с аргументами в Bash

Основные блоки
Передача аргументов
Доступ к аргументам
Обработка входных данных
Обработка ошибок
Массивы аргументов

В контексте Bash-скриптинга аргументы командной строки представляют собой параметры, которые передаются скрипту при его запуске. Аргументы позволяют скрипту получать данные или инструкции из командной строки, что делает Bash-скрипт более гибким и настраиваемым.

Передача аргументов

При запуске Bash-скрипта аргументы передаются скрипту после его имени. Например:

./myscript.sh arg1 arg2 arg3

Здесь arg1, arg2 и arg3 являются аргументами, которые скрипт myscript.sh будет обрабатывать.

Доступ к аргументам

Внутри скрипта аргументы командной строки можно получить с помощью специальных переменных:

- $0 – имя скрипта.
- $1 – первый аргумент, $2 – второй аргумент и так далее до $9.
- $# – количество аргументов, переданных скрипту.
- $@ – все аргументы, переданные скрипту, в виде списка.
- $* – все аргументы в виде одной строки (с пробелами между ними).
- $? – код завершения последней команды.

Пример кода на Bash, использующего аргументы командной строки:

```
#!/bin/bash

#  Выводит имя самого скрипта. Переменная $0 содержит путь к скрипту, как он был вызван

echo "Имя скрипта: $0"

# Выводит первый аргумент, переданный скрипту. Переменная $1 содержит значение первого аргумента

echo "Первый аргумент: $1"

# Выводит второй аргумент, переданный скрипту. Переменная $2 содержит значение второго аргумента

echo "Второй аргумент: $2"

# Выводит количество аргументов, переданных скрипту. Переменная $# содержит общее количество аргументов

echo "Количество аргументов: $#"

# Выводит все переданные аргументы. Переменная $@ содержит все аргументы в виде отдельных слов, что позволяет корректно обрабатывать аргументы, содержащие пробелы

echo "Все аргументы: $@"
```

Сохраним этот код в файл с именем myscript.sh, а затем выполним скрипт, передав ему аргументы «яблоко» и «банан»:

```
./myscript.sh яблоко банан
```

Мы видим, что скрипт выполнил свою работу корректно.

Обработка входных данных

Скрипты могут принимать аргументы, чтобы определить, какие файлы обрабатывать или какие действия выполнять. Напишем скрипт, который будет проверять, существует ли файл filename.txt.

```
#!/bin/bash

# Эта строка проверяет, существует ли файл, путь к которому передан в качестве первого аргумента скрипту ($1). Флаг -f используется для проверки, является ли указанный путь существующим файлом

if[ -f "$1" ]

# Если условие проверки (файл существует) истинно, выполняется код, находящийся между then и else

  then

    # Если файл существует, выводится сообщение, подтверждающее это, где $1 заменяется на имя файла, переданное в качестве аргумента

    echo "Файл $1 существует."

# Если условие проверки (файл существует) ложно, выполняется код, находящийся после else

else

  # Если файл не существует, выводится сообщение, информирующее об этом.

  echo "Файл $1 не найден."

# Завершает конструкцию if

fi
```

Выполним этот скрипт, передав аргументом файл filename.txt:

```
./myscript.sh filename.txt
```

Отлично! Скрипт выполнил свою работу корректно.

Теперь передадим аргументом несуществующий файл:

```
./myscript.sh main.py
```

Скрипт сообщает о том, что файл main.py не найден.

Обработка ошибок

Напишем скрипт, задачей которого является проверка наличия необходимого количества аргументов и их корректности.

```
#!/bin/bash

# Эта строка проверяет количество переданных аргументов командной строки. $# содержит количество аргументов, переданных скрипту. Флаг -lt используется для проверки, меньше ли это количество двух

if[ $# -lt 2 ]

  # Если условие проверки ($# -lt 2) истинно, то есть если передано менее двух аргументов, выполняется код, находящийся между then и fi

  then

    # Если передано недостаточно аргументов, выводится сообщение об ошибке

    echo "Ошибка: недостаточно аргументов."

    # Эта строка выводит подсказку по использованию скрипта. $0 представляет собой имя скрипта. Сообщение объясняет, что необходимо передать два аргумента для корректной работы скрипта.

    echo "Использование: $0 <arg1><arg2>"

    # Завершает выполнение скрипта с ошибкой

    exit 1

# Завершает конструкцию if

fi
```

Выполним этот скрипт с аргументами «яблоко» и «банан

./myscript.sh яблоко банан

Мы видим, что скрипт выполнил свою работу корректно

Скрипту необходимо передать как минимум два аргумента. Если передать менее двух аргументов, будет выведено сообщение об ошибке. Кроме того, при неправильном использовании будет выведена подсказка. Давайте убедимся, что это так, передав аргумент «яблоко»:

./myscript.sh яблоко

Как оказалось, все верно!

Массивы аргументов

Напоследок рассмотрим использование $@ для работы с аргументами как с массивом. В Bash массивы представляют собой структуру данных, позволяющую хранить несколько значений в одной переменной. Каждый элемент массива имеет уникальный индекс,

начинающийся с нуля. Использование массивов особенно удобно, когда необходимо одновременно обрабатывать несколько связанных значений.

«$@» – это специальная конструкция в Bash, которая используется для обработки аргументов, переданных скрипту. Она позволяет работать с аргументами как с массивом.

Напишем скрипт.

```
#!/bin/bash

# Это начало цикла for, который перебирает все аргументы, переданные скрипту. "$@" — это специальная переменная, содержащая все аргументы в виде отдельных слов, что позволяет корректно обрабатывать аргументы, содержащие пробелы. Каждый аргумент поочередно будет присвоен переменной arg

forargin "$@"

  # Указывает начало блока команд, который будет выполняться для каждого аргумента. В данном случае команда всего одна

  do

    # Эта команда выводит на экран строку с текущим аргументом, присвоенным переменной arg

    echo "Аргумент: $arg"

  # Завершает цикл for

  done
```

Выполним этот скрипт с аргументами «яблоко», «банан» и «апельсин»:

./myscript.sh яблоко банан апельсин

Мы видим, что скрипт выполнил свою работу корректно.

5. Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации по модулю

Промежуточная аттестация проводится в форме тестирования по каждой теме модуля. Критерии оценки уровня освоения темы:

- оценки «зачтено» заслуживает обучающийся, ответивший верно на 2 и более из 3 вопросов теста;
- оценки «не зачтено» заслуживает обучающийся, ответивший верно менее, чем на 2 из 3 вопросов теста.

Критерии оценки уровня освоения модуля:

- все тесты по темам сданы на оценку «зачтено».

Пример теста для промежуточной аттестации

1. Какой символ(-ы) используется для объявления комментария в Bash?
   - #
   - /* */
   - --
   - //

2. Что делает строка #!/bin/bash?
   - Указывает имя файла, который нужно открыть
   - Подключает внешнюю библиотеку Bash
   - Это комментарий в коде
   - Указывает, что скрипт следует выполнять с помощью интерпретатора Bash

3. Какую команду нужно использовать для того, чтобы сделать скрипт script.sh исполняемым?

   - chmod +w script.sh
   - chmod +x script.sh
   - chmod 755 script.sh
   - chmod 777 script.sh

6. Учебно-методическое и информационное обеспечение модуля

1. ArchWiki [Электронный ресурс] URL:https://wiki.archlinux.org
2. DebianWiki [Электронный ресурс] URL:https://wiki.debian.org/
3. Linux по-русски [Электронный ресурс] URL:https://rus-linux.net/

Приложение № 7

РАБОЧАЯ ПРОГРАММА МОДУЛЯ 7
(Работа с документами форматов Word и Excel в LibreOffice)

По дополнительной общеобразовательной программе –
дополнительной общеразвивающей программе
«Системный администратор Linux»

1. Цель и задачи модуля

Целью освоения модуля «Работа с документами форматов Word и Excel в LibreOffice» является развитие навыков эффективного использования программного обеспечения LibreOffice для создания, редактирования и форматирования текстовых документов и электронных таблиц, освоение основных функций и инструментов, необходимых для работы с данными, а также умение осуществлять обмен информацией между различными форматами документов.

Задачи:

1. Ознакомиться с основными компонентами и инструментами программ LibreOfficeWriter (для работы с текстовыми документами) и LibreOfficeCalc (для работы с таблицами).
2. Обучиться созданию текстовых документов, форматированию текста, использованию стилей и шаблонов (в LibreOfficeWriter), созданию и редактированию таблиц, использованию формул и функций (в LibreOfficeCalc).
3. Изучить различные методы форматирования, такие как изменение шрифтов, выравнивание текста, добавление списков и таблиц.
4. Изучить способы вставки изображений, графиков и других объектов в документы и таблицы
5. Научиться сохранять, открывать и конвертировать документы в различные форматы, в том числе .docx и .xlsx.

2. Планируемые результаты освоения модуля

Знать	Уметь
1. Основные функции и возможности LibreOfficeWriter и LibreOfficeCalc. 2. Форматы файлов, используемые в LibreOffice. 3. Основные принципы работы с шаблонами документов. 4. Способы вставки и редактирования графических объектов.	1. Создавать и форматировать текстовые документы в Writer. 2. Работать с таблицами и формулами в Calc. 3. Использовать стили и оформления для унификации документов. 4. Производить вставку и редактирование графических объектов. 5. Экспортировать и сохранять документы в различных форматах.

3. Объем модуля

Вид учебной работы	Количество часов
Лекции	3
Самостоятельная работа	1,5
Форма контроля	тест, практические задания
Общая трудоемкость по учебному плану	4,5

4. Структура и содержание модуля

4.1. Структура модуля

№ п/п	Название темы модуля	Количество часов			Форма контроля
		Всего	Лекции	Самостоятельная работа	
1.	Работа с документами форматов Word и Excel в LibreOffice	4,5	3	1,5	тесты, практические задания

4.2. Содержание модуля

Номер темы	Содержание темы
1	Что такое LibreOffice? Работа с LibreOffice Writer Практика. Работа с LibreOfficeWriter Работа с LibreOfficeCalc

Содержание лекции

Чтотакое LibreOffice?

Основные блоки
Расширения LibreOffice
Обновления LibreOffice: Новые версии
Альтернативы LibreOffice

Вы успешно завершили шесть основных модулей курса и теперь приступили к седьмому. Подчеркнем, что седьмой и восьмой модули следует рассматривать как дополнение к основной программе курса, которую вы уже изучили.

В этом и следующем уроках мы познакомимся с LibreOffice – мощным офисным пакетом с открытым исходным кодом, который предоставляет обширные возможности для работы с текстовыми документами, таблицами, презентациями и векторной графикой. LibreOffice разрабатывается сообществом из более чем 480 программистов под эгидой некоммерческого фонда The Document Foundation за счет пожертвований отдельных лиц и организаций. LibreOffice бесплатен как для частного, так и для образовательного или коммерческого использования.

LibreOffice включает в себя текстовый процессор Writer, табличный процессор Calc, программу для подготовки презентаций Impress и другие приложения, позволяющие эффективно работать с текстовыми и графическими данными. В Ubuntu LibreOfficeиспользуется в качестве стандартного офисного приложения.

Не стоит воспринимать эти уроки как исчерпывающее руководство по LibreOffice. Поскольку наш курс не ориентирован на глубокое изучение офисных приложений, мы ограничимся лишь основами работы с этим инструментом. Наша цель – предоставить

базовое представление о функционале LibreOffice, чтобы при необходимости вы могли уверенно использовать его для работы с текстовыми документами и таблицами.

Расширения LibreOffice

LibreOffice поддерживает множество расширений, которые могут значительно обогатить его функциональные возможности. Расширения LibreOffice представляют собой программные модули, которые можно установить дополнительно к стандартной конфигурации LibreOffice для добавления новых функций к одному из приложений (Writer, Calc, Impress и т.д.) или всем приложениям вместе. На официальном сайте вы можете найти и установить расширения, которые отвечают вашим требованиям, а также поддержку, если с использованием офисного пакета возникнут какие-либо проблемы.

Обновления LibreOffice: Новые версии

Важно отметить, что офисный пакет LibreOffice постоянно развивается, и новые версии программного обеспечения регулярно выходят, предлагая улучшения и исправления. Например, 12 сентября 2024 года состоялся релиз LibreOffice 24.8.1, который включает в себя обновления и улучшения для текстового процессора LibreOfficeWriter.

Альтернативы LibreOffice

Существует несколько альтернатив LibreOffice. Наибольшей популярностью пользуются следующие решения:

- Apache OpenOffice – бесплатный офисный пакет с открытым исходным кодом, совместимый с форматами Microsoft Office. Его функциональность и интерфейс схожи с LibreOffice. По умолчанию ApacheOpenOffice использует формат OpenDocument, а также дополнительно поддерживает форматы Microsoft Office, такие как .doc .xls и .ppt. Важно отметить, что LibreOffice и Apache OpenOffice происходят от одного исходного проекта, так как LibreOffice является форкомOpenOffice. Несмотря на это, офисные пакеты, о которых идет речь, имеют некоторые различия в функциональности и поддержке.
- OnlyOffice (в прошлом Teamlab) – еще один вариант, предлагающий как облачные, так и десктопные версии. Включает онлайн-редакторы текстовых документов, электронных таблиц, презентаций, форм и PDF-файлов. Более ориентирован на совместную работу и интеграцию с облачными сервисами. Предлагаются как бесплатные, так и платные решения.
- Google Docs – бесплатное облачное решение, которое не только предоставляет пользователю большую часть возможностей Word и Excel, но и обладает своим уникальным функционалом. Например, интеграцией с другими сервисами Google, групповым чатом и режимом совместного редактирования. Наверняка вы уже неоднократно пользовались этим сервисом и имеете представление о его возможностях.

Каждое из этих решений имеет свои сильные стороны и особенности, так что выбор подходящего варианта зависит от ваших конкретных потребностей и предпочтений.

5. Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации по модулю

Промежуточная аттестация проводится в форме тестирования по каждой теме модуля. Критерии оценки уровня освоения темы:

- оценки «зачтено» заслуживает обучающийся, ответивший верно на 2 и более из 3 вопросов теста;
- оценки «не зачтено» заслуживает обучающийся, ответивший верно менее, чем на 2 из 3 вопросов теста.

Критерии оценки уровня освоения модуля:

- все тесты по темам сданы на оценку «зачтено».

Пример теста для промежуточной аттестации

1. Какие форматы рекомендуется использовать для сохранения документов в LibreOfficeWriter для совместимости с Microsoft Word?
   - Только .docx
   - .docx и .doc
   - .xlsx и .xls
   - Только .doc

2. Какую проверку следует произвести, чтобы убедиться, что документ, сохраненный в LibreOfficeWriter, правильно отображается в Microsoft Word?
   - Проверить наличие обновлений для LibreOffice
   - Убедиться, что документ открывается в Microsoft Word
   - Открыть документ в Microsoft Word и использовать опцию «Проверка совместимости»
   - Переместить документ на другой компьютер

3. Какой формат документов не поддерживается LibreOffice Writer?
   - .pdf
   - .doc
   - .odt
   - .docx

6. Учебно-методическое и информационное обеспечение модуля

1. ArchWiki [Электронный ресурс] URL:https://wiki.archlinux.org
2. Linux по-русски [Электронный ресурс] URL: https://rus-linux.net
3. Официальный сайт проекта LibreOffice [Электронный ресурс] URL: https://www.libreoffice.org/

Приложение № 8

РАБОЧАЯ ПРОГРАММА МОДУЛЯ 8
(Работа с печатными устройствами в Ubuntu 24.04)

По дополнительной общеобразовательной программе –
дополнительной общеразвивающей программе
«Системный администратор Linux»

1. Цель и задачи модуля

Целью освоения модуля «Работа с печатными устройствами в Ubuntu 24.04» является приобретение знаний и навыков, необходимых для эффективного управления печатью, установки и настройки принтеров и сканеров, а также решения распространенных проблем, связанных с печатью в операционной системе Ubuntu 24.04.

Задачи:

1. Изучить основы работы с принтерами и сканерами в операционной системе Ubuntu 24.04.
2. Освоить процесс установки и настройки драйверов для различных моделей принтеров и сканеров, включая использование встроенных средств Ubuntu 24.04 и стороннего ПО.
3. Познакомиться с интерфейсом управления печатью в Ubuntu 24.04, включая работу с очередями печати, настройку параметров печати и решение распространенных проблем.
4. Научиться использовать ПО для сканирования для сканирования документов и их последующей обработки.
5. Познакомиться с дополнительными утилитами и инструментами, которые могут улучшить взаимодействие с печатными и сканирующими устройствами, такими как CUPS и hplip для HP-принтеров.

2. Планируемые результаты освоения модуля

Знать	Уметь
1. Основы работы с принтерами и сканерами в операционной системе Ubuntu 24.04, включая ключевые термины. 2. Основные утилиты и приложения для работы с печатными устройствами в Ubuntu 24.04. 3. Принципы сетевой печати и настройки сетевых принтеров.	1. Устанавливать драйверы для принтеров и сканеров. 2. Конфигурировать принтеры и сканеры в системных настройках Ubuntu 24.04. 3. Производить печать и сканирование документов в Ubuntu 24.04. 4. Решать возможные проблемы и методы их устранения при работе с принтерами и сканерами.

3. Объем модуля

Вид учебной работы	Количество часов
Лекции	2
Самостоятельная работа	1
Форма контроля	тест
Общая трудоемкость по учебному плану	3

4. Структура и содержание модуля

4.1. Структура модуля

№ п/п	Название темы модуля	Количество часов			Форма контроля
		Всего	Лекции	Самостоятельная работа	
1.	Работа с печатными устройствами в Ubuntu 24.04	3	2	1	тесты

4.2. Содержание модуля

Номер темы	Содержание темы
1	Настройка сетевого принтера и сканера в Ubuntu 24.04 Где скачать необходимые драйвера?

Содержание лекции

Где скачать необходимые драйвера?

Основные блоки
Установка драйверов через стандартные репозитории Ubuntu
Установка LibreOffice Writer в Ubuntu 24.04
HP
Brother
Установка драйверов через стандартные репозитории Ubuntu
Установка драйверов с сайта производителя
Проверка совместимости

В этом уроке мы поговорим про поиск и установку необходимых драйверов для сетевых принтеров и сканеров. Начнем с принтеров.

Установка драйверов через стандартные репозитории Ubuntu

В репозиториях Ubuntu можно найти драйвера для широкого спектра принтеров. Например, следующая команда установит драйвера для поддержки большинства принтеров:

sudoaptinstallprinter-driver-gutenprint -y

Установка драйверов с сайта производителя

Если решение, предложенное выше, не помогло, стоит проверить наличие драйверов на сайте производителя. Именитые производители, такие как HP и Brother, нередко предоставляют свои собственные драйвера и утилиты для дистрибутивов Linux. Обычно на странице поддержки можно найти список поддерживаемых операционных систем.

HP

Необходимые драйвера можно найти на официальном сайте производителя. Кроме того, для принтеров HP предлагается программа HPLIP, которая предоставляет драйвера и утилиты. Установить ее можно командой:

sudoaptinstallhplip -y

Brother

Brother предоставляет собственные пакеты драйверов для Ubuntu. Они доступны на сайте производителя.

1. Перейдите на официальный сайт Brother

2. Найдите вашу модель принтера (или сканера) и загрузите соответствующий драйвер для ОС Linux

3. Следуйте инструкциям, которые, как правило, прилагаются вместе с драйверами

После установки драйверов рекомендуется перезагрузить систему. После этого ваш принтер должен быть готов к использованию.

Переходим к сканерам.

Установка драйверов через стандартные репозитории Ubuntu

Зачастую драйвера для сканеров уже включены в стандартные репозитории Ubuntu. Для начала убедитесь, что у вас установлены необходимые пакеты:

sudoaptinstallsanesane-utils -y

Чтобы проверить, обнаружен ли сканер, можно воспользоваться утилитой sane-find-scanner. Если утилита еще не установлена, ее также нужно будет установить:

sudosane-find-scanner -y

Установка драйверов с сайта производителя

Если сканер требует специальных драйверов, которые не входят в пакет sane, имеет смысл посетить сайт производителя сканера. В разделе поддержки или загрузок могут присутствовать необходимые драйвера.

После установки драйверов рекомендуется перезагрузить систему.

Проверка совместимости

Если драйвера по умолчанию или от производителя не помогают решить проблему, возможно, сканер не поддерживается. В этом случае имеет смысл проверить базы данных поддерживаемых устройств, такие как SANE Supported Devices.

5. Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации по модулю

Промежуточная аттестация проводится в форме тестирования по каждой теме модуля. Критерии оценки уровня освоения темы:

- оценки «зачтено» заслуживает обучающийся, ответивший верно на 2 и более из 3 вопросов теста;
- оценки «не зачтено» заслуживает обучающийся, ответивший верно менее, чем на 2 из 3 вопросов теста.

Критерии оценки уровня освоения модуля:

- все тесты по темам сданы на оценку «зачтено».

Пример теста для промежуточной аттестации

1. Какой пакет обеспечивает драйвера для поддержки большинства принтеров в Ubuntu через стандартные репозитории?
   - hplip
   - printer-driver-gutenprint
   - xsane
   - sane

2. Что рекомендуется сделать в качестве следующего шага после установки драйверов?
   - Обновить информацию об актуальных версиях доступных пакетов ПО
   - Создать точку восстановления системы
   - Перезагрузить систему
   - Установить дополнительные пакеты

3. Что следует делать, если после установки драйверов принтер или сканер по-прежнему не работает?
   - Проверить подключение устройства
   - Обратиться в техническую поддержку Canonical
   - Отказаться от использования принтера и сканера на Linux или рассмотреть переход на Windows
   - Убедиться, что драйвера установлены правильно и соответствуют модели устройства

6. Учебно-методическое и информационное обеспечение модуля

1. ArchWiki [Электронный ресурс] URL:https://wiki.archlinux.org
2. Официальный сайт Brother [Электронный ресурс] URL: https://www.brother.com/apps/ps/ru/
3. SANE Supported Devices [Электронныйресурс] URL: http://www.sane-project.org/sane-supported-devices.html