

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ»

«УТВЕРЖДАЮ»

Проректор по учебной работе  Г.Ю. Нагорная

«16» 01 2026 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Введение в операционную систему UNIX

Уровень образовательной программы бакалавриат

Направление подготовки 01.03.02 Прикладная математика и информатика

Направленность(профиль) «Математические и информационные системы и технологии в астрономии»

Форма обучения: очная

Срок освоения ОП 4 года

Институт Цифровых технологий

Кафедра разработчик РПД Астрофизика

Выпускающая кафедра Астрофизика

Начальник  
учебно-методического управления

 Семенова Л. У.

Директор института ЦТ

 Кумратова А. М.

И. О. заведующего выпускающей кафедрой

 Валявин Г. Г.

г. Черкесск, 2026 г

## СОДЕРЖАНИЕ

- 1. Цели освоения дисциплины**
  - 2. Место дисциплины в структуре образовательной программы**
  - 3. Планируемые результаты обучения по дисциплине**
  - 4. Структура и содержание дисциплины**
    - 4.1. Объем дисциплины и виды учебной работы
    - 4.2. Содержание дисциплины
      - 4.2.1. Разделы (темы) дисциплины, виды учебной деятельности и формы контроля
      - 4.2.2. Лекционный курс
      - 4.2.3. Практические занятия
    - 4.3. Самостоятельная работа обучающегося
  - 5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине**
  - 6. Образовательные технологии**
  - 7. Учебно-методическое и информационное обеспечение дисциплины**
    - 7.1. Перечень основной и дополнительной учебной литературы
    - 7.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»
    - 7.3. Информационные технологии, лицензионное программное обеспечение
  - 8. Материально-техническое обеспечение дисциплины**
    - 8.1. Требования к аудиториям (помещениям, местам) для проведения занятий
    - 8.2. Требования к оборудованию рабочих мест преподавателя и обучающихся
    - 8.3. Требования к специализированному оборудованию
  - 9. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья**
- Приложение 1. Фонд оценочных средств**

## 1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины «Введение в операционную систему Linux» является дать обучающимся систематизированные сведения о принципах построения, проектирования и функционирования современных операционных систем (ОС), а также изучение основ работы с UNIX-подобными операционными системами. Дисциплина охватывает основы интерфейса командной строки, применение стандартных утилит, знакомство с текстовыми редакторами, основы администрирования.

### Основные задачи курса:

2. знакомство с различными операционными системами,
3. знакомство с особенностями построения и функционирования семейства операционных систем Linux;
  1. получение практических навыков работы с ОС;
  2. изучение учетных записей, параметров рабочей среды пользователя;
  3. управление дисками и файловыми системами;
  4. получение базовых навыков работы в системном программировании, а именно: работа в командной строке с интерпретатором bash и написание bash-скриптов.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

2.1. Учебная дисциплина «Введение в операционную систему Linux» относится к обязательной части, Блока 1. Дисциплины (модули), имеет тесную связь с другими дисциплинами.

2.2. В таблице приведены предшествующие и последующие дисциплины, направленные на формирование компетенций дисциплины в соответствии с матрицей компетенций ОП.

### Предшествующие и последующие дисциплины, направленные на формирование компетенций

№ п/п	Предшествующие дисциплины	Последующие дисциплины
1.	Основы информатики	Базы данных в операционной системе UNIX
2.	Архитектура ЭВМ	Основы Python в операционной системе UNIX
3.	Операционные системы	

### 3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Планируемые результаты освоения образовательной программы (ОП) – компетенции обучающихся определяются требованиями стандарта по направлению подготовки 01.03.02 Прикладная математика и информатика и формируются в соответствии с матрицей компетенций ОП

<b>№ п/п</b>	<b>Номер/индекс компетенции</b>	<b>Наименование компетенции (или ее части)</b>	<b>В результате изучения учебной дисциплины обучающиеся должны:</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
1	ПК-2	Способен применять современные информационные и коммуникационные сервисы и программные комплексы в различных сферах деятельности	ПК-2.1 Работает с современными информационными и коммуникационными сервисами при создании программных комплексов ПК-2.2Способен программировать на современных прикладных платформах, настраивать и тестировать создаваемые программные комплексы

#### 4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

##### 4.1. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Вид учебной работы	Всего часов	Семестры	
		№2	
		Часов	
1	2	2	
<b>Аудиторная контактная работа (всего)</b>	<b>20</b>	<b>20</b>	
В том числе:			
Лекции (Л)	10	10	
Практические занятия (ПЗ), Семинары (С)	10	10	
<b>Контактная внеаудиторная работа, в том числе:</b>	<b>1,7</b>	<b>1,7</b>	
Групповые и индивидуальные консультации	1,7	1,7	
<b>Самостоятельная работа обучающегося (СРО) (всего)</b>	<b>50</b>	<b>50</b>	
<i>Работа с электронными источниками</i>	8	8	
<i>Подготовка к практическим занятиям</i>	8	8	
<i>Подготовка к текущему тестовому контролю</i>	8	8	
<i>Подготовка к промежуточному контролю/ контрольные вопросы</i>	10	10	
<i>Подготовка к коллоквиуму</i>	8	8	
<i>Работа с книжными источниками</i>	8	8	
<b>Промежуточная аттестация</b>	Зачет	<b>3</b>	<b>3</b>
	Прием зач., час.	<b>0,3</b>	<b>0,3</b>
<b>ИТОГО: Общая трудоемкость</b>	<b>Часов</b>	<b>72</b>	<b>72</b>
	<b>зачетных единиц</b>	<b>2</b>	<b>2</b>

## 4.2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 4.2.1 Разделы (темы) дисциплины, виды учебной деятельности и формы контроля

№ п/п	Наименование раздела дисциплины	Виды учебной деятельности, включая самостоятельную работу обучающихся (в часах)					Формы текущей и промежуточной аттестации)
		Л	ЛР	ПЗ	СРО	Всего	
1	2	3	4	5	6	7	8
<b>Семестр 2</b>							
1.	Раздел 1. История операционных систем и основы ОС Linux	2		2	10	10	Текущий тестовый контроль, контрольные вопросы коллоквиум, практические работы
2.	Раздел 2. Командная строка в Unix	2		2	10	14	
3.	Раздел 3. Интерпретатор команд bash	6		6	20	32	
4.	Раздел 4. Работа с прикладными программами в ОС Unix.			4	10	14	
	Контактная внеаудиторная работа					1.7	Групповые и индивидуальные консультации
	Промежуточная аттестация					0.3	<b>Зачет</b>
<b>Итого часов во 2 семестре:</b>		<b>10</b>		<b>14</b>	<b>50</b>	<b>72</b>	

#### 4.2.2. Лекционный курс

п/	Наименование раздела (темы) дисциплины	Наименование темы лекции	Содержание лекции	Всего часов
	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>Семестр 2</b>				
	Раздел 1. Основы ОС Unix	Тема 1. Основы ОС Unix.	История UNIX  Стандартная структура каталогов  Авторизация в UNIX, права доступа пользователей.	2
	Раздел 2. Командная строка в Unix	Тема 2. Командная строка в Unix.	Команды echo, whoami, pwd  Утилиты работы с файловой системой  Утилиты обработки текстов.  Перенаправление ввода вывода. Команды echo, cut, grep.  Подстановка переменных и имен файлов в командной строке.  Тестовый редактор nano	2
	Раздел 3. Интерпретатор команд	Тема 2. Интерпретатор команд	Программирование в интерпретаторе команд: команда test; условный оператор if; циклы while, for; команды switch, read	6
	<b>Всего часов во 2 семестре:</b>			<b>10</b>

#### 4.2.3. Лабораторный практикум – не предусмотрен

#### 4.2.4. Практические занятия

11	Наименование раздела дисциплины	Наименование практического занятия	Содержание практического занятия	Всего часов
1	2	3	4	5
<b>Семестр 2</b>				
1.	Раздел 1. Основы ОС Unix	Тема 1. Основы ОС Unix.	Стандартная структура каталогов Авторизация в UNIX, права доступа пользователей.	2
2.	Раздел 2. Командная строка в Unix	Тема 2. Командная строка в Unix.	Команды echo, whoami, pwd Утилиты работы с файловой системой Перенаправление ввода вывода. Использование программ фильтров Подстановка переменных и имен файлов в командной строке. Утилиты обработки текстов Тестовый редактор nano	2
3.	Раздел 3. Интерпретатор команд	Тема 3. Интерпретатор команд Bash.	Программирование в интерпретаторе команд: условный оператор if; циклы while, for; команды switch, read	2
4.	Раздел 4. Работа с прикладными программами в ОС Unix	Тема 5. Прикладные программы в ОС Unix (на примере астрономических прикладных программ).	Установка и удаление программ. Менеджеры пакетов RPM и Yum. Астрономические программные приложения – ALADIN и topcat	4
<b>Всего часов во 2 семестре:</b>				<b>10</b>

#### 4.3. САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ

№ п/п	Наименование раздела дисциплины	№ п/п	Виды СРО	Всего часов
1	2		4	5
<b>Семестр 2</b>				
1.	Раздел 1. Основы ОС Unix	1.1	Проработка лекций - включает чтение конспекта лекций, профессиональной литературы, периодических изданий. Подготовка к практическим занятиям.	10
		1.2	Работа с электронными источниками. Подготовка к практическим занятиям. Подготовка к текущему тестовому контролю. Подготовка к промежуточному контролю/ контрольные вопросы. Подготовка к коллоквиуму. Работа с книжными источниками. Подготовка контрольной работы	
		1.3	Проработка лекций - включает чтение конспекта лекций, профессиональной литературы, периодических изданий. Подготовка к практическим занятиям.	
2.	Раздел 2. Командная строка в Unix	2.1	Проработка лекций - включает чтение конспекта лекций, профессиональной литературы, периодических изданий. Подготовка к практическим занятиям.	10
		2.2	Работа с электронными источниками. Подготовка к практическим занятиям. Подготовка к текущему тестовому контролю. Подготовка к промежуточному контролю/ контрольные вопросы. Подготовка к коллоквиуму.	
		2.3	Работа с книжными источниками. Подготовка контрольной работы	
		2.4	Проработка лекций - включает чтение конспекта лекций, профессиональной литературы, периодических изданий. Подготовка к практическим занятиям.	
		2.5	Проработка лекций - включает чтение конспекта лекций, профессиональной литературы, периодических изданий. Подготовка к практическим занятиям.	
		2.6	Работа с электронными источниками. Подготовка к практическим занятиям. Подготовка к текущему тестовому контролю. Подготовка к промежуточному контролю/ контрольные вопросы. Подготовка к коллоквиуму.	

3.	Раздел 3. Интерпретатор команд	3.1	Проработка лекций - включает чтение конспекта лекций, профессиональной литературы, периодических изданий. Подготовка к практическим занятиям.	20
		3.2	Работа с электронными источниками. Подготовка к практическим занятиям. Подготовка к текущему тестовому контролю. Подготовка к промежуточному контролю/ контрольные вопросы. Подготовка к коллоквиуму. Работа с книжными источниками. Подготовка контрольной работы	
		3.3	Проработка лекций - включает чтение конспекта лекций, профессиональной литературы, периодических изданий. Подготовка к практическим занятиям.	
4	Раздел 4. Прикладные программы в ОС Unix.		Установка и удаление программ. Менеджеры пакетов RPM и Yum. Астрономические программные приложения – ALADIN и torcat.	10
<b>Итого часов во 2 семестре:</b>				<b>50</b>

## **5. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ**

### **5.1. Методические указания для подготовки обучающихся к лекционным занятиям**

Лекция является основной формой обучения в высшем учебном заведении. Записи лекций в конспектах должны быть избирательными, полностью следует записывать только определения. В конспекте рекомендуется применять сокращение слов, что ускоряет запись. Вопросы, возникающие в ходе лекции, рекомендуется записывать на полях и после окончания лекции обратиться за разъяснением к преподавателю.

Работа над конспектом лекции осуществляется по этапам:

- повторить изученный материал по конспекту;
- непонятные положения отметить на полях и уточнить;
- неоконченные фразы, пропущенные слова и другие недочеты в записях устранить, пользуясь материалами из учебника и других источников;
- завершить техническое оформление конспекта (подчеркивания, выделение главного, выделение разделов, подразделов и т.п.).

Самостоятельную работу следует начинать с доработки конспекта, желательно в тот же день, пока время не стерло содержание лекции из памяти. Работа над конспектом не должна заканчиваться с прослушивания лекции. После лекции, в процессе самостоятельной работы, перед тем, как открыть тетрадь с конспектом, полезно мысленно восстановить в памяти содержание лекции, вспомнив ее структуру, основные положения и выводы.

С целью доработки необходимо прочитать записи, восстановить текст в памяти, а также исправить описки, расшифровать не принятые ранее сокращения, заполнить пропущенные места, понять текст, вникнуть в его смысл. Далее прочитать материал по рекомендуемой литературе, разрешая в ходе чтения, возникшие ранее затруднения, вопросы, а также дополнения и исправляя свои записи. Записи должны быть наглядными, для чего следует применять различные способы выделений. В ходе доработки конспекта углубляются, расширяются и закрепляются знания, а также дополняется, исправляется и совершенствуется конспект. Еще лучше, если вы переработаете конспект, дадите его в новой систематизации записей. Это, несомненно, займет некоторое время, но материал вами будет хорошо проработан, а конспективная запись его приведена в удобный для запоминания вид. Введение заголовков, скобок, обобщающих знаков может значительно повысить качество записи. Этому может служить также подчеркивание отдельных мест конспекта красным карандашом, приведение на полях или на обратной стороне листа краткой схемы конспекта и др.

Подготовленный конспект и рекомендуемая литература используется при подготовке к практическому занятию. Подготовка сводится к внимательному прочтению учебного материала, к выводу с карандашом в руках всех утверждений и формул, к решению примеров, задач, к ответам на вопросы, предложенные в конце лекции преподавателем или помещенные в рекомендуемой литературе. Примеры, задачи, вопросы по теме являются средством самоконтроля.

Непременным условием глубокого усвоения учебного материала является знание основ, на которых строится изложение материала. Обычно преподаватель напоминает, какой ранее изученный материал и в какой степени требуется подготовить к очередному занятию. Эта рекомендация, как и требование систематической и серьезной работы над всем лекционным курсом, подлежит безусловному выполнению. Потери логической связи как внутри темы, так и между ними приводит к негативным последствиям: материал учебной дисциплины перестает основательно восприниматься, а творческий труд подменяется утомленным переписыванием. Обращение к ранее изученному материалу не только помогает восстановить в памяти известные положения, выводы, но и приводит

разрозненные знания в систему, углубляет и расширяет их. Каждый возврат к старому материалу позволяет найти в нем что-то новое, переосмыслить его с иных позиций, определить для него наиболее подходящее место в уже имеющейся системе знаний. Неоднократное обращение к пройденному материалу является наиболее рациональной формой приобретения и закрепления знаний. Очень полезным, но, к сожалению, еще мало используемым в практике самостоятельной работы, является предварительное ознакомление с учебным материалом. Даже краткое, беглое знакомство с материалом очередной лекции дает многое. Обучающиеся получают общее представление о ее содержании и структуре, о главных и второстепенных вопросах, о терминах и определениях. Все это облегчает работу на лекции и делает ее целеустремленной.

## **5.2. Методические указания для подготовки обучающихся к практическим занятиям**

Ведущей дидактической целью практических занятий является систематизация и обобщение знаний по изучаемой теме, приобретение практических навыков по тому или другому разделу курса, закрепление полученных теоретических знаний. Практические занятия сопровождают и поддерживают лекционный курс. Подготовка к практикумам носит различный характер, как по содержанию, так и по сложности исполнения.

Многие практические занятия требуют большой исследовательской работы, изучения дополнительной научной литературы. Прежде чем приступить к выполнению такой работы, обучающемуся необходимо ознакомиться обстоятельно с содержанием задания, уяснить его, оценить с точки зрения восприятия и запоминания все составляющие его компоненты. Это очень важно, так как при проработке соответствующего материала по конспекту лекции или по рекомендованной литературе могут встретиться определения, факты, пояснения, которые не относятся непосредственно к заданию. Обучающийся должен хорошо знать и понимать содержание задания, чтобы быстро оценить и отобрать нужное из читаемого. Далее, в соответствии со списком рекомендованной литературы, необходимо отыскать материал к данному заданию по всем пособиям.

Весь подобранный материал нужно хотя бы один раз прочитать или внимательно просмотреть полностью. По ходу чтения помечаются те места, в которых содержится ответ на вопрос, сформулированный в задании. Читая литературу по теме, обучающийся должен мысленно спрашивать себя, на какой вопрос задания отвечает тот или иной абзац прорабатываемого пособия. После того, как материал для ответов подобран, желательно хотя бы мысленно, а лучше всего устно или же письменно, ответить на все вопросы. В случае если обнаружится пробел в знаниях, необходимо вновь обратиться к литературным источникам и проработать соответствующий раздел. Только после того, как преподаватель убедится, что обучающийся хорошо знает необходимый теоретический материал, что его ответы достаточно аргументированы и доказательны, можно считать обучающегося подготовленным к выполнению практических работ.

## **5.3 Методические указания по самостоятельной работе обучающегося**

Самостоятельная работа обучающегося предполагает различные формы индивидуальной учебной деятельности: конспектирование научной литературы, сбор и анализ практического материала в СМИ, проектирование, выполнение тематических и творческих заданий и пр. Выбор форм и видов самостоятельной работы определяется индивидуально-личностным подходом к обучению совместно преподавателем и обучающимся. Формы текущего контроля успеваемости и промежуточной аттестации обучающихся.

Содержание внеаудиторной самостоятельной работы обучающегося по дисциплине «Введение в операционную систему UNIX» включает в себя следующие виды

деятельности:

- работа с электронными источниками;
- подготовка к практическим занятиям;
- подготовка к текущему тестовому контролю;
- подготовка к промежуточному контролю;
- подготовка к коллоквиуму;
- подготовка к контрольной работе;
- работа с книжными источниками.

Самостоятельная работа с электронными источниками, с книжными источниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у обучающихся свое отношение к конкретной проблеме.

Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем по каждой теме семинарского или практического занятия, что позволяет студентам проявить свою индивидуальность в рамках выступления на данных занятиях, выявить широкий спектр мнений по изучаемой проблеме.

Тестирование представляет собой средство контроля усвоения учебного материала темы или раздела дисциплины. При самостоятельной подготовке к тестированию необходимо:

а) проработать информационный материал по дисциплине, проконсультироваться с преподавателем по вопросу выбора учебной литературы;

б) выяснить все условия тестирования заранее, узнать, сколько тестов будет предложено, сколько времени отводится на тестирование, какова система оценки результатов и т.д.

в) приступая к работе с тестами, внимательно и до конца прочитать вопрос и предлагаемые варианты ответов, выбрать правильные (их может быть несколько).

г) в процессе решения желательно применять несколько подходов в решении задания, это позволяет максимально гибко оперировать методами решения, находя каждый раз оптимальный вариант.

д) если встретился чрезвычайно трудный вопрос, не тратить много времени на него, перейти к другим тестам, вернуться к трудному вопросу в конце.

е) обязательно оставить время для проверки ответов, чтобы избежать механических ошибок.

Коллоквиумом называется форма промежуточного контроля знаний обучающегося, которая проводится в виде собеседования преподавателя и обучающегося по самостоятельно подготовленной теме. Целью коллоквиума является формирование у обучающегося навыков анализа теоретических проблем на основе самостоятельного изучения учебной и научной литературы.

Подготовка к коллоквиуму начинается с установочной консультации преподавателя, на которой он разъясняет развернутую тематику проблемы, рекомендует литературу для изучения и объясняет процедуру проведения коллоквиума. При подготовке к коллоквиуму от обучающегося требуется:

– владение изученным в ходе учебного процесса материалом, относящимся к рассматриваемой проблеме;

– знание разных точек зрения, высказанных в экономической литературе по соответствующей проблеме, умение сопоставлять их между собой;

– наличие собственного мнения по обсуждаемым вопросам и умение его аргументировать.

– основательно и подробно изложить, как именно указанная проблема представлена в научной и учебной литературе. Важно показать умение использовать знание теории для

объективного и всестороннего анализа исследуемой проблемы; показать свое отношение к вопросу, который изучается; выделить положительное, определить недостатки, сформулировать наиболее эффективные меры относительно их устранения.

### **Методические рекомендации по подготовке к тестовому контролю**

Тесты - это задания, предусматривающие конкретный, краткий, четкий ответ на имеющиеся эталоны ответов. Готовясь к тестированию, необходимо проработать информационный материал по дисциплине.

Обучающемуся необходимо проконсультироваться с преподавателем по вопросу выбора учебной литературы; четко выясните все условия тестирования заранее.

Приступая к работе с тестами, внимательно и до конца прочтите вопрос и предлагаемые варианты ответов. Выберите правильные. В процессе решения желательно применять несколько подходов в решении задания. Это позволяет максимально гибко оперировать методами решения, находя каждый раз оптимальный вариант.

### **Работа с литературными источниками и интернет ресурсами**

В процессе подготовки к практическим занятиям, обучающимся необходимо обратить особое внимание на самостоятельное изучение рекомендованной учебной и учебно-методической, а также научной литературы.

Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у обучающихся свое отношение к конкретной проблеме.

Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем по каждой теме семинарского или практического занятия, что позволяет обучающимся проявить свою индивидуальность в рамках выступления на данных занятиях, выявить широкий спектр мнений по изучаемой проблеме.

### **Методические рекомендации по подготовке к коллоквиуму**

Коллоквиумом называется собеседование преподавателя и обучающегося по заранее определенным контрольным вопросам. Целью коллоквиума является формирование у обучающегося навыков анализа теоретических проблем на основе самостоятельного изучения учебной и научной литературы. На коллоквиум выносятся крупные, проблемные, нередко спорные теоретические вопросы. Упор делается на монографические работы профессора-автора данного спецкурса. От обучающегося требуется:

- владение изученным в ходе учебного процесса материалом, относящимся к рассматриваемой проблеме;
- знание разных точек зрения, высказанных в научной литературе по соответствующей проблеме, умение сопоставлять их между собой;
- наличие собственного мнения по обсуждаемым вопросам и умение его аргументировать.

Коллоквиум - это не только форма контроля, но и метод углубления, закрепления знаний обучающихся, так как в ходе собеседования преподаватель разъясняет сложные вопросы, возникающие у обучающегося в процессе изучения данного источника. Однако коллоквиум не консультация и не экзамен. Его задача добиться глубокого изучения отобранного материала, пробудить у обучающегося стремление к чтению дополнительной социологической литературы.

Подготовка к коллоквиуму начинается с установочной консультации преподавателя, на которой он разъясняет развернутую тематику проблемы, рекомендует литературу для

изучения и объясняет процедуру проведения коллоквиума. Как правило, на самостоятельную подготовку к коллоквиуму, обучающемуся отводится 3-4 недели. Методические указания состоят из рекомендаций по изучению источников и литературы, вопросов для самопроверки и кратких конспектов ответа с перечислением основных фактов и событий, относящихся к пунктам плана каждой темы. Это должно помочь обучающегося целенаправленно организовать работу по овладению материалом и его запоминанию. При подготовке к коллоквиуму следует, прежде всего, просмотреть конспекты лекций и практических занятий и отметить в них имеющиеся вопросы коллоквиума. Если какие-то вопросы вынесены преподавателем на самостоятельное изучение, следует обратиться к учебной литературе, рекомендованной преподавателем в качестве источника сведений.

Коллоквиум проводится в форме индивидуальной беседы преподавателя с каждым обучающимся или беседы в небольших группах (2-3 человека). Обычно преподаватель задает несколько кратких конкретных вопросов, позволяющих выяснить степень добросовестности работы с литературой, проверяет конспект. Далее более подробно обсуждается какая-либо сторона проблемы, что позволяет оценить уровень понимания. По итогам коллоквиума выставляется дифференцированная оценка по пятибалльной системе.

### **Промежуточная аттестация**

По итогам 2 семестра проводится зачет. При подготовке к сдаче зачета рекомендуется пользоваться материалами практических занятий и материалов, изученными в ходе текущей самостоятельной работы.

По итогам обучения проводится зачет, к которому допускаются обучающиеся, имеющие положительные результаты по защите практических работ.

Зачет проводится в устной форме, включает подготовку и ответы обучающегося на теоретические вопросы. По итогу выставляется зачет.

## 6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

№ п/п	№ семестра	Виды учебной работы	Образовательные технологии	Всего часов
1	2	3	4	4
1.	2	<i>Лекция</i> «Основы ОС Unix»	Презентация, дистанционные технологии, технология организации самостоятельной работы обучающихся с использованием электронных образовательных ресурсов и электронных образовательных изданий в Moodle	2
2.	2	<i>Лекция</i> «Командная строка в Unix»	Презентация, технология образовательных платформ, технологии организации самостоятельной работы обучающихся с использованием электронных образовательных ресурсов и электронных образовательных изданий в Moodle	2
3.	2	<i>Лекция</i> «Интерпретатор команд»	Презентация, мультимедийные технологии	2
4.	2	<i>Практическая работа</i> «Основы ОС Unix»	Презентация, игровые технологии, технологии организации самостоятельной работы обучающихся с использованием электронных образовательных ресурсов и электронных образовательных изданий в Moodle	2
5.	2	<i>Практическая работа</i> «Командная строка в Unix»	Диалоговые технологии, технология проектного обучения технологии организации самостоятельной работы обучающихся с использованием электронных образовательных ресурсов и электронных образовательных изданий в Moodle	2
6.	2	<i>Практическая работа</i> «Интерпретатор команд»	Технология проектного обучения, игровые технологии, технологии организации самостоятельной работы обучающихся с использованием электронных образовательных ресурсов и электронных образовательных изданий в Moodle	2
7.	2	<i>Практическая работа</i> Раздел 4. Прикладные программы в ОС Unix.	Диалоговые технологии, технология проектного обучения технологии организации самостоятельной работы обучающихся с использованием электронных образовательных ресурсов и электронных образовательных изданий в Moodle	2
<b>Итого часов во 2 семестре:</b>				14

## 7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 7.1. Перечень основной и дополнительной учебной литературы

#### Основная литература

1. Назаров, С. В. Современные операционные системы : учебное пособие / С. В. Назаров, А. И. Широков. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 351 с. — ISBN 978-5-4497-0385-9. —

Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/89474.html>

2. Операционные системы : учебное пособие к проведению исследовательских лабораторных работ / составители Е. О. Ткачук. — Ростов-на-Дону : Северо-Кавказский филиал Московского технического университета связи и информатики, 2018. — 127 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/89518.html>

3. Кобылянский, В. Г. Операционные системы, среды и оболочки : учебное пособие / В. Г. Кобылянский. — Новосибирск : Новосибирский государственный технический университет, 2018. — 80 с. — ISBN 978-5-7782-3517-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/91285.html>

4. Операционные системы : учебное пособие для бакалавров / составители И. В. Винокуров.

— Москва : Ай Пи Ар Медиа, 2022. — 133 с. — ISBN 978-5-4497-1406-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/115696.html>

5. Замятин, А. В. Операционные системы : учебное пособие / А. В. Замятин, С. П. Сущенко.

— Томск : Издательство Томского государственного университета, 2020. — 220 с. — ISBN 978-5-94621-935-8. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/116810.html>

6. Куль, Т. П. Операционные системы : учебное пособие / Т. П. Куль. — Минск : Республиканский институт профессионального образования (РИПО), 2019. — 311 с. — ISBN 978-985-503-940-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/93431.html>

7. Сафонов, В. О. Основы современных операционных систем : учебное пособие / В. О. Сафонов. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 826 с. — ISBN 978-5-4497-0552-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/94855.html>

#### Дополнительная литература

1. Мезенцева, Е. М. Операционные системы : лабораторный практикум / Е. М. Мезенцева, О. С. Коняева, С. В. Малахов. — Самара : Поволжский государственный университет телекоммуникаций и информатики, 2017. — 214 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/75395.html>

2. Басыня, Е. А. Операционные системы : учебно-методическое пособие / Е. А. Басыня, А. В. Сафронов. — Новосибирск : Новосибирский государственный технический университет, 2016. — 84 с. — ISBN 978-5-7782-3106-1. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/91630.html>

3. Попов, А. А. Операционные системы : лабораторный практикум / А. А. Попов, П. С. Шаталов, М. А. Масюк ; под редакцией Г. А. Доррер. — Красноярск : Сибирский государственный

университет науки и технологий имени академика М.Ф. Решетнева, 2020. — 80 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/107209.html>

4. Безсмертный, Б. В. Основы администрирования в операционной системе FreeBSD (UNIX): учебное пособие / Б. В. Безсмертный, А. Ф. Задорожный. — Новосибирск : Новосибирский государственный архитектурно-строительный университет (Сибстрин), ЭБС АСВ, 2019. — 89 с. — ISBN 978-5-7795-0877-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/107642.html>

5. Пиляй, А. И. Базы данных и операционные системы : учебно-методическое пособие / А. И. Пиляй, А. М. Якубович. — Москва : МИСИ-МГСУ, ЭБС АСВ, 2021. — 46 с. — ISBN 978-5-7264-2951-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/122820.html>

6. Немет, Немет Э., Снайдер, Снайдер Г., Хейн, Хейн Т., Мак-Гинли, Мак-Гинли Л., Вэйли, Вэйли Б., Волкова, Я. П., Иваненко, Д. Я., Корниенко, Ю. И., Артеменко, Ю. И.; Руководство администратора LINUX; Вильямс, Москва ; Санкт-Петербург ; Киев; 2007 (2 экз.)

7. Попов, А. А. Операционные системы : лабораторный практикум / А. А. Попов, П. С. Шаталов, М. А. Масюк ; под редакцией Г. А. Доррер. — Красноярск : Сибирский.

8. Робачевский, Андрей М., А. М.; Операционная система UNIX : Учеб. пособие для студентов вузов.; БХВ-Петербург, Дюссельдорф; Киев; Москва; СПб.; 2002 (9 экз.)

## 7.2 Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

<http://window.edu.ru> - Единое окно доступа к образовательным ресурсам;  
<http://fcior.edu.ru> - Федеральный центр информационно-образовательных ресурсов;  
<http://elibrary.ru> - Научная электронная библиотека.

Электронно-библиотечная система IPRbooks URL: [http:// https://ipr-smart.ru/61466.html](http://https://ipr-smart.ru/61466.html)/ ООО «Ай Пи Эр Медиа».

## 7.3. Информационные технологии, лицензионное программное обеспечение

Лицензионное программное обеспечение	Реквизиты лицензий/ договоров
Антивирус Dr.Web Desktop Security Suite	Лицензионный договор № 621 Срок действия: с 25.09.2025 до 24.09.2026
Консультант Плюс	Договор № 7 от 15.01.2026 г.
Цифровой образовательный ресурс IPR SMART	Лицензионный договор № 12873/25П от 02.07.2025 г. Срок действия: с 01.07.2025 г. до 30.06.2026 г.
Бесплатное ПО	
АстроСофт, LibreOffice, OpenOffice, МойОфис, Visual Studio Community, Sumatra PDF, 7-Zip, Adobe Acrobat Reader, MySQL, Virtual box, PostgreSQL, РЕДОС 8.0, графический редактор GIMP, TOPCAT, ALADIN, MIDAS, IRAF, ds9, Opera, Cromium, Yandex.	

## **8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

### **8.1. Требования к аудиториям (помещениям, местам) для проведения занятий**

**Учебная аудитория для проведения занятий лекционного типа**

**Специализированная мебель:** Кафедра - 1шт., доска меловая - 1шт., парты - 27шт., стулья - 61шт., Технические средства обучения, служащие для предоставления учебной информации большой аудитории: Проектор - 1шт. Экран моторизованный - 1шт.

**Учебная аудитория для проведения занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации**

**Специализированная мебель:**

Специализированная мебель: Кафедра - 1шт., доска меловая - 1шт., парты - 27шт., стулья - 61шт., Технические средства обучения, служащие для предоставления учебной информации большой аудитории: Проектор - 1шт. Экран моторизованный - 1шт.

**Учебная аудитория для проведения занятий лекционного и лабораторного типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации**

**Специализированная мебель:**

Компьютерные столы - 13шт., стулья - 20шт., книжный шкаф - 1шт., доска маркерная - 1шт.; Лабораторное оборудование, технические средства обучения, служащие для предоставления учебной информации большой аудитории: Персональный компьютер (сервер) – 1шт. Персональный компьютер (учебный) – 10шт. Доска интерактивная/экран - 1шт. Проектор – 1 шт.

### **Отдел обслуживания электронными изданиями**

**Специализированная мебель:**

Рабочие столы на 1 место – 24 шт. Стулья – 24 шт.

Набор демонстрационного оборудования и учебно-наглядных пособий, обеспечивающих тематические иллюстрации:

Интерактивная система

- 1 шт. Монитор – 21 шт. Сетевой терминал -18 шт. Персональный компьютер

-3 шт. МФУ – 2 шт.

Принтер –1шт.

Помещение для хранения и профилактического обслуживания учебного оборудования Ауд. 11 Специализированная мебель:

Стеллажи – 1 шт., шкаф – 1 шт., стул -1 шт., кресло компьютерное – 4 шт., стол – 5 шт.

Профилактическое оборудование:

Перфоратор -1 шт.

Аккумуляторная дрель-шуруповерт – 1 шт., наборы отверток -2 шт., пылесос -1 шт., клещи обжимные – 3 шт., тестер блоков питания -1 шт., мультиметр -1 шт., фен термовоздушный паяльный -1 шт., паяльник -3 шт.

Учебное пособие (персональный компьютер в комплекте) – 2 шт., пассатижи – 1 шт., бокорезы- 1 шт.

Коммутатор -1 шт., внешний DVD привод -1 шт., внешний жесткий диск - 1 шт.

### **8.2. Требования к оборудованию рабочих мест преподавателя и обучающихся**

1. Рабочее место преподавателя, оснащенное компьютером с доступом в Интернет.
2. Рабочие места обучающихся, оснащенные компьютерами с доступом в Интернет, предназначенные для работы в электронной образовательной среде.

### **8.3. Требования к специализированному оборудованию**

- нет

## **9. ОСОБЕННОСТИ РЕАЛИЗАЦИИ ДИСЦИПЛИНЫ ДЛЯ ИНВАЛИДОВ И ЛИЦ СОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ**

Для обеспечения образования инвалидов и обучающихся с ограниченными возможностями здоровья разрабатывается (в случае необходимости) адаптированная образовательная программа, индивидуальный учебный план с учетом особенностей их психофизического развития и состояния здоровья, в частности применяется индивидуальный подход к освоению дисциплины, индивидуальные задания: рефераты, письменные работы и, наоборот, только устные ответы и диалоги, индивидуальные консультации, использование диктофона и других записывающих средств для воспроизведения лекционного и семинарского материала.

В целях обеспечения обучающихся инвалидов и лиц с ограниченными возможностями здоровья комплектуется фонд основной учебной литературой, адаптированной к ограничению электронных образовательных ресурсов, доступ к которым организован в БИЦ Академии. В библиотеке проводятся индивидуальные консультации для данной категории пользователей, оказывается помощь в регистрации и использовании сетевых и локальных электронных образовательных ресурсов, предоставляются места в читальном зале.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ  
«Введение в операционную систему Linux»**

**1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ**  
**«Введение в операционную систему Linux»**

**1. Компетенции, формируемые в процессе изучения дисциплины**

Индекс	Формулировка компетенции
ПК-2	Способен применять современные информационные и коммуникационные сервисы и программные комплексы в различных сферах деятельности

**2. Этапы формирования компетенции в процессе освоения дисциплины**

Основными этапами формирования указанных компетенций, при изучении обучающимися дисциплины, являются последовательное изучение содержательно связанных между собой разделов (тем) учебных занятий. Изучение каждого раздела (темы) предполагает овладение обучающимися необходимыми компетенциями. Результат аттестации обучающихся на различных этапах формирования компетенций показывает уровень освоения ими компетенций.

Поэтапное формирование компетенций прямо связано с местом дисциплины в образовательной программе.

Разделы (темы) дисциплины	Формируемые компетенции (коды)
	ПК-2
Основы ОС Unix	+
Командная строка в Unix	+
Интерпретатор команд	+

### 3. Показатели, критерии и средства оценивания компетенций, формируемых в процессе изучения дисциплины

**ПК-2** - Способен применять современные информационные и коммуникационные сервисы и программные комплексы в различных сферах деятельности

Индикаторы достижения компетенции	Критерии оценивания результатов обучения				Средства оценивания результатов обучения	
	неудовлетворительно	удовлетворительно	хорошо	отлично	Текущий контроль	Промежуточная аттестация
ПК-2.1 Работает с современными информационными и коммуникационными сервисами при создании программных комплексов	Не работает с современными информационными и коммуникационными сервисами при создании программных комплексов	Частично работает с современными информационными и коммуникационными сервисами при создании программных комплексов	Хорошо работает с современными информационными и коммуникационными сервисами при создании программных комплексов	Отлично работает с современными информационными и коммуникационными сервисами при создании программных комплексов	Коллоквиум, контрольные вопросы, тестирование, индивидуальные задания к лабораторным работам.	Зачет
ПК-2.2 Способен программировать на современных прикладных платформах, настраивать и тестировать создаваемые программные комплексы	Не способен программировать на современных прикладных платформах, настраивать и тестировать создаваемые программные комплексы	Частично способен программировать на современных прикладных платформах, настраивать и тестировать создаваемые программные комплексы	Способен программировать на современных прикладных платформах, настраивать и тестировать создаваемые программные комплексы	Отлично способен программировать на современных прикладных платформах, настраивать и тестировать создаваемые программные комплексы	Коллоквиум, контрольные вопросы, тестирование, индивидуальные задания к лабораторным работам	Зачет

#### 4. Комплект контрольно-оценочных средств по дисциплине

##### «Введение в операционную систему LINUX»

##### Вопросы к зачету

##### по дисциплине «Введение в операционную систему LINUX»

1. Каково назначение операционной системы? Почему говорят об операционной системе как о виртуальной машине? Какими ресурсами и как управляет операционная система?
2. Из каких частей состоит ядро UNIX?
3. Из каких этапов состоит загрузка операционной системы UNIX?
4. Архитектура операционной системы: что такое ядро и прикладные программы? Чем отличаются монолитные и микроядерные системы?
5. Какие можно выделить классы операционных систем? В чём заключаются их отличия? В чём заключается особенность архитектуры ОС UNIX?
6. Какие способы установки программных продуктов имеются в ОС UNIX? В чём их принципиальное различие. Приведите пример использования каждого из способов.
7. Опишите применение различных команд поиска (файлов, директорий, элемента текста и т.д.) известные в ОС UNIX. Приведите примеры.
8. Что такое терминал? Какие бывают терминалы?
9. Что такое конфликты в системах с пакетами и как они могут разрешаться? Какова роль репозитория пакетов?
10. Какие существуют инструменты и способы для проверки обновлений установленного программного обеспечения и последующего его обновления?
11. Какие средства разработки приложений и сценариев существуют в ОС UNIX?
12. Назовите типы файлов в ОС UNIX. В чём все они схожи, каковы отличия между ними?
13. Командная строка ОС UNIX: из каких частей состоит, основные управляющие клавиши, примеры команд?
14. Что такое системные службы? Как организованы системные службы в ОС UNIX-системах, наследующих схему загрузки UNIX System V?
15. Какими отличительными особенностями обладает файловая система ОС UNIX?
16. Интерпретатор `bash` - автодополнение, история команд. Специальные символы сокращения `~`, `~-`, `~+` .
17. Понятие стандартного ввода/вывода.
18. Перенаправление стандартного ввода/вывода из программы в файл, из файла в программу, из программы в программу.
19. Интерпретатор команд `bash` как среда программирования.
20. Условия и циклы в `bash`.
21. Код завершения программы. Программа `test` как логический оператор в среде `bash`.
22. Файлы `passwd` и `group`. Имя пользователя его уникальный номер.
23. Домашний каталог пользователя в UNIX.
24. Подстановка результата работы программы как параметра.
25. Переменные среды, команда `export`.
26. Команды работы с текстом `cat`, `head`, `tail`, `grep`, `more`, `less`.
27. Права пользователя на файлы. Различия в интерпретации флагов чтения и исполнения для файлов и каталогов.
28. Команды изменения прав пользователя на файлы: `chmod`, `chgrp` (`chown`).
29. Программа `dd` - копирование образа диска. Программы `tar` и `gz` – архивирование.
30. Редактор текста `nano`
31. Прикладные программы - калькулятор `bc`.
32. Поиск файлов на диске - команда `find`.

## Контрольные вопросы

### по дисциплине «Введение в операционную систему Linux»

Некоторые контрольные вопросы по дисциплине «Введение в операционную систему Linux»:

1. Каково назначение операционной системы?
2. Какими ресурсами и как управляет операционная система?
3. Архитектура операционной системы: что такое ядро и прикладные программы?

Чем отличаются монолитные и микроядерные системы?

4. В чём заключается особенность архитектуры UNIX?
5. Какие программы называют демонами? Приведите примеры.
6. Из каких частей состоит ядро UNIX?
7. Какими отличительными особенностями обладает файловая система UNIX?
8. Назовите типы файлов в UNIX. В чём все они схожи, каковы отличия между ними?
9. Чем жёсткие ссылки отличаются от символических? Какими преимуществами обладают символические ссылки?
10. Какие каталоги стандартизованы в UNIX? Что обычно хранится в каждом из них?
11. Какие существуют наиболее распространённые схемы доступа? В чём заключаются основные отличия между ними? Какая схема доступа используется в UNIX?
12. Какие существуют права доступа в UNIX? Какие из них являются специфичными для простых файлов, а какие для директорий?

## Вопросы для коллоквиума

### по дисциплине «Введение в операционную систему Linux»

1. Что такое ядро операционной системы Linux? Ядро — основная программа, которая управляет ресурсами компьютера, распределяя их между разными пользователями и выполняя различные задачи.
2. Что такое оболочка? Оболочка выступает в качестве интерфейса между пользователем и системой, принимает команды и настраивает их для выполнения.
3. В чём разница между системным вызовом, библиотечной функцией и командой Linux? Системный вызов — часть программирования ядра, библиотечная функция — программа, не являющаяся частью ядра, но доступная пользователям системы, команды UNIX — автономные программы, которые могут включать в свои программы как системные вызовы, так и библиотечные функции.
4. Что такое суперпользователь? Суперпользователь — пользователь особого типа, который имеет открытый доступ ко всем файлам и командам в системе.
5. Как определяется и устанавливается путь в UNIX? Каждый раз, когда вводится команда, переменная с именем PATH или путь определяет, в каком каталоге оболочка будет искать эту команду. Можно указать путь вручную с помощью команды «set path = [путь к каталогу]».
6. Можно ли видеть информацию о процессе во время его выполнения? Каждый процесс однозначно идентифицируется идентификатором процесса, с помощью команды ps можно просмотреть сведения и статус процесса.
7. Как узнать имя (путь) текущего каталога в UNIX? Для этого используется команда pwd.
8. Как узнать, какие файлы находятся в текущем каталоге? Для этого используется команда ls.
9. Как просмотреть список текущих процессов с помощью команды ps и в интерактивной форме top?
10. Как посылать сигналы процессам по их названию, уникальному номеру и т. д.?
11. Как приостанавливать работу процессов с помощью командного интерпретатора Bash?
12. Как восстанавливать работу процесса в фоновом и в основном процессах?

**Тестовые вопросы**  
**по дисциплине «Введение в операционную систему Linux»**

**Вопрос №1** Операционная система – это:

- 1) Часть программного обеспечения, выступающая в качестве интерфейса между приложениями (и пользователями) и аппаратурой компьютера.
- 2) Программный продукт предназначенный для работы в интернете
- 3) Совокупность программного и аппаратного обеспечения компьютера

**Вопрос №2** Ядро операционной системы – это \_\_\_\_\_ ?

**Вопрос №4** Операционная система выполняет следующие функции:

- 1) Предоставление пользователю вместо реальной аппаратуры компьютера некой расширенной машины. Повышение эффективности использования компьютера.
- 2) Управление внешней памятью компьютера и управление внутренними ресурсами
- 3) Управление программными приложениями компьютера

**Вопрос № 5** Ресурсы вычислительной системы можно разбить на :

- 1) Первичные – аппаратные ресурсы и вторичные – логические, программные и информационные ресурсы.
- 2) Программные приложения и аппаратные средства
- 3) Файловую системы и драйверы

**Вопрос №5** Традиционная архитектура операционной системы включает в себя:

- 1) Монолитное ядро, реализующее основные функции по управлению аппаратными ресурсами и организующее среду для выполнения пользовательских процессов
- 2) Монолитное ядро, реализующее основные функции по управлению внешними носителями
- 3) Ядро, реализующее основные функции по управлению главными ресурсами поддержки интернет

**Вопрос №6** Микроядерная архитектура.

- 1) распределяет функции ОС между микроядром и входящими в состав ОС системными сервисами, реализованными в виде процессов, равноправных с пользовательскими приложениями.
- 2) распределяет функции ОС между компьютером и устройством ввода – вывода
- 3) распределяет функции ОС между внешними носителями и внутренними

**Вопрос №7** Мультипрограммные ОС

- 1) обеспечивающие одновременную работу нескольких пользователей
- 2) поддерживающие сетевую работу компьютеров
- 3) обеспечивающие запуск одновременно нескольких программ

**Вопрос №8** Особенности современных универсальных операционных систем

- 1) поддержка многозадачности
- 2) поддержка сетевых функций

- 3) обеспечение безопасности и защиты данных
- 4) предоставление большого набора системных функций разработчикам приложений

**Вопрос №9** Утверждения относительно понятия «API-функция» являющиеся правильными

- 1) API-функции определяют прикладной программный интерфейс
- 2) API-функции используются при разработке приложений для доступа к ресурсам компьютера
- 3) API-функции реализуют самый нижний уровень ядра системы
- 4) API-функции — это набор аппаратно реализованных функций системы

**Вопрос №10** Особенности характерные для ОС **Linux**

- 1) открытость и доступность исходного кода
- 2) ориентация на использование оконного графического интерфейса
- 3) использование языка высокого уровня C
- 4) возможность достаточно легкого перехода на другие аппаратные платформы

**Вопрос №11** Типы операционных систем используемых наиболее часто в настоящее время

- 1) системы семейства Windows
- 2) системы семейства Unix/Linux
- 3) системы семейства MS DOS
- 4) системы семейства IBM OS 360/370

**Вопрос №12** Задачи необходимые решать при создании мультипрограммных ОС

- 1) защита кода и данных разных приложений, размещенных вместе в основной памяти
- 2) централизованное управление ресурсами со стороны ОС
- 3) переключение процессора с одного приложения на другое
- 4) необходимость размещения в основной памяти кода и данных сразу многих приложений

**Вопрос №13** Соотношение между используемыми на СЕРВЕРАХ операционных систем сложилось в настоящее время

- 1) примерно поровну используются системы семейств Windows и Unix/Linux
- 2) около 10 % — системы семейства Windows, около 90 % — системы семейства Unix/Linux
- 3) около 90 % — системы семейства Windows, около 10 % — системы семейства Unix/Linux
- 4) около 30 % — системы семейства Windows, около 30 % — системы семейства Unix/Linux, около 40 % — другие системы

**Вопрос №14** Утверждения относительно понятия «Ядро операционной системы» являющиеся правильными

- 1) ядро реализует наиболее важные функции ОС
- 2) подпрограммы ядра выполняются в привилегированном режиме работы процессора
- 3) ядро в сложных ОС может строиться по многоуровневому принципу

4) ядро всегда реализуется на аппаратном уровне

**Вопрос №15** Координаты текущего положения мыши при нажатии левой кнопки  
\_\_\_\_\_?

**Вопрос № 16** Состав программного уровня подсистемы ввода/вывода

- 1) драйверы
- 2) диспетчер ввода/вывода
- 3) системные вызовы
- 4) контроллеры

**Вопрос № 17** Понятие «порт ввода/вывода» \_\_\_\_\_?

**Вопрос №18** Правильные утверждения относительно понятия прерывания являются

- 1) прерывания — это механизм реагирования вычислительной системы на происходящие в ней события
- 2) прерывания используются для синхронизации работы основных устройств вычислительной системы
- 3) прерывания возникают в непредсказуемые моменты времени
- 4) прерывания — это основной механизм планирования потоков.

## Комплект заданий для практических работ по дисциплине «Введение в операционную систему UNIX»

### Практическая работа №1

Тема: Основы ОС Unix.

**Цель занятия:** знакомство с командной строкой UNIX, командами по работе с файловой системой и командами обработки текстовых файлов.

#### Средства обучения

- 1) Персональный компьютер
- 2) Операционная система Unix

#### Порядок выполнения работы:

- ознакомьтесь с краткими теоретическими сведениями
- изучите функциональность и предназначение «рабочего стола» ОС
- ознакомьтесь и апробируйте приложения
- смените системное время, обои «рабочего стола»
- попрактикуйтесь с предназначением «горячих клавиш»
- научитесь запускать командную строку и выполнять действия описанные ниже (в кратких теоретических сведениях).

#### Основные понятия:

- сеанс работы
- виртуальные консоли
- оболочка
- рабочая среда
- удаленная консоль

#### Используемые команды:

alias - вывод списка или создание нового алиаса для команды  
date - вывод или установка системной даты и времени  
df - вывод информации об использовании дискового пространства  
env - вывод информации о среде, запуск приложений с измененными переменными среды  
exit - завершение сеанса работы  
hostname - вывод имени машины  
ma - вывод справки по команде  
uname - вывод системной информации  
whereis - вывод информации о расположении файла  
which - вывод полного пути до файла вызываемого командой

UNIX — многозадачная многопользовательская операционная система. Система может работать в режиме с графической оболочкой или без нее. В рамках курса изучения ОС UNIX все работы будут проводиться в системе, работающей в многопользовательском режиме с поддержкой сети без графической оболочки.

Системная консоль — это монитор и клавиатура, связанные непосредственно с системой. Для входа в систему под несколькими учетными записями, используя один монитор и одну клавиатуру, система обеспечивает доступ к виртуальным консолям, которые позволяют войти в систему под несколькими именами в одно время. Сеанс работы начинается со входа в систему, для чего пользователь должен ввести свое имя и пароль. Для завершения сеанса работы с системой вы можете воспользоваться командой exit или сочетанием клавиш CTRL-D.

Оболочка (shell, командный интерпретатор) — это программа, которая воспринимает введенные пользователем данные и транслирует это в системные команды. Оболочка запускается сразу после входа пользователя в систему. Используя язык обращения к

оболочке можно создавать скрипты.

Рабочая среда — это множество переменных, к которым имеют доступ все выполняемые команды.

Файл `/etc/profile` содержит переменные среды на уровне всей системы. Файл `.bash_profile` содержит переменные среды пользователя. При входе в систему первыми скриптами, выполняемыми оболочкой, являются эти скрипты.

### **Задание 1**

1. Войдите в систему, используя имя пользователя и пароль, выданные вам преподавателем.
2. Определите имя машины. Найдите две разные команды, которыми можно это сделать.
3. Определите архитектуру процессора машины, используемой вами.
4. Выведите на экран время и дату в формате «31/12/2099 23:59».
5. Выведите на экран переменные среды. Объясните, что означают переменные SHELL, USER, PATH, HOME, LANG, MAIL, PS1.
6. Измените переменную PS1, что изменилось?
7. Запустите приложение `bash`, изменив переменную домашнего каталога пользователя. Проверьте, что изменения вступили в силу. Как это можно сделать?
8. Определите местоположение в системе файла вызываемого командой `bash`, используя команду `which`.

### **Пользователи и группы**

Основные понятия:

- учетная запись пользователя
- домашний каталог
- идентификатор пользователя

**Используемые команды:**

`id` - вывод информации о текущем пользователе

`groups` - просмотр списка групп, участником которых является текущий пользователь

`finger` - поиск информации о пользователях

`chsh` - изменение стандартной оболочки пользователя

`chfn` - изменение информации о пользователе, выводимой командой

`finger last` - вывод списка пользователей, заходивших в систему в последнее время

`mail` - отправление и чтение почтовых сообщений

`mesg` - включение/выключение отправки прямых сообщений на терминал пользователя

`passwd` - изменение пароля пользователя

`who` - вывод списка пользователей, находящихся в системе в текущий момент

`whoami` - имя текущего пользователя

`write` - отправка прямого сообщения на терминал пользователя

Для идентификации пользователей в системе используются учетные записи. Вводя имя пользователя и пароль при входе в систему, вы представляетесь системе, позволяя ей открыть вашу домашнюю папку, выдать вам права на нужные каталоги и подключить другие специфичные для вас атрибуты. Каждому пользователю соответствует набор сведений:

- системное имя пользователя — имя, которое вы используете при входе в систему;
- идентификатор пользователя (UID) — уникальный номер пользователя в системе;
- идентификатор группы (GID) — номер основной группы, к которой относится пользователь;
- комментарий (как правило, полное имя — имя и фамилия или псевдоним пользователя, по которому другие пользователи могут определить, кому принадлежит учетная запись);

- домашний каталог — личный каталог пользователя, на доступ к которому пользователь имеет полные права;
- начальная оболочка — оболочка, запускаемая системой при входе пользователя в систему.

Пользователи системы состоят в группах, используемых для организации доступа нескольких пользователей к ресурсам. Каждый пользователь состоит минимум в одной группе, название которой, как правило, совпадает с именем пользователя, и которая создается вместе с учетной записью пользователя.

Пользователь может самостоятельно изменить свой пароль для входа в систему, используя команду `passwd`. Изменить начальную оболочку можно, используя команду `chsh`.

С помощью команды `who` можно получить информацию о пользователях, которые вошли в систему. На экране появится примерно такая информация:

```
[elvis@station elvis]$ who
elvis    tty2          May  5 15:07
root     tty1          May  3 07:50
blondie  :0            May  5 08:48
blondie  pts/0        May  5 09:03 (:0.0)
```

В первом столбце отображается список пользователей, которые вошли в систему, в последних столбцах — дата и время входа. Данные второго столбца показывают, откуда пользователь вошел в систему. Первые два пользователя (`elvis` и `root`) использовали для входа `tty1` и `tty2`, что означает первую и вторую виртуальную консоль соответственно. `:0` для пользователя `blondie` означает вход в с использованием графического интерфейса, а `pts/0` относится к первому из терминалов, открытых в графической среде.

### Задание 2

1. Определите имя вашего пользователя.
2. Определите имена пользователей, работающих с системой в данный момент. Найдите себя в этом списке.
3. Определите имена трех последних пользователей кроме вас, заходивших в систему.
4. Определите, в каких группах состоит ваш пользователь.
5. Откройте файл `/etc/passwd`, используя команду `more /etc/passwd`. Найдите запись с данными вашего пользователя. Проанализируйте все поля записи и опишите их назначение. Используйте справку по файлу (`man 5 passwd`) для ознакомления со значением всех полей.
6. Выведите на экран информацию о пользователе, используя команду `finger`. Попробуйте изменить полное имя пользователя и добавить свой номер телефона при помощи команды `chfn`.
7. Договоритесь с соседним пользователем об организации обмена прямыми сообщениями. Обменяйтесь с ним сообщениями в режиме прямого диалога.
8. Исследуйте возможности блокирования и разблокирования средств приема сообщений.
9. По договоренности с коллегами обменяйтесь несколькими почтовыми сообщениями. Проанализируйте возможности обработки поступивших почтовых сообщений.
10. Проанализируйте с использованием команды `history` содержание лабораторной работы, продумайте ответы на нижеприведенные контрольные вопросы и сдайте выполненную работу преподавателю. После получения зачета по работе – уничтожьте все созданные файлы и корректно выйдите из системы.

### Контрольные вопросы:

1. Опишите процесс загрузки операционной системы.
2. Что такое уровни инициализации, и зачем они нужны?
3. В чем заключается процедура авторизации пользователя? Цель авторизации? Какие действия совершает система после того, как пользователь ввел пароль, и до того, как пользователь получает возможность передавать системе какие-то команды?

4. Опишите две реальные ситуации, когда вам может понадобиться получить дату и время в консоли UNIX.
5. Что такое среда пользователя? Опишите, как добавить новую переменную в среду, как изменить значение существующей переменной для одного пользователя и всех пользователей в системе.
6. Опишите одну реальную ситуацию, когда вам может понадобиться изменить переменную среды при запуске приложения.
7. Для чего используются группы пользователей?
8. Зачем нужны идентификаторы пользователей?
9. Объясните содержание и назначение каждого поля регистрационной записи.
10. В чем отличие в диалоге прямыми сообщениями и почтовыми?

#### **Содержание отчета:**

Отчет должен содержать:

Наименование темы

Цель работы

Содержание работы

Ответы на контрольные вопросы

## **Практическая работа №2** **Тема: Командная строка в Unix.**

### **Цель работы**

Освоение основ командной строки Unix/Linux, включая навигацию, управление файлами, просмотр информации о системе и запуск команд через командную оболочку (shell, например, Bash).

### **Средства обучения**

- 1) Персональный компьютер
- 2) Операционная система Unix

### **Порядок выполнения работы:**

- ознакомьтесь с краткими теоретическими сведениями и необходимыми материалами по данной теме
- создайте рабочий каталог с вашими инициалами
- в данном каталоге необходимо создать текстовый файл, и открыть для редактирования
- текст в файле может содержать произвольную информацию
- создайте ещё один каталог с произвольным названием, в который скопируете и далее переименуйте вышеописанный текстовый файл
- удалите свою рабочую папку
- поэкспериментируйте с командами **pwd**, **whoami**, **who**, **uptime**, **free**.

### **Краткие теоретические сведения:**

**whoami** Ввести имя под которым я зарегистрирован.

**date** Вывести или изменить дату и время. Например, чтобы установить дату и время равную 2000-12-31 23:57, следует выполнить команду:

```
date 123123572000
```

**time** Получить информацию о времени, нужного для выполнения процесса + еще кое-какую информацию. Не путайте эту команду с date. Например: Я могу определить как много времени требуется для вывода списка файлов в директории, набрав последовательность:

```
time ls
```

**at** Запуск команды в определенное время  
**who** Определить кто из пользователей работает на машине.  
**uptime** Количество времени прошедшего с последней перезагрузки.  
**ps a** Список текущих процессов.  
**pidof** Вывод номера процесса  
**top** Интерактивный список текущих процессов отсортированных по использованию сри.  
**uname** Вывести системную информацию.  
**free** Вывести информацию по памяти.  
**df -h** (=место на диске) Вывести информацию о свободном и используемом месте на дисках (в читабельном виде).  
**du / -bh | more** (=кто сколько занял) Вывод детальной информации о размере файлов по директориям начиная с корневой (в читабельном виде).  
**cat /proc/cpuinfo** Системная информация о процессоре. Заметьте, что файла в /proc директории - не настоящие файлы. Они используются для получения информации, известной системе.  
**cat /proc/interrupts** Используемые прерывания.  
**cat /proc/version** Версия ядра Linux и другая информация  
**cat /proc/filesystems** Вывести используемые в данный момент типы файловых систем.  
**lsmod** (как root) Вывести информацию о загруженных в данный момент модулях ядра.  
**set|more** Вывести текущие значения переменных окружения.  
**chmod** режим файл  
**kill.killall.xkill** Завершение работы процесса  
**nice** Запуск команд с пониженным приоритетом  
**echo \$PATH** Вывести значение переменной окружения "PATH" Эта команда может использоваться для вывода значений других переменных окружения. Воспользуйтесь командой set, для получения полного списка.

Также дополнительные команды, изучите в материалах, предложенных для ознакомления.

#### **Контрольные вопросы:**

2. Дайте характеристику командам: **who, mkdir, cat [имя\_файла] | more, nano**.
4. Как получить справку о необходимой команде?
5. С какой целью используется команда **cal, head**?

#### **Содержание отчета:**

Отчет должен содержать:

- Наименование темы
- Цель работы
- Содержание работы
- Ответы на контрольные вопросы

### **Практическая работа №3**

**Тема: Интерпретатор команд**

Изучение командного интерпретатора **bash**

**Цель работы:** ознакомление с базовыми правилами создания и применения командных скриптов в **Linux**.

#### **Краткий обзор:**

Командный интерпретатор **bash** это обычная прикладная программа, выполняющая

команды пользователя, задаваемые либо в интерактивном (диалоговом) режиме, либо в виде командных файлов, называемых также скриптами.

**Bash** можно рассматривать как своеобразный язык программирования, позволяющий создавать новые программы с достаточно сложными функциями, используя в качестве основных операций вызовы других, более простых программ.

Базовой конструкцией языка **bash** является простая команда. Она состоит из имени команды и, если это необходимо, параметров, разделенных пробелами. Имя команды – это обычно имя исполняемого файла (либо двоичной программы, либо **bash**-скрипта).

### Перенаправление ввода/вывода

Большинство команд UNIX выводят результат своей работы в текстовом виде на устройство стандартного вывода. По умолчанию это означает, что результаты выводятся на монитор пользователя. Однако стандартный вывод легко может быть перенаправлен в файл или на устройство. Для этого в команде используются символы '>' и '>>'.

Многие команды используют также стандартный ввод (хэндл 0), который по умолчанию означает данные, вводимые с клавиатуры терминала. Признаком конца ввода служит комбинация Ctrl+D. Стандартный ввод также может быть перенаправлен для чтения данных из файла или с устройства (с помощью символа '<'), или даже непосредственно из текста команды.

Для выполнения команды shell запускает отдельный процесс. Результатом выполнения команды является код завершения процесса, который может затем быть проанализирован. Нулевое значение кода обычно означает нормальное завершение, значение, большее нуля – ошибку.

### Создание сценариев на языке оболочки **bash**

Сложная обработка текстов требует выполнения довольно длинных последовательностей команд. Поскольку ввод всех команд такой цепочки может занять достаточно много времени и чреват ошибками, в Unix существует понятие скрипта (сценария). Скрипт представляет собой любую последовательность команд, сохраненную в текстовом файле. Скрипт создается и редактируется средствами любого текстового редактора (команды `ee,ed,red,vi,ex`), также скрипт может создаваться программой или другим скриптом.

### Использование переменных

Как при интерактивной работе, так и при выполнении скриптов могут определяться и использоваться переменные, имеющие строковые значения. Имя переменной может быть последовательностью букв, цифр и знака подчеркивания, начинающейся с буквы или знака подчеркивания. Для присваивания значений переменным может использоваться оператор присваивания "=". При необходимости присвоить **bash**-переменной значение, содержащее пробелы и другие специальные знаки, оно заключается в кавычки. Например:

```
var_1=13 # "13" - это не число, а строка из двух цифр.
```

```
var_2="ОС UNIX" # здесь двойные кавычки необходимы, так как в строке есть пробел.
```

Обратите внимание, что переменная и ее значение должны быть записаны без пробелов относительно символа "=".

Для получения значения переменной перед ее именем записывается символ '\$'.

```
echo $var_1 # вывод значения переменной $var_1
```

Также значение переменной можно прочитать с клавиатуры:

```
read var_1
```

### Структурные операторы **bash**

Shell, как и любой язык программирования, содержит набор операторов управления порядком выполнения команд, таких как `if`, `case`, `while`, `until`, `for`, `break` и некоторые другие. Логические выражения, используемые в операторах управления, строятся на основе кодов завершения команд, при этом специальная команда `test` позволяет проверить разнообразные

условия, такие, как существование и тип указанного файла, равенство или неравенство строковых и числовых выражений и т.п.

### 1. Команда TEST

Команда test проверяет выполнение некоторого условия. С использованием этой встроенной команды формируются операторы выбора и цикла языка shell. Существует два возможных формата команды test:

**test условие      или      [ условие ]**

Если вы будете использовать второй вариант **bash** будет распознавать команду test по открывающей скобке "[", поэтому между скобками и содержащимся в них условием обязательно должны быть пробелы. Пробелы должны быть и между значениями и символом сравнения или операции. Не забывайте о противоположном требовании для операции присваивания значений переменным.

Имеется три типа проверок:

- оценка числовых значений;
- оценка типа файла;
- оценка строк.

Для чисел синтаксис следующий:

N op M, где N, M –числовые переменные, операция op принимает следующие значения:

-eq      равно  
-ne      не равно  
-le      меньше или равно  
-ge      больше или равно  
-lt      меньше  
-gt      больше

#### Синтаксис для файла:

op filename, где операция op может принимать следующие значения:

-s файл существует и не пуст  
-f обычный файл  
-d файл является директорией  
-w имеется право на запись  
-r имеется право на чтение

Синтаксис для строк:

S op R, где операция op может принимать следующие значения:

= эквивалентность

!= неэквивалентность

op R, где операция op может принимать следующие значения:

-z строка пустая

-n строка непустая

### 2. Условный оператор IF

Синтаксис:

```
if <условие>
then <список команд>
[elif <условие> then <список команд>]
[else <альтернативный список команд >]
fi
elif [ $z = 815 ]
```

Пример:

```
echo -n "Введите код города "
read z
if [ $z = 423 ]
then echo "Владивосток"
elif [ $z = 495 ]
then echo "Москва"
```

```

then echo "Мурманск"
else echo "Неверный код города"
fi

```

Несколько проверок могут быть объединены логическими операциями –a (and) и -o (or).

### 3. Оператор выбора CASE

Синтаксис:

```

case <выражение> in
значение 1) < список команд> ;;
значение 2) < список команд> ;;
значение 3) и т.д.
esac

```

Пример:

```

echo -n "Введите код города "
read z
case $z in
423) echo Владивосток ;;
495) echo Москва ;;
815) echo Мурманск ;;
*) echo "Неизвестный город" ;;

```

esac

В конце каждой строки с альтернативным значением обязательно ставится ";;", так как в каждом случае может быть выполнено несколько команд. Если эти команды будут записаны в одну строку, то символ ";" будет использоваться как разделитель команд.

Обычно последняя строка выбора имеет шаблон \*. Эта строка выбирается, если не произошло совпадение значения переменной (здесь \$z) ни с одним из ранее записанных шаблонов, ограниченных скобкой ")".

### 4. Оператор цикла FOR

В цикле при каждой итерации переменной будет присвоен очередной элемент списка, над которым будут выполнены указанные команды.

Синтаксис:

```

for <переменная>[in <список значений>]
do <список команд>
done

```

Пример:

```

for n in 1 2 3
do echo $n
done

```

### 5. Оператор цикла WHILE с истинным условием

Выполнение команд повторяется до тех пор, пока условие выполняется, т.е. код завершения последней команды в цикле равен "0", или цикл не будет прерван изнутри специальными командами ("break", "continue" или "exit").

Синтаксис:

Пример:

```

p=1
while <условие>
do <команда>
done

```

```

i=1
a=5
b=2
while test $i -le $b
do
p=`expr $p "*" $a `
i=`expr $i + 1 `
done

```

### 6. Оператор цикла UNTIL с ложным условием

Этот оператор инвертирует условие повторения по сравнению с оператором while, т.е. команды будут выполняться до тех пор, пока условие не выполнится.

Синтаксис:

```

until <условие>
do <команды>
done

```

Пример:

```

until date | grep 12:00:
do
sleep 30

```

done

В данном примере каждые 30 секунд выполняется командная строка условия. Команда "date" выдает текущую дату и время. Команда "grep" получает эту информацию через конвейер и пытается совместить заданный шаблон "12:00:" с временем, выдаваемым командой "date". При несовпадении "grep" выдает код возврата "1", что соответствует значению "ложь", и цикл "выполняет ожидание" в течение 30 секунд, после чего повторяется выполнение условия. Когда условие станет истинным, работа цикла закончится.

### **Ход работы:**

В ходе выполнения практической работы необходимо создать командный файл, решающий задачу в соответствии с полученным заданием (см. Варианты заданий), а затем выполнить следующие шаги:

1. Сохранить последовательность команд в файле, проверить ее выполнение с помощью команды sh.
2. Придать файлу-скрипту право на выполнение, проверить выполнение скрипта запуском из разных каталогов.
3. После того, как будут выполнены проверки 1, 2, создать в своем домашнем каталоге подкаталог /bin и перенести в него файл-скрипт. Включить созданный подкаталог в переменную PATH и проверить запуск скрипта из нескольких разных каталогов без указания пути.

### **Варианты заданий:**

1. Создать скрипт, осуществляющий вывод на экран меню, состоящего из следующих пунктов:

1. текущий пользователь,
2. смена пароля пользователя,
3. запущенные процессы,
4. выход.

Пользователь выбирает один из пунктов и вводит соответствующий номер. После выполнения выбранной команды (см. Приложение), необходимо снова вывести меню пользователю. Процесс повторять до тех пор, пока не будет выбран пункт меню «выход» (команда exit).

2. Создать скрипт, который запрашивает у пользователя пароль. Если пароль введен верно, постранично отображается в длинном формате с указанием скрытых файлов содержимое каталога /etc. Если неверно, запрос на ввод пароля повторяется.

3. Создать скрипт, который собирает все текстовые файлы текущего каталога в один архивный файл my.tar (команда TAR), после паузы просматривает содержимое файла my.tar, затем командой GZIP сжимает архивный файл my.tar.

4. Создать скрипт, осуществляющий переход в родительский каталог, формирующий файл с листингом каталога и возвращающийся в исходный каталог.

5. Создать скрипт, формирующий текстовый файл со списком файлов в домашнем каталоге. Затем вывести этот список на экран в алфавитном порядке.

6. Создать скрипт для определения является ли введенный пользователем год високосным или нет. (Необходимо для организации вычислений использовать команду expr, см. приложение)

7. Создать скрипт, преобразующий десятичное число в двоичное. (Необходимо для организации вычислений использовать команду expr, см. приложение)

### **Контрольные вопросы**

1. Какое назначение имеют **bash** - файлы?
2. Как создать **bash** - файл и сделать его выполняемым?
3. Какие типы переменных используются в **bash** - файлах?
4. Назовите операторы создания циклов.
5. В каком случае перед именем переменной необходимо указывать знак \$?

**Отчет о работе должен содержать:**

1. Цель работы
2. Задание
3. Ход работы
4. Листинг командного скрипта
5. Выводы по работе
6. Ответы на контрольные вопросы

## Практическая работа №5

### Тема: Прикладные программы в ОС Unix.

Установка и удаление программ. Менеджеры пакетов RPM и Yum. Установка пакетов и основы использования.

**Цель:** научиться работать с программами установки ПО в семействе операционных систем семейства UNIX при помощи RPM-пакетов.

Теоретическое обоснование

Различия между системой управления пакетами и инсталлятором:

<i>Package Management System</i>	<i>Installer</i>
Обычно часть операционной системы	Каждый продукт поставляется вместе со своим инсталлятором
Использует одну базу данных для установки	Производит свою установку
Может проверять и управлять всеми пакетами в системе	Работает только со связанным продуктом
Один производитель	Много производителей
Один формат пакета	Много форматов инсталлятора

#### Методические указания

Для выполнения заданий вам потребуется вспомнить навыки работы в unix системах, а также доступ в интернет и (по возможности) работоспособный компьютер с Linux, на котором можно будет выполнять все встречающиеся команды.

#### Задания

Изучите опции и назначение команды `rpm`

Составьте таблицу команд *работы с пакетами rpm*

#### Основные команды работы с RPM

Распространенные программы установки семейства UNIX

Большинство дистрибутивов операционных систем на базе GNU, Linux и BSD имеет встроенные системы управления пакетами, с помощью которых можно устанавливать как необходимые компоненты операционной системы, так и стороннее программное обеспечение, зачастую, даже если оно использует собственный установщик, которым это не предусмотрено.

Наиболее известные системы управления пакетами: RPM, dpkg, Pacman, Portage, Entropy, PiSi.

Существует две основных формы распространения ПО для LINUX: в исходных текстах и в виде исполняемых модулей. И в том и в другом случае пакет ПО может поставляться либо в виде tar-gz архива, либо в виде rpm-пакета.

Проще всего установить ПО, представленное в виде rpm-пакета, содержащего исполняемые файлы, этот способ и рассмотрим первым.

**rpm**-пакеты — это специальным образом подготовленные архивы, предназначенные для обработки программой rpm. В Интернет rpm-пакеты можно найти на различных серверах.

Основные команды работы с RPM			
Команда	Описание команды	Дополнительные опции	Когда используется опция
	Установить RPM-пакет с проверкой зависимостей		Если нужно установить более старый пакет, т.е. нужно сделать даунгрейд,
	Установить RPM-пакет без проверки зависимостей		Если система пытается ставить пакет, но дает сообщения, что файлы из пакета конфликтуют с файлами из другого пакета, нужно добавить опцию
	Проверить, какие файлы будут установлены из rpm-архива. Выполнять команду надо в каталоге, где лежит rpm-файл. Будут показаны только имена файлов, без путей		
	Узнать информацию о пакете в rpm-архиве (Название, описание, версия и т.д.) Выполнять команду надо в каталоге, где лежит rpm-файл		
	Узнать, какие файлы находятся в пакете		
	Узнать, какому пакету принадлежит файл:		
	Выдать список всех установленных пакетов		
	Найти пакет, установленный в системе, по части имени		
	Узнать, установлен или нет пакет, расположенный в репозитории		
	Удаление пакета: Чтобы удалить пакет, замените имя_пакета действительным именем первоначального пакета.		

### Контрольные вопросы

1. Что такое инсталляция программного обеспечения?
2. Какие варианты установки программного обеспечения вы знаете?
3. Расскажите о программе RPM

### Форма отчёта

Конспект. Таблица. Ответы на вопросы.

## **5. Методические материалы, определяющие процедуры оценивания компетенции**

### **5.1 Критерии оценивания качества ответа на контрольные вопросы**

Оценка **«отлично»** выставляется за глубокое знание предусмотренного программой материала, за умение четко, лаконично и логически последовательно отвечать на поставленные вопросы.

Оценка **«хорошо»** – за твердое знание основного (программного) материала, за грамотные, без существенных неточностей ответы на поставленные вопросы.

Оценка **«удовлетворительно»** – за общее знание только основного материала, за ответы, содержащие неточности или слабо аргументированные, с нарушением последовательности изложения материала.

Оценка **«неудовлетворительно»** – за незнание значительной части программного материала, за существенные ошибки в ответах на вопросы, за неумение ориентироваться в материале, за незнание основных понятий дисциплины.

### **5.2 Критерии оценивания качества выполнения практических работ**

Оценка **«зачтено»** выставляется обучающемуся, если практическая работа выполнена правильно и обучающийся ответил на все вопросы, поставленные преподавателем на защите.

Оценка **«не зачтено»** выставляется обучающемуся, если практическая работа выполнена неправильно или обучающийся не проявил глубоких теоретических знаний при защите работы

### **5.3 Критерии оценивания коллоквиума**

Оценка **«отлично»** выставляется за глубокое знание предусмотренного программой материала, за умение четко, лаконично и логически последовательно отвечать на поставленные вопросы.

Оценка **«хорошо»** – за твердое знание основного (программного) материала, за грамотные, без существенных неточностей ответы на поставленные вопросы.

Оценка **«удовлетворительно»** – за общее знание только основного материала, за ответы, содержащие неточности или слабо аргументированные, с нарушением последовательности изложения материала.

Оценка **«неудовлетворительно»** – за незнание значительной части программного материала, за существенные ошибки в ответах на вопросы, за неумение ориентироваться в материале, за незнание основных понятий дисциплины.

### **5.4 Критерии оценивания тестирования**

При тестировании все верные ответы берутся за 100%. 90%-100% отлично 75%-90% хорошо, 60%-75% удовлетворительно, менее 60% неудовлетворительно