

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
Обнинский институт атомной энергетики –
филиал федерального государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский ядерный университет «МИФИ»
(ИАТЭ НИЯУ МИФИ)

Утверждено на заседании
УМС ИАТЭ НИЯУ МИФИ
Протокол №2-8/2024 От 30.08.2024

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Работа с облачными серверами с ОС Linux

Шифр, название дисциплины

01.04.02 «Прикладная математика и информатика»

Шифр, название специальности/направления подготовки

Математическое моделирование и прикладной анализ данных

Название программы магистратуры

магистр

(Квалификация (степень) выпускника)

Форма обучения: очная

г. Обнинск 2024 г.

Программа составлена в соответствии с требованиями образовательного стандарта высшего образования национального исследовательского ядерного университета «МИФИ» по направлению подготовки 01.04.02 – Прикладная математика и информатика. (квалификация (степень) магистр).

Программу составил:

_____ С.В. Ермаков, доцент, к.ф.-м.н, доцент

Рецензент:

_____ Г.Е. Деев, доцент, к.ф.-м.н, доцент

Программа рассмотрена на заседании ОИКС

(протокол № 5/7 от «30» июля от 2024 г.)

Руководитель направления подготовки 01.04.02
«Прикладная математика и информатика»

_____ Ермаков С.В.

« ____ » _____ 2024 г.

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения ООП магистратуры обучающийся должен овладеть следующими результатами обучения по дисциплине:

Коды компетенций	Результаты освоения ООП <i>Содержание компетенций*</i>	Перечень планируемых результатов обучения по дисциплине**
ОПК-2	Способен совершенствовать и реализовывать новые математические методы решения прикладных задач	З-ОПК-2 Знать основные понятия, математические методы решения прикладных задач, принципы математического моделирования и методы верификации. У-ОПК-2 Уметь применять полученную теоретическую базу для решения практических задач В-ОПК-2 Владеть основными математическими методами решения прикладных задач
ОПК-3	Способен разрабатывать математические модели и проводить их анализ при решении задач в области профессиональной деятельности	З-ОПК-3 Знать основные методы и принципы математического моделирования, методы построения математических моделей типовых профессиональных задач, способы нахождения решений математических моделей и содержательной интерпретации полученных результатов. У-ОПК-3 Уметь составлять математические модели типовых профессиональных задач, находить способы их решения и профессионально интерпретировать смысл полученного результата. В-ОПК-3 Владеть методами построения математических моделей типовых профессиональных задач, способами нахождения решений математических моделей и содержательной интерпретации полученных результатов

2. Место дисциплины в структуре ООП магистратуры

Дисциплина реализуется в рамках общенаучного модуля.

Для освоения дисциплины необходимы компетенции, сформированные в рамках изучения следующих дисциплин: Github, SQL для анализа данных

Дисциплина изучается на 1 курсе в 2 семестре.

3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся

Общая трудоемкость (объем) дисциплины составляет 2 зачетных единиц (з.е.), 72 академических часа.

3.1. Объем дисциплины по видам учебных занятий (в часах)

	Семестр		
	№ 1	№ 2	Всего
	Количество часов на вид работы:		
Контактная работа обучающихся с преподавателем			
Аудиторные занятия (всего)		64	64
В том числе:			
<i>лекции</i>		32	32
<i>практические занятия</i>		32	32
<i>лабораторные занятия</i>			
Промежуточная аттестация			
В том числе:			
<i>зачет</i>		Зачет	Зачет
<i>экзамен</i>		-	-
Самостоятельная работа обучающихся (всего)		8	8
В том числе:			
<i>проработка учебного (теоретического) материала</i>			
<i>выполнение индивидуальных заданий</i>			
<i>подготовка ко всем видам контрольных испытаний текущего контроля успеваемости (в течение семестра)</i>			
<i>подготовка ко всем видам контрольных испытаний промежуточной аттестации (по окончании семестра)</i>		8	8
<i>Всего (часы):</i>		72	72
<i>Всего (зачетные единицы):</i>		2	2

4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах)

№ п/п	Наименование раздела /темы дисциплины	Общая трудоёмкость всего (в часах)	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость (в часах)				Формы текущего контроля успеваемости
			Аудиторные учебные занятия			СРО	
			Лек	Сем/Пр	Лаб		
1.			32	32	-	8	

1.1.	Командная строка, shell, bash, окружение рабочего стола	3	1	1			
1.2.	sudo apt install, обновление, репозитории, дистрибутив	3	1	1	-		
1.3.	Создание базовых скриптов, alias, аргументы команд, получение помощи по командам	4	1	1	-		
1.4.	Навигация по файловой системе, ее изучение	4	1	1	-		
1.5.	Создание файлов и ссылок	4	1	1			
1.6.	Просмотр и редактирование файлов	4	1	1	-		
1.7.	Ввод/вывод/конвейеры	4	1	1	-		
1.8.	Переменные, env, подстановки, экранирование	4	1	1			
1.9	Регулярные выражения, grep, sed, find	4	1	1			
1.10	Введение в bash: Циклы и ветвления	4	1	1			
1.11	Введение в bash: работа с аргументами командной строки	6	2	2			
1.12	Привилегии, для файлов	6	2	2			
1.13	Процессы, диспетчер задач	6	2	2			
1.14	Работа с сетью, ping, ssh, curl	6	2	2		2	
1.15	Создание пользователя группы	6	2	2			
1.16	Сборка из исходных файлов	6	2	2			
1.17	Своп, df, du	6	2	2			
1.18	Nginx	6	2	2		2	
1.19	Управление базой данных через консоль	6	2	2		2	
1.20	Cron	6	2	2			
1.21	Systemd	6	2	2		2	

4.2. Содержание дисциплины, структурированное по разделам (темам)

Лекционный курс

№	Наименование раздела /темы дисциплины	Содержание
1.1.	Командная строка, shell, bash, окружение рабочего стола	интерфейс для взаимодействия с операционной системой, ввод команд для выполнения. Shell — интерпретатор команд, например, bash, zsh, fish; как запускать и настроить shell. Bash — популярный shell, его особенности, синтаксис, переменные, скрипты. Окружение рабочего стола — настройки GUI (например, GNOME, KDE), работа с окнами, панели задач, настройка ярлыков и прочее.
1.2.	sudo apt install, обновление, репозитории, дистрибутив	sudo apt install — установка пакетов с помощью apt (Advanced Package Tool) в дистрибутивах на базе

		<p>Debian/Ubuntu.</p> <p>Обновление — использование команд <code>apt update</code>, <code>apt upgrade</code>, <code>apt full-upgrade</code>, а также управление версиями пакетов.</p> <p>Репозитории — работа с репозиториями (сохранение, добавление, удаление), настройка источников пакетов (<code>/etc/apt/sources.list</code>).</p> <p>Дистрибутивы — различие между дистрибутивами Linux (Debian, Ubuntu, Fedora, Arch) и их особенности.</p>
1.3.	Создание базовых скриптов, <code>alias</code> , аргументы команд, получение помощи по командам	<p>Создание базовых скриптов — создание и запуск <code>bash</code>-скриптов, использование <code>Shebang</code> (<code>#!/bin/bash</code>).</p> <p><code>Alias</code> — создание псевдонимов для команд с помощью <code>alias</code>, их использование для сокращения команд.</p> <p>Аргументы команд — работа с аргументами команд в <code>bash</code>-скриптах с использованием <code>\$1</code>, <code>\$2</code> и т.д.</p> <p>Получение помощи — использование команд <code>man</code>, <code>info</code>, <code>--help</code> для получения справки по командам.</p>
1.4.	Навигация по файловой системе, ее изучение	<p>Навигация — использование команд <code>cd</code>, <code>ls</code>, <code>pwd</code> для перехода по каталогам.</p> <p>Просмотр структуры файлов — команды <code>tree</code>, <code>ls -R</code> для отображения структуры каталогов.</p> <p>Символические и жесткие ссылки — понимание различий и применение команд <code>ln</code>, <code>ln -s</code>.</p>
1.5.	Создание файлов и ссылок	<p>Создание файлов — команды для создания файлов: <code>touch</code>, <code>echo</code>, редактирование с помощью <code>nano</code>, <code>vim</code>.</p> <p>Ссылки — создание символических и жестких ссылок с помощью <code>ln</code>, управление ссылками.</p>
1.6.	Просмотр и редактирование файлов	<p>Просмотр файлов — использование команд <code>cat</code>, <code>less</code>, <code>more</code>, <code>head</code>, <code>tail</code> для просмотра содержимого.</p> <p>Редактирование файлов — текстовые редакторы <code>nano</code>, <code>vim</code>, <code>emacs</code>, как работать с ними.</p>
1.7.	Ввод/вывод/конвейеры	<p>Ввод/вывод — использование стандартных потоков ввода (<code>stdin</code>), вывода (<code>stdout</code>) и ошибок (<code>stderr</code>).</p> <p>Конвейеры (<code>pipes</code>) — передача вывода одной команды на ввод другой с помощью <code> </code>.</p> <p>Перенаправление — перенаправление вывода с помощью <code>></code>, <code>>></code>, <code>2></code> и чтение с <code>&</code>.</p>
1.8.	Переменные, <code>env</code> , подстановки, экранирование	<p>Переменные — создание и использование переменных в <code>bash</code>, как задать переменные окружения.</p> <p><code>env</code> — управление переменными окружения, использование команды <code>env</code> для вывода всех переменных.</p> <p>Подстановки — командная подстановка с помощью обратных кавычек <code>`command`</code> или <code>\$(command)</code>.</p> <p>Экранирование — использование обратного слэша (<code>\</code>) для экранирования символов.</p>
1.9	Регулярные выражения, <code>grep</code> , <code>sed</code> , <code>find</code>	<p>Регулярные выражения — основы синтаксиса регулярных выражений для поиска и обработки текста.</p> <p><code>grep</code> — использование для поиска текста в файлах с поддержкой регулярных выражений.</p> <p><code>sed</code> — потоковый редактор для редактирования текста, замены и удаления.</p> <p><code>find</code> — поиск файлов по имени, типу, размеру и другим</p>

		критериям.
1.10	Введение в bash: Циклы и ветвления	Циклы — основы использования циклов for, while, until. Условия и ветвления — использование if, else, elif, case для принятия решений в скриптах.
1.11	Введение в bash: работа с аргументами командной строки	Аргументы командной строки — обработка аргументов скрипта через \$1, \$2, \$@, \$#. Пример обработки флагов и параметров
1.12	Привилегии, для файлов	Права доступа — чтение, запись, выполнение для владельца, группы и других (r, w, x). Команда chmod — изменение прав доступа, chown, chgrp — изменение владельца и группы. Системные права и sudo — настройка привилегий для команд.
1.13	Процессы, диспетчер задач	Процессы — управление процессами с помощью ps, top, htop, kill, pkill. Фоновые процессы — использование &, fg, bg, jobs для управления процессами. Таблицы процессов — изучение вывода команды ps, что такое PID, PPID.
1.14	Работа с сетью, ping, ssh, curl	ping — проверка доступности хоста в сети. ssh — удалённый доступ и управление серверами через SSH. curl — использование для получения данных из сети, работа с HTTP-запросами
1.15	Создание пользователя/группы	useradd, usermod, userdel — создание, изменение и удаление пользователей. groupadd, groupdel — создание и удаление групп. Управление правами пользователей — изменение паролей с помощью passwd, управление правами с помощью chmod, chown.
1.16	Сборка из исходных файлов	Компиляция программ — основы компиляции исходников с использованием make, gcc, cmake. Зависимости — как работать с зависимостями при сборке программ.
1.17	Своп, df, du	Своп (swap) — работа с виртуальной памятью, настройка своп-файлов и разделов. df — отображение информации о дисках и файловых системах. du — проверка использования дискового пространства на уровне каталогов.
1.18	Nginx	Установка и настройка — базовая настройка веб-сервера Nginx. Конфигурация — настройка виртуальных хостов, проксирование, балансировка нагрузки. Журналирование — работа с логами и мониторинг Nginx.
1.19	Управление базой данных через консоль	MySQL/PostgreSQL — основы работы с СУБД через консоль: создание, редактирование и удаление баз данных, таблиц, пользователей.

		Запросы — написание SQL-запросов для работы с данными.
1.20	Cron	Cron — настройка периодических задач с использованием cron, создание расписаний для выполнения команд. crontab — управление заданиями, синтаксис cron-выражений.
1.21	Systemd	Системные сервисы — создание, настройка и управление сервисами с помощью systemctl. Юнит-файлы — создание и настройка юнитов для служб. Системные журналы — работа с логами через journalctl.

Практические/семинарские занятия

№	Наименование раздела /темы дисциплины	Содержание
1.1.	Командная строка, shell, bash, окружение рабочего стола	интерфейс для взаимодействия с операционной системой, ввод команд для выполнения. Shell — интерпретатор команд, например, bash, zsh, fish; как запускать и настроить shell. Bash — популярный shell, его особенности, синтаксис, переменные, скрипты. Окружение рабочего стола — настройки GUI (например, GNOME, KDE), работа с окнами, панели задач, настройка ярлыков и прочее.
1.2.	sudo apt install, обновление, репозитории, дистрибутив	sudo apt install — установка пакетов с помощью apt (Advanced Package Tool) в дистрибутивах на базе Debian/Ubuntu. Обновление — использование команд apt update, apt upgrade, apt full-upgrade, а также управление версиями пакетов. Репозитории — работа с репозиториями (сохранение, добавление, удаление), настройка источников пакетов (/etc/apt/sources.list). Дистрибутивы — различие между дистрибутивами Linux (Debian, Ubuntu, Fedora, Arch) и их особенности.
1.3.	Создание базовых скриптов, alias, аргументы команд, получение помощи по командам	Создание базовых скриптов — создание и запуск bash-скриптов, использование Shebang (#!/bin/bash). Alias — создание псевдонимов для команд с помощью alias, их использование для сокращения команд. Аргументы команд — работа с аргументами команд в bash-скриптах с использованием \$1, \$2 и т.д. Получение помощи — использование команд man, info, --help для получения справки по командам.
1.4.	Навигация по файловой системе, ее изучение	Навигация — использование команд cd, ls, pwd для перехода по каталогам. Просмотр структуры файлов — команды tree, ls -R для отображения структуры каталогов. Символические и жесткие ссылки — понимание различий и применение команд ln, ln -s.

1.5.	Создание файлов и ссылок	Создание файлов — команды для создания файлов: touch, echo, редактирование с помощью nano, vim. Ссылки — создание символических и жестких ссылок с помощью ln, управление ссылками.
1.6.	Просмотр и редактирование файлов	Просмотр файлов — использование команд cat, less, more, head, tail для просмотра содержимого. Редактирование файлов — текстовые редакторы nano, vim, emacs, как работать с ними.
1.7.	Ввод/вывод/конвейеры	Ввод/вывод — использование стандартных потоков ввода (stdin), вывода (stdout) и ошибок (stderr). Конвейеры (pipes) — передача вывода одной команды на ввод другой с помощью . Перенаправление — перенаправление вывода с помощью >, >>, 2> и чтение с &.
1.8.	Переменные, env, подстановки, экранирование	Переменные — создание и использование переменных в bash, как задать переменные окружения. env — управление переменными окружения, использование команды env для вывода всех переменных. Подстановки — командная подстановка с помощью обратных кавычек `command` или \$(command). Экранирование — использование обратного слэша (\) для экранирования символов.
1.9	Регулярные выражения, grep, sed, find	Регулярные выражения — основы синтаксиса регулярных выражений для поиска и обработки текста. grep — использование для поиска текста в файлах с поддержкой регулярных выражений. sed — потоковый редактор для редактирования текста, замены и удаления. find — поиск файлов по имени, типу, размеру и другим критериям.
1.10	Введение в bash: Циклы и ветвления	Циклы — основы использования циклов for, while, until. Условия и ветвления — использование if, else, elif, case для принятия решений в скриптах.
1.11	Введение в bash: работа с аргументами командной строки	Аргументы командной строки — обработка аргументов скрипта через \$1, \$2, \$@, \$#. Пример обработки флагов и параметров
1.12	Привилегии, для файлов	Права доступа — чтение, запись, выполнение для владельца, группы и других (r, w, x). Команда chmod — изменение прав доступа, chown, chgrp — изменение владельца и группы. Системные права и sudo — настройка привилегий для команд.
1.13	Процессы, диспетчер задач	Процессы — управление процессами с помощью ps, top, htop, kill, pkill. Фоновые процессы — использование &, fg, bg, jobs для управления процессами. Таблицы процессов — изучение вывода команды ps, что такое PID, PPID.

1.14	Работа с сетью, ping, ssh, curl	ping — проверка доступности хоста в сети. ssh — удалённый доступ и управление серверами через SSH. curl — использование для получения данных из сети, работа с HTTP-запросами
1.15	Создание пользователя/группы	useradd, usermod, userdel — создание, изменение и удаление пользователей. groupadd, groupdel — создание и удаление групп. Управление правами пользователей — изменение паролей с помощью passwd, управление правами с помощью chmod, chown.
1.16	Сборка из исходных файлов	Компиляция программ — основы компиляции исходников с использованием make, gcc, cmake. Зависимости — как работать с зависимостями при сборке программ.
1.17	Своп, df, du	Своп (swap) — работа с виртуальной памятью, настройка своп-файлов и разделов. df — отображение информации о дисках и файловых системах. du — проверка использования дискового пространства на уровне каталогов.
1.18	Nginx	Установка и настройка — базовая настройка веб-сервера Nginx. Конфигурация — настройка виртуальных хостов, проксирование, балансировка нагрузки. Журналирование — работа с логами и мониторинг Nginx.
1.19	Управление базой данных через консоль	MySQL/PostgreSQL — основы работы с СУБД через консоль: создание, редактирование и удаление баз данных, таблиц, пользователей. Запросы — написание SQL-запросов для работы с данными.
1.20	Cron	Cron — настройка периодических задач с использованием cron, создание расписаний для выполнения команд. crontab — управление заданиями, синтаксис cron-выражений.
1.21	Systemd	Системные сервисы — создание, настройка и управление сервисами с помощью systemctl. Юнит-файлы — создание и настройка юнитов для служб. Системные журналы — работа с логами через journalctl.

Лабораторные занятия

Не предусмотрены.

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

В качестве учебно-методических материалов используется рекомендованная литература.

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

6.1. Паспорт фонда оценочных средств по дисциплине

№ п/п	Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции (или её части) / и ее формулировка	Наименование оценочного средства
1.3	Управление базой данных через консоль	ОПК-2:	Контрольная работа № 1
2.1.	Создание пользователя/группы	ОПК-3.	Контрольная работа № 1
2.3.	Свертка обобщенных функций	ОПК-2:	Контрольная работа № 2
2.4.	Ввод/вывод/конвейеры	ОПК-3.	Контрольная работа № 2
3.1.	Systemd	ОПК-2:	Контрольная работа № 2

6.2. Типовые контрольные задания или иные материалы

6.2.1. Зачет

В билете два теоретических вопроса и один практический

Теоретические вопросы билета:

1. Как с помощью команды `sudo apt install` установить пакет в Ubuntu?
2. Что такое репозиторий в Linux и как его добавить в систему?
3. Как создать базовый скрипт в bash и выполнить его?
4. Для чего используется команда `alias` и как создать собственный алиас?
5. Как получить справку по команде в командной строке?
6. Какие команды используются для навигации по файловой системе в Linux?
7. Чем отличается команда `ln` от `ln -s` при создании ссылок?
8. Как можно просматривать содержимое файла в Linux?
9. Как редактировать текстовый файл в командной строке?
10. Что такое конвейеры в Linux и как они используются для обработки данных?
11. Как задаются и используются переменные в bash? Приведите пример.
12. Чем отличается подстановка команд от подстановки переменных в bash?
13. Что такое регулярные выражения и как они используются в `grep`?
14. Как использовать команду `sed` для замены текста в файле?
15. Что такое циклы и ветвления в bash? Приведите пример использования.
16. Как передаются аргументы командной строки в bash-скрипт?
17. Как проверить и изменить привилегии для файла в Linux с помощью команд `chmod`, `chown`?
18. Как управлять процессами в Linux с помощью команд `ps`, `top`, `kill`?
19. Какие основные команды используются для работы с сетью в Linux (например, `ping`, `ssh`, `curl`)?

Критерий оценки – правильность и полнота ответа на вопросы. Оценка выставляется по шкале от 0 до 40 баллов: теоретические вопросы – 30 баллов, 10 баллов – дополнительные вопросы.

Зачет считается сданным при оценке не ниже 25 баллов.

6.2.2. Контрольная работа № 1

Задача: Администрирование Linux-системы

Создание и управление пользователями и группами:

Создайте нового пользователя с именем student и установите ему пароль.

Создайте группу с именем developers и добавьте в неё пользователя student.

Измените домашнюю директорию пользователя student на /home/student_data.

Управление файлами и правами доступа:

Создайте каталог /data/projects и файл в нем с именем project.txt.

Установите права на каталог так, чтобы только владелец мог читать и записывать файлы в нем, а остальные пользователи не имели доступа.

Создайте символическую ссылку на файл project.txt в домашней директории пользователя student.

Работа с процессами:

Найдите и выведите список всех работающих процессов, запущенных пользователем student.

Завершите процесс с именем example_process (если он существует) с помощью команды kill.

Убедитесь, что процесс завершен.

Установка и настройка программного обеспечения:

Установите пакет nginx с помощью apt (или другой команды, в зависимости от вашей системы).

Запустите службу nginx и убедитесь, что она работает, проверив статус с помощью systemctl.

Настройте nginx так, чтобы он обслуживал простой HTML-файл с приветствием в каталоге /var/www/html.

Автоматизация с помощью Cron:

Создайте задачу в cron, которая будет ежедневно в 5 утра создавать резервную копию каталога /data/projects в /data/backup.

Мониторинг системы:

Используя команду df, покажите информацию о дисковом пространстве на всех разделах.

Проверьте использование оперативной памяти с помощью команды free.

Требования:

В каждой задаче покажите команды, которые вы использовали, а также их вывод.

Объясните, какие действия вы выполняли, и зачем они были необходимы.

6.2.2. Контрольная работа № 2

Создать скрипт на bash, который будет создавать резервные копии каталога /home/user/Documents и сохранять их в каталог /home/user/Backups. Каждая копия должна быть уникальной и иметь имя, основанное на текущей дате и времени в формате backup_YYYY-MM-DD_HH-MM-SS.tar.gz.

Скрипт должен:

- Проверить, существует ли каталог /home/user/Backups. Если нет, создать его.

- Сжать содержимое каталога /home/user/Documents в архив.
- Сохранить архив в каталог /home/user/Backups.
- Принудительно удалить старые архивы, которые старше 7 дней.

Настроить cron-задачу, которая будет выполнять этот скрипт ежедневно в 2:00 ночи.

б) критерии оценивания компетенций (результатов) – правильная работа кода программы, понимание алгоритма метода оптимизации, умение вывести необходимые для алгоритма формулы.

в) описание шкалы оценивания:

Каждая задача оценивается по шкале от 0 до 10 баллов.

Контрольная работа считается выполненной успешно при суммарной оценке не ниже 18 баллов.

6.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Форма аттестации	Наименование оценочного средства	Баллы
Зачет (100 баллов)	Контрольная работа № 1	30
	Контрольная работа № 2	30
	Ответы на билет	40

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

а) основная учебная литература:

1. Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley, Dan Mackin – Linux Administration Handbook – 2020 – 1040 страниц
2. William Shotts – The Linux Command Line – 2020 – 504 страницы
3. Mark Sobell – A Practical Guide to Linux Commands, Editors, and Shell Programming – 2021 – 936 страниц
4. Richard Blum – Linux Command Line and Shell Scripting Bible – 2021 – 784 страницы
5. Chris Brown – The Linux Programming Interface – 2020 – 1520 страниц

б) дополнительная учебная литература:

1. James Turnbull – The Linux Bible – 2020 – 864 страницы
2. Sander van Vugt – Linux for Beginners – 2020 – 380 страниц
3. Jason Cannon – Linux for Beginners: An Introduction to the Linux Operating System and Command Line – 2020 – 128 страниц
4. Ravi Saive – Linux Command Line: A Complete Introduction – 2020 – 308 страниц
5. David Clinton – Mastering Linux Security and Hardening – 2021 – 614 страниц

8. Перечень ресурсов* информационно-телекоммуникационной сети «Интернет» (далее - сеть «Интернет»), необходимых для освоения дисциплины

-

9. Методические указания для обучающихся по освоению дисциплины

Вид учебного занятия	Организация деятельности студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначить вопросы, термины, материал, который вызывает трудности, пометить и попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации, на практическом занятии.
Практические занятия	Проработка рабочей программы, уделяя особое внимание целям и задачам, структуре и содержанию дисциплины. Работа с конспектом лекций, просмотр рекомендуемой литературы. Изучение выбранной предметной области на примерах решения задач семинарских занятий, индивидуальных домашних заданий.
Курсовая работа	Не предусмотрена
Контрольная работа	Ознакомиться с основной и дополнительной литературой, включая справочные издания, зарубежные источники, основополагающие термины. Попрактиковаться в решении аналогичных домашних задач по всем темам контрольных работ.
Лабораторная работа	Не предусмотрена.
Подготовка к зачету	При подготовке к зачету необходимо ориентироваться на конспекты лекций и рекомендуемую литературу.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Издательская система LaTeX для подготовки докладов, презентаций и учебного материала.

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Видеопроектор, компьютер, издательская система LaTeX для подготовки докладов, презентаций и учебного материала.

12. Иные сведения и (или) материалы

12.1. Перечень образовательных технологий, используемых при осуществлении образовательного процесса по дисциплине

Часов в интерактивной форме – 8.

В ходе практических занятий происходит публичное обсуждение каждой решаемой задачи. При этом студенты высказывают свои мнения по выбору наиболее простого способа поиска оптимального решения.

После решения домашних работ на консультациях проводится разбор допущенных студентами ошибок.

12.2. Формы организации самостоятельной работы обучающихся (темы, выносимые для самостоятельного изучения; вопросы для самоконтроля; типовые задания для самопроверки)

Некоторые темы изучаются студентами самостоятельно. Для изучения используется приведённая в списке основная и дополнительная литература. Контроль освоения материала осуществляется при проверке контрольных работ, домашнего задания и на зачете.

№	Тема и часть, изучаемая (осваиваемая) самостоятельно
1.1	Основы работы с системами на основе Linux (CentOS, Fedora, Debian)
1.2	Управление пакетами в Linux (APT, YUM, DNF, Snap)
1.3	Конфигурация и оптимизация системы
1.4	Безопасность Linux: файлы конфигурации, SELinux, AppArmor
1.5	Мониторинг и производительность системы (top, htop, iostat, vmstat, netstat)
1.6	Резервное копирование и восстановление данных в Linux (rsync, tar, dd)
1.7	Автоматизация задач с использованием Ansible, Puppet, или Chef
1.8	Работа с контейнерами в Linux (Docker, Podman)
1.9	Настройка веб-серверов (Apache, Nginx, Lighttpd)
1.10	Сетевые утилиты в Linux (ifconfig, netplan, ip, nslookup, traceroute)

Вопросы и задания для самоконтроля по всем темам:

1. Чем командная строка отличается от графического интерфейса пользователя?
2. Как установить пакет с помощью команды `sudo apt install`?
3. Что такое репозиторий в Linux и как добавить новый репозиторий в систему?
4. Как создать и запустить базовый `bash`-скрипт?
5. Как создать алиас для часто используемой команды в `bash`?
6. Какие команды используются для навигации по файловой системе в Linux?
7. Чем отличается символическая ссылка от жесткой в Linux? Как их создавать?
8. Как редактировать текстовые файлы в командной строке с использованием редакторов, таких как `папо` или `vim`?
9. Что такое конвейеры в Linux и как они используются для передачи данных между командами?
10. Как задать переменные в `bash` и использовать их в скриптах?
11. Как применяются регулярные выражения в `grep` для поиска текста в файлах?
12. Как выполнить замену текста в файле с использованием команды `sed`?
13. Как в `bash` устроены циклы и ветвления? Приведите пример их использования.
14. Как передаются аргументы командной строки в `bash`-скрипт и как их обработать?
15. Как проверить и изменить права доступа к файлам с помощью команд `chmod` и `chown`?

12.3. Краткий терминологический словарь

Командная строка (CLI)	Командная строка (Command Line Interface, CLI) — это текстовый интерфейс для взаимодействия с операционной системой, где пользователь вводит команды для выполнения различных задач, таких как управление файлами, запуск программ, администрирование системы и т. д.
Базовые скрипты (bash-скрипты)	Базовые скрипты — это текстовые файлы, содержащие последовательность команд, которые могут быть выполнены интерпретатором командной строки (например, bash). Эти скрипты позволяют автоматизировать процессы и выполнять повторяющиеся задачи.
Конвейер (Pipe):	Конвейер (pipe) — это механизм в командной строке Linux, позволяющий направлять вывод одной команды как ввод для другой. Символ используется для создания конвейера, что позволяет комбинировать команды и обрабатывать данные без необходимости использования временных файлов. Например, команда <code>ls grep "file"</code> покажет файлы, содержащие слово "file".
Символическая ссылка	Символическая ссылка (или symlink) — это специальный тип файла, который ссылается на другой файл или каталог в файловой системе. В отличие от жесткой ссылки, символическая ссылка может ссылаться на объекты в разных файловых системах и может указывать на несуществующие файлы (т.е. быть "битой").