

*Федеральное государственное автономное образовательное учреждение высшего образования
«Российский университет дружбы народов»*

Факультет физико-математических и естественных наук

Рекомендовано МССН
02.00.00 «Компьютерные и
информационные науки»

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Наименование дисциплины

Операционные системы

Рекомендуется для направления подготовки

02.03.01 – Математика и компьютерные науки

1. Цели и задачи дисциплины

Целью дисциплины является введение учащихся в предметную область современных операционных систем. В процессе преподавания дисциплины решаются следующие задачи: - анализ принципов построения и архитектур операционных систем; - обучение работе в операционной системе типа Unix.

2. Место дисциплины в структуре ОП ВО

Цикл, к которому относится дисциплина «Операционные системы»: Б1 «Дисциплины (модули)», обязательная часть.

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

Предшествующие и последующие дисциплины, направленные на формирование компетенций

Универсальные компетенции

Шифр и наименование компетенции	Предшествующие дисциплины	Последующие дисциплины (группы дисциплин)

Общепрофессиональные компетенции

Шифр и наименование компетенции	Предшествующие дисциплины	Последующие дисциплины (группы дисциплин)
ОПК-5	<ul style="list-style-type: none">Архитектура компьютеров	<ul style="list-style-type: none">Вычислительные системы, сети и телекоммуникациимодуль Системное администрирование и моделирование сетей передачи данныхИнформационная безопасность

Профессиональные компетенции (вид профессиональной деятельности: производственно-технологический)

Шифр и наименование компетенции	Предшествующие дисциплины	Последующие дисциплины (группы дисциплин)
ПК-3	<ul style="list-style-type: none">Архитектура компьютеров	<ul style="list-style-type: none">Вычислительные системы, сети и телекоммуникациимодуль Системное администрирование и моделирование сетей передачи данныхИнформационная безопасность

3. Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций: ОПК-5, ПК-3.

Расшифровка компетенций

- ОПК-5. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.
 - ОПК-5.1 Знает основные положения и концепции прикладного и системного программирования, архитектуры компьютеров и сетей (в том числе и глобальных), современные языки программирования, технологии создания и эксплуатации программных продуктов и программных комплексов
 - ОПК-5.2 Умеет использовать основные положения и концепции прикладного и системного программирования, архитектуры компьютеров и сетей (в том числе и глобальных), современные языки программирования, технологии создания и эксплуатации программных продуктов и программных комплексов в профессиональной деятельности
 - ОПК-5.3 Имеет практические навыки применения современных информационных технологий для решения задач профессиональной деятельности
- ПК-3 Способен осуществлять администрирование прикладного программного обеспечения, сетевой подсистемы и систем управления базами данных инфокоммуникационной системы организации
 - ПК-3.1 Знает основы архитектуры, устройства и функционирования информационно-вычислительных систем и сетевых подсистем инфокоммуникационной системы организации; методику установки и администрирования программных систем и сетевых подсистем инфокоммуникационной системы организации
 - ПК-3.2 Умеет настраивать и администрировать программные системы, сетевые подсистемы и базы данных инфокоммуникационной системы организации
 - ПК-3.3 Имеет практический опыт эксплуатации и администрирования программных систем, сетевых подсистем и баз данных инфокоммуникационной системы организации

В результате изучения дисциплины студент должен:

Знать:

- типы и виды современных операционных систем;
- концепции, принципы функционирования и конфигурирования современных операционных систем;
- архитектуру современных операционных систем.

Уметь:

- применять в профессиональной деятельности операционные системы;
- осуществлять элементарное конфигурирование операционных систем для использования в профессиональной деятельности.

Владеть:

- навыками установки и элементарной настройки операционных систем для использования в профессиональной деятельности;
- способностью использовать современные инструментальные средства, установленные под операционной системой.

4. Объем дисциплины и виды учебной работы

Общая трудоёмкость дисциплины составляет 4 зачетных единицы.

Вид учебной работы	Всего часов	Модуль
Аудиторные занятия (всего)	54	54
Лекции	18	18
Практические занятия (ПЗ)		
Семинары (С)		
Лабораторные работы (ЛР)	36	36
Самостоятельная работа (всего)	90	90
Общая трудоемкость	144	144
Зачётных единиц	4	4

5. Содержание дисциплины

5.1. Содержание разделов дисциплины

№	Наименование раздела дисциплины	Содержание раздела
1	Общие принципы ОС UNIX	<ul style="list-style-type: none">• Введение в операционную систему UNIX. Типы ОС.• Архитектура UNIX. Файлы и устройства.• Архитектура UNIX. Процессы.• Терминал и командная строка.
2	Начала администрирования ОС UNIX	<ul style="list-style-type: none">• Введение в безопасность UNIX.• Сеть в UNIX.• Управление службами.• Управление программным обеспечением.

5.2. Разделы дисциплин и виды занятий

№	Наименование раздела дисциплины	Лекц	Практ. зан.	Лаб. зан.	Сем.	СРС	Всего час.
1	Общие принципы ОС UNIX	9		18		45	72
2	Начала администрирования ОС UNIX	9		18		45	72
	Всего часов	18		36		90	144

6. Лабораторный практикум

№ раздела дисциплины	Наименование лабораторных работ	Трудоёмкость (час.)
1	Установка и конфигурация операционной системы на виртуальную машину	3
1	Система управления версиями Git	3

1	Обработка файлов в формате Markdown	3
1	Знакомство с операционной системой Linux	3
1	Основы интерфейса взаимодействия пользователя с системой Unix на уровне командной строки	3
1	Анализ файловой структуры UNIX. Команды для работы с файлами и каталогами	3
1	Поиск файлов. Перенаправление ввода-вывода. Просмотр запущенных процессов	3
2	Командная оболочка Midnight Commander	3
2	Текстовый редактор vi	3
2	Текстовый редактор emacs	3
2	Программирование в командном процессоре ОС UNIX. Командные файлы	3
2	Программирование в командном процессоре ОС UNIX. Ветвления и циклы	3
2	Программирование в командном процессоре ОС UNIX. Расширенное программирование	3
2	Средства для создания приложений в ОС UNIX	3
2	Именованные каналы	3
	Всего часов	36

7. Практические занятия (семинары)

Не предусмотрены.

8. Материально-техническое обеспечение дисциплины

Мультимедийная аудитория для проведения лекционных занятий. Компьютерные (дисплейные) классы с доступом к сети Интернет и электронно-образовательной среде Университета для выполнения обучающимися лабораторных работ по дисциплине, для проведения обучающимися самостоятельной работы и компьютерного тестирования обучающихся (при необходимости).

9. Информационное обеспечение дисциплины

Программное обеспечение:

- продукты Microsoft - операционная система, пакет офисных приложений, MS Teams и др. (подписка Enrollment for Education Solutions (EES)).
- ОС Linux.
- Офисный пакет LibreOffice (лицензия MPL-2.0).
- ПО для просмотра формата pdf (например, evince (лицензия GPL-2+ CC-BY-SA-3.0)).
- GNU Midnight Commander (Лицензия GNU GPL 3).
- Редактор emacs (лицензия GPL).
- Редактор vi (лицензия BSD).
- Компилятор gcc (лицензия GPL).
- Система управления версиями Git (Лицензия GNU GPL 2).

Базы данных, информационно-справочные и поисковые системы:

- Сайт библиотеки РУДН <http://lib.rudn.ru/>.
- Сайт ТУИС <http://esystem.pfur.ru/>.

10. Учебно-методическое обеспечение дисциплины

1. Основная литература

- Кулябов Д.С. Операционные системы: лабораторные работы: учебное пособие / Д.С. Кулябов, М.Н. Геворкян, А.В. Королькова, А.В. Демидова. — М. : Изд-во РУДН, 2016. — 117 с. — ISBN 978-5-209-07626-1 : 139.13; То же [Электронный ресурс]. — URL: <http://lib.rudn.ru/MegaPro2/Download/MObject/6118>.
- Робачевский А.М. Операционная система UNIX [текст] : Учебное пособие / А.М. Робачевский, С.А. Немнюгин, О.Л. Стесик. — 2-е изд., перераб. и доп. — СПб. : БХВ-Петербург, 2005, 2010. — 656 с. : ил. — ISBN 5-94157-538-6 : 164.56. (ЕТ 60)
- Таненбаум Эндрю. Современные операционные системы [Текст] / Э. Таненбаум. — 2-е изд. — СПб. : Питер, 2006. — 1038 с. : ил. — (Классика Computer Science). — ISBN 5-318-00299-4 : 446.05. (ЕТ 50)

2. Дополнительная литература

- Ван Стеен М., Эндрю Таненбаум Распределенные системы. Принципы и парадигмы [Текст] / Э. Таненбаум, в.М. Стеен. — СПб. : Питер, 2003. — 877 с. : ил. — (Классика Computer science). — ISBN 5-272-00053-6 : 377.52. (ЕТ 50)
- Сафонов, В.О. Основы современных операционных систем : учебное пособие / В.О. Сафонов. — Москва : Интернет-Университет Информационных Технологий, 2011. — 584 с. — (Основы информационных технологий). — ISBN 978-5-9963-0495-0 ; То же [Электронный ресурс]. — URL: <http://biblioclub.ru/index.php?page=book&id=233210>.
- Немет Эви. UNIX — руководство системного администратора [Текст] / Э. Немет, Г. Снайдер, С. Сибасс; Э.Немет, Г.Снайдер, С.Сибасс, Х.Р.Трент. — 3-е изд. — СПб. : Питер, 2004. — 925 с. : ил. — (Для профессионалов). — ISBN 0-13-020601-6. — ISBN 5-318-00754-6 : 280.00. (ЕТ 30)
- Бек Л. Введение в системное программирование [Текст] / Л. Бек; Пер. с англ. Н.А.Богомолова, В.М.Вязовского и С.Е.Морковина; Под ред. Л.Н.Королева. — М. : Мир, 1988. — 448 с. : ил. — ISBN 5-03-000011-9 : 2.60. (ЕТ 3)
- Дьяконов Владимир Юрьевич. Системное программирование [Текст] : Учебное пособие для вузов / В.Ю. Дьяконов, В.А. Китов, И.А. Калинин; Под ред. А.Л.Горелика. — М. : Высшая школа, 1990. — 221 с. : ил. — ISBN 5-06-000732-4 : 0.55.

11. Методические указания для обучающихся по освоению дисциплины

Учебным планом на изучение дисциплины отводится один модуль. В дисциплине предусмотрены контактные часы в виде лекций и лабораторного практикума. В течение модуля выполняются лабораторные работы, подготовка и презентация доклада и контрольные мероприятия.

11.1 Методические указания по самостоятельному освоению теоретического материала по дисциплине

Лекционный материал дисциплины охватывает темы, указанные в разделе 5.1 программы дисциплины. В ТУИС (<http://esystem.rudn.ru>) по темам лекций размещены презентации и видео записи лекций. Рекомендуется по указанным темам в дополнение изучить литературу, указанную в п. 10 программы дисциплины и учебно-методические материалы в ТУИС (<http://esystem.pfur.ru>).

11.2 Методические указания по выполнению лабораторных работ

- Задания по лабораторным работам выполняются индивидуально каждым студентом в дисплейных классах в соответствии с календарным планом и методическими указаниями по выполнению лабораторных работ по дисциплине.
- Часть лабораторных работ предусматривает задания для индивидуальной самостоятельной работы студента, обязательные для выполнения.
- Выполнение заданий для самостоятельной работы позволяет студенту приобрести дополнительные навыки и закрепить знания по изучаемой теме.
- По результатам выполнения каждой лабораторной работы студентом готовится отчёт. Отчёты в электронном виде сдаются студентом на проверку через соответствующие разделы ТУИС (<http://esystem.pfur.ru>).
- В качестве ответа на лабораторную работу в ТУИС необходимо загрузить:
 1. Ссылку на скринкаст с выполнением лабораторной работы (на *youtube*).
 2. Ссылку на скринкаст с презентацией лабораторной работы (на *youtube*).
 3. Ссылку на репозиторий на *Github* (Для лабораторных работ №2-№15).
 4. Следует загрузить отдельными файлами (не общим архивом):
 - Для лабораторных работ №1-№2: отчёт в docx; отчёт в pdf; презентацию в pdf.
 - Для лабораторных работ №3-№15: отчёт в markdown; отчёт в docx (сделанный из markdown); отчёт в pdf (сделанный из markdown); архив с исходными материалами markdown (текстовые файлы, скриншоты и т. д.); презентацию в pdf (сделанная из markdown); презентацию в markdown.
- Срок сдачи указан для каждой лабораторной работы. В случае сдачи лабораторной не в срок, ставится не более 50% от максимального балла.

11.3. Рекомендации по подготовке доклада

Доклад – это публичное развёрнутое изложение по заданной теме.

Целями подготовки доклада являются: - внесение знаний из дополнительной литературы; - систематизация материала по теме; - развитие навыков самостоятельной работы с литературой; - пробуждение познавательного интереса к научному познанию.

Основными задачами подготовки доклада являются: - выработка умений излагать содержание материала в короткое время; - выработка умений ориентироваться в материале и отвечать на вопросы; - выработка умений самостоятельно обобщать и представлять материал, делать выводы.

Доклад должен состоять из трех частей: вступление, основная часть и заключение.

Вступление должно содержать: название доклада, сообщение основной идеи, современную оценку предмета изложения, краткое перечисление рассматриваемых вопросов, форму изложения.

Основная часть должна раскрывать суть затронутой темы. Задача основной части - представить достаточно данных для того, чтобы слушатели заинтересовались темой и захотели ознакомиться с материалами. При этом логическая структура основного блока должна содержать наглядные материалы, аудио-визуальные или визуальные материалы (представление рисунков, таблиц графиков).

Заключение должно содержать ясное чёткое обобщение и краткие выводы.

Время доклада – 5–7 мин. Чтение доклада при выступлении запрещено.

11.4. Рекомендации по подготовке презентации доклада

Презентация представляет собой последовательность сменяющих друг друга слайдов. Количество слайдов пропорционально содержанию и продолжительности выступления. На первом слайде обязательно представляется тема выступления и сведения об авторе. На слайды помещается фактический и иллюстративный материал (таблицы, графики,

фотографии и пр.), который является уместным и достаточным средством наглядности, помогает в раскрытии стержневой идеи доклада.

В этом случае к слайдам предъявляются следующие требования: - выбранные средства визуализации информации (таблицы, схемы, графики и т. д.) соответствуют содержанию; - использованы иллюстрации хорошего качества (высокого разрешения), с четким изображением, максимальное количество графической информации на одном слайде – 2 рисунка (графики, схемы и т.д.) с текстовыми комментариями (не более 2 строк к каждому).

Наиболее важная информация должна располагаться в центре экрана. Обычный слайд, без эффектов анимации, должен демонстрироваться на экране не менее 10–15 секунд. Слайд с анимациями в среднем должен находиться на экране не меньше 40–60 секунд (без учета времени на случайно возникшее обсуждение). Для всех слайдов презентации необходимо использовать один и тот же шаблон оформления, кегль – для заголовков – не меньше 24 пунктов, для информации – для информации не менее 18. Наилучшей цветовой гаммой для презентации являются контрастные цвета фона и текста (белый фон – чёрный текст; темно-синий фон – светло-желтый текст и т. д.). Рекомендуется не злоупотреблять прописными буквами и не смешивать разные типы шрифтов в одной презентации.

12. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)

ФОС по дисциплине представлен в приложении к данной программе.
Программа составлена в соответствии с требованиями ОС ВО РУДН.

Разработчик:

профессор кафедры
прикладной информатики и теории вероятностей

Руководитель программы

Заведующий кафедрой
прикладной информатики и теории вероятностей




Д. С. Кулябов

К. Е. Самуйлов

*Федеральное государственное автономное образовательное учреждение высшего образования
«Российский университет дружбы народов»*

Факультет физико-математических и естественных наук

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Наименование дисциплины

Операционные системы

Рекомендуется для направления подготовки

02.03.01 – Математика и компьютерные науки

Паспорт фонда оценочных средств по дисциплине

Дисциплина: **Операционные системы**

Направление: 02.03.01 – Математика и компьютерные науки

Код контролируемой компетенции или её части	Контролируемый раздел дисциплины	Контролируемая тема дисциплины	A 1.1	A 1.2	A 2	A 3	A 4	Баллы темы	Баллы раздела	
ОПК-5, ПК-3	Общие принципы ОС UNIX	Введение в операционную систему UNIX.	6	2	12	1	10	28	52	
		Архитектура UNIX. Файлы и устройства	6	2		1		8		
		Архитектура UNIX. Процессы	6	2		1		8		
		Терминал и командная строка	6	2		1		8		
	Начала администрирования ОС UNIX	Введение в безопасность UNIX	6	2		1	10	28	48	
		Сеть в UNIX	6	2		1		8		
		Управление службами	6	2		1		8		
		Управление программным обеспечением	3	1		1		4		
		Итого:		45	15	12	8	20	100	100

Активности:

- А1.1: Лабораторные работы. Выполнение.
- А1.2: Лабораторные работы. Самопроверка и взаимопроверка.
- А2: Доклады по темам.
- А3: Тесты к лекциям.
- А4: Итоговый контроль знаний (тест).

Процесс изучения дисциплины направлен на формирование следующих компетенций: ОПК-5, ПК-3.

Расшифровка компетенций

- ОПК-5. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.
 - ОПК-5.1 Знает основные положения и концепции прикладного и системного программирования, архитектуры компьютеров и сетей (в том числе и гло-

- бальных), современные языки программирования, технологии создания и эксплуатации программных продуктов и программных комплексов
- ОПК-5.2 Умеет использовать основные положения и концепции прикладного и системного программирования, архитектуры компьютеров и сетей (в том числе и глобальных), современные языки программирования, технологии создания и эксплуатации программных продуктов и программных комплексов в профессиональной деятельности
- ОПК-5.3 Имеет практические навыки применения современных информационных технологий для решения задач профессиональной деятельности
- ПК-3 Способен осуществлять администрирование прикладного программного обеспечения, сетевой подсистемы и систем управления базами данных инфокоммуникационной системы организации
 - ПК-3.1 Знает основы архитектуры, устройства и функционирования информационно-вычислительных систем и сетевых подсистем инфокоммуникационной системы организации; методику установки и администрирования программных систем и сетевых подсистем инфокоммуникационной системы организации
 - ПК-3.2 Умеет настраивать и администрировать программные системы, сетевые подсистемы и базы данных инфокоммуникационной системы организации
 - ПК-3.3 Имеет практический опыт эксплуатации и администрирования программных систем, сетевых подсистем и баз данных инфокоммуникационной системы организации

Балльно-рейтинговая система оценки уровня знаний

Сводная оценочная таблица дисциплины

Раздел	Тема	A1.1	A1.2	A2	A3	A4	Баллы темы	Баллы раздела
P1	Введение в операционную систему UNIX	6	2	12	1	10	28	52
P1	Архитектура UNIX. Файлы и устройства	6	2		1		8	
P1	Архитектура UNIX. Процессы	6	2		1		8	
P1	Терминал и командная строка	6	2		1		8	
P2	Введение в безопасность UNIX	6	2		1	10	28	48
P2	Сеть в UNIX	6	2		1		8	
P2	Управление службами	6	2		1		8	
P2	Управление программным обеспечением	3	1		1		4	
Итого:		45	15	12	8	20	100	100

Разделы:

- P1: Общие принципы ОС UNIX
- P2: Начала администрирования ОС UNIX

Активности:

- A1.1: Лабораторные работы. Выполнение.
- A1.2: Лабораторные работы. Самопроверка и взаимопроверка.
- A2: Доклады по темам.
- A3: Тесты к лекциям.
- A4: Итоговый контроль знаний (тест).

Таблица соответствия баллов и оценок

Баллы БРС	Традиционные оценки в РФ	Баллы для перевода оценок	Оценки	Оценки ECTS
86–100	5	95–100	5+	A
		86–94	5	B

69–85	4	69–85	4	C
51–68	3	61–68	3+	D
		51–60	3	E
0–50	2	31–50	2+	FX
		0–30	2	F

Правила применения БРС

- Раздел (тема) учебной дисциплины считаются освоенными, если студент набрал более 50% от возможного числа баллов по этому разделу (теме).
- Студент не может быть аттестован по дисциплине, если он не освоил все темы и разделы дисциплины, указанные в сводной оценочной таблице дисциплины.
- По решению преподавателя и с согласия студентов, не освоивших отдельные разделы (темы) изучаемой дисциплины, в течение учебного семестра могут быть повторно проведены мероприятия текущего контроля успеваемости или выданы дополнительные учебные задания по этим темам или разделам. При этом студентам за данную работу засчитывается минимально возможный положительный балл (51% от максимального балла).
- При выполнении студентом дополнительных учебных заданий или повторного прохождения мероприятий текущего контроля полученные им баллы засчитываются за конкретные темы. Итоговая сумма баллов не может превышать максимального количества баллов, установленного по данным темам.
- График проведения мероприятий текущего контроля успеваемости формируется в соответствии с календарным планом курса. Студенты обязаны сдавать все задания в сроки, установленные преподавателем.
- Время, которое отводится студенту на выполнение мероприятий текущего контроля успеваемости, устанавливается преподавателем. По завершении отведённого времени студент должен сдать работу преподавателю, вне зависимости от того, завершена она или нет.
- Использование источников (в том числе конспектов лекций и лабораторных работ) во время выполнения контрольных мероприятий возможно только с разрешения преподавателя.
- Отсрочка в прохождении мероприятий текущего контроля успеваемости считается уважительной только в случае болезни студента, что подтверждается наличием у него медицинской справки, заверенной круглой печатью КДЦ РУДН, предоставляемой преподавателю не позднее двух недель после выздоровления. В этом случае выполнение контрольных мероприятий осуществляется после выздоровления студента в срок, назначенный преподавателем. В противном случае отсутствие студента на контрольном мероприятии признается неуважительным.
- Если в итоге за семестр студент получил 0–50 баллов, то студенту разрешается добор необходимого (до 51) количества баллов путём повторного одноразового выполнения предусмотренных контрольных мероприятий, при этом по усмотрению преподавателя аннулируются соответствующие предыдущие результаты. Ликвидация задолженностей проводится в сроки, согласованные с деканатом.
- Баллы за доклады по темам фиксируются только после доклада во время контактных часов.
- Оценки за лабораторные работы состоят из оценки за выполнение лабораторной работы и оценки за совместное оценивание лабораторной работы.
- Доклады:
 - Темы докладов распределены по лекциям.

- При представлении темы после лекции, к которой она привязана, оценка снижается.
- Оценка формируется из следующих элементов:
 - оформление презентации;
 - как сделан доклад;
 - содержание доклада;
 - оформление доклада.
- Оценка выставляется только после выкладывания на сайт презентации и текста доклада.
- Для получения оценки обязательно представление презентации во время соответствующего лекционного занятия.

Примерный перечень оценочных средств

по дисциплине Операционные системы

Аудиторная работа

Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
Лабораторная работа	Система практических заданий, направленных на формирование практических навыков у обучающихся	Фонд практических заданий
Презентация (защита) доклада	Средство контроля способностей обучающихся представить перед аудиторией результаты проделанной работы	Темы докладов

Самостоятельная работа

Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
Подготовка отчетов по результатам выполнения лабораторных работ	Форма проверки качества выполнения студентами лабораторных работ в соответствии с утвержденной программой	Фонд практических заданий
Доклад	Продукт самостоятельной работы студента, представляющий собой публичное выступление по представлению полученных результатов решения определенной учебно-практической, учебно-исследовательской или научной темы	Темы докладов

Учебным планом на изучение дисциплины отводится один модуль. В дисциплине предусмотрен контактные часы в форме лабораторного практикума, контрольные мероприятия по проверке отчетов по лабораторным работам, подготовка и презентация доклада. Оценка ставится по результатам работы в семестре.

Оценивание результатов освоения дисциплины производится в соответствии с балльно-рейтинговой системой.

Критерии оценки по дисциплине

95–100 баллов:

- полное и своевременное выполнение на высоком уровне лабораторных работ с оформлением отчетов, успешное прохождение контрольных мероприятий, предусмотренных программой курса;
- систематизированное, глубокое и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- использование научной терминологии, стилистически грамотное, логически правильное изложение ответов на вопросы, умение делать обоснованные выводы;
- безупречное владение программным обеспечением, умение эффективно использовать его в постановке и решении научных и профессиональных задач;
- выраженная способность самостоятельно и творчески решать поставленные задачи;
- полная самостоятельность и творческий подход при изложении материала по программе дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной программой дисциплины и преподавателем.

86–94 балла:

- полное и своевременное выполнение на хорошем уровне лабораторных работ с оформлением отчетов, успешное прохождение контрольных мероприятий, предусмотренных программой курса;
- систематизированное, глубокое и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- использование научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- хорошее владение программным обеспечением, умение эффективно использовать его в постановке и решении научных и профессиональных задач;
- способность самостоятельно решать поставленные задачи в нестандартных производственных ситуациях;
- усвоение основной и дополнительной литературы, нормативных и законодательных актов, рекомендованных программой дисциплины и преподавателем.

69–85 баллов:

- своевременное выполнение на хорошем уровне лабораторных работ с оформлением отчетов, прохождение контрольных мероприятий, предусмотренных программой курса;
- хороший уровень культуры исполнения лабораторных работ;
- систематизированное и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- владение программным обеспечением, умение использовать его в постановке и решении научных и профессиональных задач;
- способность самостоятельно решать проблемы в рамках программы дисциплины;
- усвоение основной литературы;

51–68 баллов:

- выполнение на удовлетворительном уровне лабораторных работ с оформлением отчетов, прохождение контрольных мероприятий, предусмотренных программой курса;
- систематизированное и полное освоение навыков и компетенций по всем разделам программы дисциплины;

- удовлетворительное владение программным обеспечением, умение использовать его в постановке и решении научных и профессиональных задач;
- способность решать проблемы в рамках программы дисциплины;
- удовлетворительное усвоение основной литературы;

31–50 баллов – НЕ ЗАЧТЕНО:

- не выполнение, несвоевременное выполнение или выполнение на неудовлетворительном уровне лабораторных работ, не прохождение контрольных мероприятий, предусмотренных программой курса;
- недостаточно полный объем навыков и компетенции в рамках программы дисциплины;
- неумение использовать в практической деятельности научной терминологии, изложение ответа на вопросы с существенными стилистическими и логическими ошибками;
- слабое владение программным обеспечением по разделам программы дисциплины, некомпетентность в решении стандартных (типовых) производственных задач;
- способность решать проблемы в рамках программы дисциплины;
- удовлетворительное усвоение основной литературы;

0–30 баллов – НЕ ЗАЧТЕНО:

- отсутствие умений, навыков, знаний и компетенции в рамках программы дисциплины;
- невыполнение лабораторных заданий, не прохождение контрольных мероприятий, предусмотренных программой курса; отказ от ответов по программе дисциплины;
- игнорирование занятий по дисциплине по неуважительной причине.

Темы докладов

- Введение в операционную систему UNIX.
 - История зарубежных операционных систем.
 - История отечественных операционных систем.
 - История и диалекты операционной системы UNIX.
 - Поддержка мультипрограммирования и разделения времени в операционных системах.
 - Обзор и сравнительный анализ операционных систем Plan9 и Inferno.
 - Обзор операционной системы FreeBSD.
 - Обзор дистрибутива Linux Gentoo.
 - Сравнительный анализ линейки операционных систем семейства Windows.
 - Сравнительный анализ дистрибутивов Linux: Ubuntu, Fedora.
 - Обзор фреймовых оконных менеджеров для linux (ion, awesome и т.д.).
 - Особенности построения серверных операционных систем.
 - Основные производители серверных операционных систем.
 - Основные производители клиентских операционных систем.
 - Кластерные операционные системы.
 - Обзор коммерческих Unix-операционных систем различных производителей.
 - Обзор свободно распространяемых Unix-операционных систем различных производителей.
 - Обзор Linux-операционных систем различных производителей.
 - Операционные системы реального времени.
 - Обзор стандартов, регламентирующих разработку операционных систем.
 - Операционные системы многопроцессорных компьютеров.
 - Виртуальные машины и их операционные системы.
 - Особенности построения сетевых операционных систем.
- Архитектура UNIX. Файлы и устройства.
 - Обзор концепции файла и типов файлов в файловых системах.
 - Обзор расширений имён файлов.
 - Обзор методов доступа к файлам.
 - Обзор операций над директориями и методов реализации директорий в файловых системах.
 - Обзор концепции монтирования и методов монтирования файловых систем в операционных системах.
 - Обзор виртуальных файловых систем.
 - Файловая система FAT.
 - Файловые системы с индексированным размещением файлов.
 - Обзор системы NFS.
 - Обзор видов внешних устройств в современных компьютерах.
 - Обзор видов прерываний и методов их обработки в современных компьютерах.
 - Метод DMA и DMA-контроллеры.
 - Обзор архитектур реализаций ввода-вывода в операционных системах.
 - Обзор сетевых устройств в компьютерных системах.
 - Библиотеки sys и os языка python для работы с файловой системой (манипуляция с файлами и директориям).
- Архитектура UNIX. Процессы.
 - Понятие процесса, виды процессов и его состояния в операционных системах.

- Реализация системных структур ОС для управления процессами (блок управления процессом, системные очереди) и операций над ними.
- Обзор подходов к созданию дочерних процессов и разделению ресурсов дочерним и родительским процессами в операционных системах.
- Обзор алгоритмов планирования в Linux.
- Обзор алгоритмов планирования в Windows.
- Планирование и диспетчеризация процессоров в системе Solaris.
- Планирование и диспетчеризация процессоров в системе Mac OS.
- История синхронизации процессов.
- Сравнение возможностей достоинств и недостатков различных средств синхронизации процессов.
- Концепция семафора и ее использование для синхронизации процессов.
- Концепция монитора и ее использование для синхронизации процессов.
- Концепция критической области и ее использование для синхронизации процессов.
- Обзор методов откатки и подкачки в операционных системах.
- Обзор стратегий распределения основной памяти.
- Обзор архитектур таблиц страниц.
- Обзор методов сегментно-страничной организации памяти в компьютерных системах.
- Методы управления памятью в операционных системах.
- Методы адресации в компьютерных системах.
- Этапы обработки программы от исходного кода к исполняемому в операционных системах.
- Физическая и логическая адресация в компьютерных системах.
- Организация оверлейной структуры программ.
- Обзор методов реализации виртуальной памяти в операционных системах.
- Страничная организация по требованию.
- Сегментная организация по требованию.
- Сравнительный анализ алгоритмов замещения страниц.
- Межпроцессное взаимодействие с использованием Unix сокетов и сетевых сокетов.
- Межпроцессное взаимодействие: очереди сообщений, разделяемая память и семафоры.
- Межпроцессное взаимодействия на базе протокола 9P20- Реализация с помощью библиотек r9p.
- Межпроцессные взаимодействия на базе протокола plumbing. Принципы работы.
- Межпроцессное взаимодействия на базе протокола 9P20- Реализация с помощью POSIX совместимых библиотек.
- Терминал и командная строка.
 - Настройка csh. Специфика по сравнению с bash.
 - Настройка zsh. Специфика по сравнению с bash.
 - Настройка tcsh. Специфика по сравнению с bash.
 - Сравнение систем управления версиями git, mercurial и bazaar.
 - Применение awk для разбора текстовых файлов с данными.
 - Утилиты-мультиплексоры screen и tmux.
 - Использование rsync для резервного копирования данных.
 - Планировщик задач cron и ему подобные. Настройка и примеры заданий.
- Введение в безопасность UNIX.
 - Методы организации безопасности в операционных системах.

- Современные методы криптографии.
- Trustworthy Computing Initiative.
- Схема аутентификации пользователей с помощью логинов и паролей.
- Методы защиты от атаки типа переполнение буфера.
- Проверка неизменности системных файлов и директорий.
- Методы шифрования на основе закрытого ключа.
- Методы шифрования на основе открытого ключа.
- Обзор методов защиты файлов в операционных системах.
- Сеть в UNIX.
 - Обзор архитектур распределенных систем и локальных сетей.
 - Обзор сетевых топологий.
 - Архитектура и организация локальных сетей.
 - Архитектура и организация глобальных сетей.
 - Архитектура, параметры и организация беспроводных сетей.
 - Обзор стратегий маршрутизации и маршрутизаторов.
 - Архитектура и функционирование DNS.
 - Методы сетевых соединений.
 - Методы разрешения коллизий в сетях.
 - Сетевые возможности Linux.
 - Настройка сервера в Linux.
 - Сравнительный анализ сетевых возможностей Windows и Linux.
 - Методика работы пользователя в сети Linux.
- Управление службами.
 - Система инициализации SystemV.
 - Система инициализации SystemD.
 - Система инициализации Upstart.
 - Система инициализации Openrc.
 - Система инициализации Runinit.
 - Система инициализации Launchd.
- Управление программным обеспечением.
 - Система управления пакетами rpm. Управление зависимостями: yum, dnf.
 - Система управления пакетами dpkg. Управление зависимостями: aptitude, apt.
 - Система управления пакетами packman.
 - Система управления пакетами portage.
 - Система управления пакетами NuGet.
 - Система управления пакетами Chocolatey.

Методические указания и шкала оценок.

Рекомендации по подготовке доклада.

Доклад – это публичное развёрнутое изложение по заданной теме.

Целями подготовки доклада являются: - внесение знаний из дополнительной литературы; - систематизация материала по теме; - развитие навыков самостоятельной работы с литературой.

Основными задачами подготовки доклада являются: - выработка умений излагать содержание материала в короткое время; - выработка умений ориентироваться в материале и отвечать на вопросы; - выработка умений самостоятельно обобщать и представлять материал, делать выводы.

Доклад должен состоять из трех частей: вступление, основная часть и заключение.

Выступление должно содержать: название доклада, сообщение основной идеи, современную оценку предмета изложения, краткое перечисление рассматриваемых вопросов. Основная часть состоит из нескольких разделов, постепенно раскрывающих тему. Возможно использование иллюстрации (графики, диаграммы, фотографии, карты, рисунки). Для обоснования темы используется ссылка на источники с доказательствами, взятыми из литературы. Изложение материала должно быть связным, последовательным и доказательным. Способ изложения материала должен носить конспективный или тезисный характер. Заключение должно содержать ясное четкое обобщение и краткие выводы.

Время доклада – 5-7 мин (2-5 машинописных листа текста с докладом). Чтение доклада при выступлении – запрещено.

Рекомендации по подготовке презентации доклада.

Презентация представляет собой последовательность сменяющих друг друга слайдов. Количество слайдов пропорционально содержанию и продолжительности выступления. На первом слайде обязательно представляется тема выступления и сведения об авторе. На слайды помещается фактический и иллюстративный материал (таблицы, графики, фотографии и пр.), который является уместным и достаточным средством наглядности, помогает в раскрытии стержневой идеи доклада.

В этом случае к слайдам предъявляются следующие требования: - выбранные средства визуализации информации (таблицы, схемы, графики и т. д.) соответствуют содержанию; - использованы иллюстрации хорошего качества (высокого разрешения), с четким изображением, максимальное количество графической информации на одном слайде – 2 рисунка (графики, схемы и т.д.) с текстовыми комментариями (не более 2 строк к каждому).

Наиболее важная информация должна располагаться в центре экрана. Обычный слайд, без эффектов анимации, должен демонстрироваться на экране не менее 10–15 секунд. Слайд с анимациями в среднем должен находиться на экране не меньше 40–60 секунд (без учета времени на случайно возникшее обсуждение). Для всех слайдов презентации необходимо использовать один и тот же шаблон оформления, кегль – для заголовков – не меньше 24 пунктов, для информации – для информации не менее 18. Наилучшей цветовой гаммой для презентации являются контрастные цвета фона и текста (белый фон – чёрный текст; темно-синий фон – светло-желтый текст и т. д.). Рекомендуется не злоупотреблять прописными буквами и не смешивать разные типы шрифтов в одной презентации.

Критерии оценки

Оценивается содержание доклада, качество подготовки презентации, качество изложения материала, качество ответов на вопросы

Фонд практических (лабораторных) заданий

по дисциплине Операционные системы

Предлагаются к выполнению 15 лабораторных работ. Отчёты по лабораторным работам выполняются студентом самостоятельно, на лабораторном занятии студент может получить консультацию и методические указания от преподавателя.

Лабораторная работа № 1. Установка и конфигурация операционной системы на виртуальную машину

1. Пользуясь методическими указаниями установить и сконфигурировать операционную систему на виртуальную машину.

Лабораторная работа № 2. Система управления версиями Git

1. Настройка git.
 1. Создайте учётную запись на <https://github.com>.
 2. Настройте систему контроля версий git с использованием сервера репозитория <https://github.com/>.
 3. Создайте структуру каталога лабораторных работ.
2. Подключение репозитория к github
 1. Создайте репозиторий на GitHub. Для примера назовём его os-intro.
 2. Рабочий каталог будем обозначать как laboratory.
 3. Вначале нужно перейти в этот каталог
 4. Инициализируем системы git
 5. Создаём заготовку для файла README.md
 6. Делаем первый коммит и выкладываем на github
3. Первичная конфигурация
 1. Добавим файл лицензии
 2. Добавим шаблон игнорируемых файлов.
 3. Просмотрим список имеющихся шаблонов
 4. Добавим новые файлы
 5. Выполним коммит
 6. Отправим на github.
4. Конфигурация git-flow
 1. Инициализируем git-flow git.
 2. Проверьте, что Вы на ветке develop.
 3. Создадим релиз с версией 1.0.0
 4. Запишем версию
 5. Добавим в индекс
 6. Зальём релизную ветку в основную ветку
 7. Отправим данные на github
 8. Создадим релиз на github.

Лабораторная работа № 3. Обработка файлов в формате Markdown

1. Сделайте отчёт по предыдущей лабораторной работе в формате Markdown.
2. В качестве отчёта просьба предоставить отчёты в 3 форматах: pdf, docx и md (в архиве, поскольку он должен содержать скриншоты, Makefile ит.д.)

Лабораторная работа № 4. Знакомство с операционной системой Linux

1. Ознакомиться с теоретическим материалом.
2. Загрузить компьютер.

3. Перейти на текстовую консоль. Сколько текстовых консолей доступно на вашем компьютере?
4. Перемещаться между текстовыми консолями. Какие комбинации клавиш необходимо при этом нажимать?
5. Зарегистрироваться в текстовой консоли операционной системы.
6. Какой логин вы при этом использовали? Какие символы отображаются при вводе пароля?
7. Завершить консольный сеанс. Какую команду или комбинацию клавиш необходимо для этого использовать?
8. Переключиться на графический интерфейс. Какую комбинацию клавиш для этого необходимо нажать?
9. Ознакомиться с менеджером рабочих столов. Как называется менеджер, запускаемый по умолчанию?
10. Поочередно зарегистрироваться в разных графических менеджерах рабочих столов (GNOME, KDE, XFCE) и оконных менеджерах (Openbox). Продемонстрировать разницу между ними сделав снимки экрана (скриншоты). Какие графические менеджеры установлены на вашем компьютере?
11. Изучить список установленных программ. Обратит внимание на предпочтительные программы для разных применений. Запусти те поочередно браузер, текстовый редактор, текстовый процессор, эмулятор консоли. Укажите названия программ.

Лабораторная работа № 5. Основы интерфейса взаимодействия пользователя с системой Unix на уровне командной строки

1. Определите полное имя вашего домашнего каталога. Далее относительно этого каталога будут выполняться последующие упражнения.
2. Выполните следующие действия: 2.1. Перейдите в каталог /tmp. 2.2. Выведите на экран содержимое каталога /tmp. Для этого используйте команду ls с различными опциями. Поясните разницу в выводимой на экран информации. 2.3. Определите, есть ли в каталоге /var/spool подкаталог с именем cron? 2.4. Перейдите в Ваш домашний каталог и выведите на экран его содержимое. Определите, кто является владельцем файлов и подкаталогов?
3. Выполните следующие действия: 3.1. В домашнем каталоге создайте новый каталог с именем newdir. 3.2. В каталоге ~/newdir создайте новый каталог с именем morefun. 3.3. В домашнем каталоге создайте одной командой три новых каталога с именами letters, memos, misk. Затем удалите эти каталоги одной командой. 3.4. Попробуйте удалить ранее созданный каталог ~/newdir командой rm. Проверьте, был ли каталог удалён. 3.5. Удалите каталог ~/newdir/morefun из домашнего каталога. Проверьте, был ли каталог удалён.
4. С помощью команды man определите, какую опцию команды ls нужно использовать для просмотра содержимое не только указанного каталога, но и подкаталогов, входящих в него.
5. С помощью команды man определите набор опций команды ls, позволяющий отсортировать по времени последнего изменения выводимый список содержимого каталога с развёрнутым описанием файлов.
6. Используйте команду man для просмотра описания следующих команд: cd, pwd, mkdir, rmdir, rm. Поясните основные опции этих команд.
7. Используя информацию, полученную при помощи команды history, выполните модификацию и исполнение нескольких команд из буфера команд.

Лабораторная работа № 6. Анализ файловой структуры UNIX. Команды для работы с файлами и каталогами

1. Осуществите вход в систему, используя соответствующее имя пользователя.
2. Выполните все примеры, приведенные в первой части описания лабораторной работы.
3. Выполните следующие действия. Запишите в отчет по лабораторной работе, используемые при этом команды.
 - (a) Скопируйте файл `/usr/include/sys/io.h` в ваш домашний каталог и назовите его `equipment`. Если такого файла нет используйте любой ругой файл вместо него.
 - (b) В домашнем каталоге создайте директорию `~/ski.places`.
 - (c) Переместите файл `equipment` в каталог `~/ski.places`.
 - (d) Переименуйте файл `~/ski.places/equipment` в `~/ski.places/equiplist`.
 - (e) Скопируйте файл `abc1` из вашего домашнего каталога в каталог `~/ski.places` и назовите его `equiplist2`.
 - (f) Создайте каталог с именем `equipment` в каталоге `~/ski.places`.
 - (g) Переместите файлы `~/ski.places/equiplist` и `equiplist2` в каталог `~/ski.places/equipment`.
 - (h) Создайте и переместите каталог `~/newdir` в каталог `~/ski.places` и назовите его `plans`.
4. Определите опции команды `chmod`, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет:
 - (a) `drwxr-g- ... australia`
 - (b) `drwx-x-x ... play`
 - (c) `-r-xr-g- ... my_os`
 - (d) `-rw-rw-r- ... feathers` При необходимости создайте нужные файлы.
5. Прodelайте приведенные ниже упражнения, записывая в отчет по лабораторной работе, используемые при этом команды:
 - (a) Просмотрите содержимое файла `/etc/password`.
 - (b) Скопируйте файл `~/feathers` в файл `~/file.old`.
 - (c) Переместите файл `~/file.old` в каталог `~/play`.
 - (d) Скопируйте каталог `~/play` в каталог `~/fun`.
 - (e) Переместите каталог `~/fun` в каталог `~/play` и назовите его `games`.
 - (f) Лишите владельца файла `~/feathers` права на чтение.
 - (g) Что произойдет, если вы попытаетесь просмотреть файл `~/feathers` командой `cat`?
 - (h) Что произойдет, если вы попытаетесь скопировать файл `~/feathers`?
 - (i) Дайте владельцу файла `~/feathers` право на чтение.
 - (j) Лишите владельца каталога `~/play` права на выполнение.
 - (k) Перейдите в каталог `~/play`. Что произошло?
 - (l) Дайте владельцу каталога `~/play` право на выполнение.
6. Прочитайте `man` по командам `mount`, `fsck`, `mkfs`, `kill` и кратко их охарактеризуйте.

Лабораторная работа № 7. Поиск файлов. Перенаправление ввода-вывода. Просмотр запущенных процессов

1. Осуществите вход в систему, используя соответствующее имя пользователя.
2. Запишите в файл `file.txt` названия файлов, содержащихся в каталоге `/etc`. Допишите в этот же файл названия файлов, содержащихся в вашем домашнем каталоге.
3. Выведите имена всех файлов из `file.txt` имеющих расширение `.conf`, после чего запишите их в новый текстовый файл `conf.txt`.

4. Определите, какие файлы в вашем домашнем каталоге имеют имена, начинавшиеся с символа `s`? Предложите несколько вариантов как это сделать.
5. Выведите на экран (по странично) имена файлов из каталога `/etc`, начинающиеся с символа `h`.
6. Запустите в фоновом режиме процесс, который будет записывать в файл `~/logfile` файлы, имена которых начинаются с `log`.
7. Удалите файл `~/logfile`.
8. Запустите из консоли в фоновом режиме редактор `gedit`.
9. Определите идентификатор процесса `gedit` используя команду `ps`, конвейер и фильтр `grep`. Можно ли определить этот идентификатор более простым способом?
10. Прочтите справку (`man`) команды `kill` после чего используйте ее для завершения процесса `gedit`.
11. Выполните команды `df` и `du`, предварительно получив более подробную информацию об этих командах, с помощью команды `man`.
12. Воспользовавшись справкой команды `find` выведите имена всех директорий, имеющих в вашем домашнем каталоге.

Лабораторная работа № 8. Командная оболочка Midnight Commander

1. Изучите структуру экрана после запуска программы `mc`.
2. Откройте дополнительную сессию. Получите на этом экране информацию о `mc` (`man mc`).
3. Выполните простейшие операции, используя управляющие клавиши для программы `mc` (активизация окна; перемещение курсора в окне, выбор группы файлов, отказ от выбора, получение информации об объёме файлового пространства, занимаемого выбранной группой файлов и т.п.)
4. Вызовите основное меню (клавиши `F9` или `ESC9`). Выполните анализ основного меню.
5. Выполните основные подкоманды для левого (правого) окна (степень подробности информации о файлах).
6. Используя возможности подменю `FILES`, выполните:
 - просмотр текстового файла,
 - редактирование текстового файла (без сохранения результатов редактирования),
 - создание поддиректории `STUD1`,
 - копирование в поддиректорию `STUD1` файла.
7. С помощью соответствующих средств подменю `COMMAND` осуществите:
 - поиск в файловой системе файла (по указанию преподавателя),
 - выбор и повторение одной из предыдущих команд,
 - выход в домашний каталог,
 - анализ файла меню и файла расширений.
8. Вызовите подменю `OPTIONS`. Освойте операции, определяющие структуру экрана `mc` (`Full screen`, `Double Width`, `Show Hidden Files` и т.д.)
9. Выполните анализ вашего домашнего каталога. Удалите все лишние файлы и каталоги. Определите объем памяти, занимаемый вашим домашним каталогом до и после чистки.

Лабораторная работа №9. Текстовый редактор vi

1. Ознакомиться с теоретическим материалом.
2. Ознакомиться с редакторами `mcedit`, `vi`, `emacs`.
3. Выполнить упражнения.

Задание 1. Создание нового файла с использованием Vi

1. Создайте каталог с именем `~/work/os/lab06`.

2. Перейдите во вновь созданный каталог.
3. Вызовите Vi и создайте файл hello.sh vi hello.sh
4. Нажмите клавишу i и вводите следующий текст.

```
#!/bin/bash HELL=Hello function hello {
LOCAL HELLO=World
echo $HELLO
}
echo $HELLO hello
```
5. Нажмите клавишу ESC для перехода в командный режим после завершения ввода текста.
6. Нажмите : (двоеточие) для перехода в режим последней строки и внизу вашего экрана появится приглашение в виде двоеточия.
7. Нажмите w (записать) и q (выйти), а затем нажмите клавишу для сохранения вашего текста и завершения работы.
8. Сделайте файл исполняемым chmod +x hello.sh

Задание 2. Редактирование существующего файла

1. Вызовите Vi на редактирование файла vi ~/work/os/lab06/hello.sh
2. Установите курсор на конец слова HELL второй строки.
3. Перейдите в режим вставки и замените на HELLO. Нажмите , чтобы вернуться в командный режим.
4. Установите курсор на четвертую строку и сотрите слово LOCAL.
5. Перейдите в режим вставки и наберите следующий текст: local нажмите ESC, чтобы вернуться в командный режим.
6. Установите курсор на последней строке файла. Вставьте после неё строку, содержащую следующий текст: echo \$HELLO.
7. Нажмите , чтобы перейти в командный режим.
8. Удалите последнюю строку.
9. Введите команду отмены изменений u для отмены последней команды.
10. Введите символ : (двоеточие) для перехода в режим последней строки. Запишите произведенные изменения на диск и выйдите из vi.

Лабораторная работа №10. Текстовый редактор emacs

1. Открыть emacs.
2. Создать файл lab1.sh C-x C-f.
3. Внести текст:

```
#!/bin/bash HELL=Hello function hello {
LOCAL HELLO=World
echo $HELLO
}
echo $HELLO hello
```
4. Сохранить файл C-x C-s.
5. Прodelать с текстом стандартные процедуры редактирования, каждое действие должно осуществляться комбинацией клавиш.
 - (a) Вырезать одной командой целую строку C-k.
 - (b) Вставить эту строку в конец файла C-y.
 - (c) Выделить область текста C-space.
 - (d) Скопировать область в буфер обмена M-w.
 - (e) Вставить область в конец файла.
 - (f) Вновь выделить эту область и на этот раз вырезать ее C-w.
 - (g) Отмените последнее действие C-/_.
6. Научитесь использовать команды по перемещению курсора.
 - (a) Переместите курсор в начало строки C-a.

- (b) Переместите курсор в конец строки C-e.
 - (c) Переместите курсор в начало буфера M-<.
 - (d) Переместите курсор в конец буфера M->.
7. Управление буферами.
- (a) Вывести список активных буферов на экран C-x C-b.
 - (b) Переместитесь во вновь открытое окно C-x o со списком открытых буферов и переключитесь на другой буфер.
 - (c) Закройте это окно C-x 0.
 - (d) Теперь вновь переключайтесь между буферами, но уже без вывода их списка на экран C-x b.
8. Управление окнами.
- (a) Поделите фрейм на 4 часть: разделите фрейм на два окна по вертикали C-x 3, а затем каждое из этих окон на две части по горизонтали C-x 2.
 - (b) В каждом из четырех созданных окон откройте новый буфер (файл) и введите несколько строк текста.
9. Режим поиска
- (a) Переключитесь в режим поиска и C-s найдите несколько слов, присутствующих в тексте.
 - (b) Переключайтесь между результатами поиска, нажимая C-s.
 - (c) Выйдите из режима поиска, нажав C-g.
 - (d) Перейдите в режим поиска и замены M-% введите текст, который следует найти и заменить, нажмите Enter, затем введите текст для замены. После того, как будут подсвечены результаты поиска нажмите ! для подтверждения замены.
 - (e) Испробуйте другой режим поиска, нажав M-s o. Объяснит, чем он отличается от обычного режима?

Лабораторная работа № 11. Программирование в командном процессоре ОС UNIX. Командные файлы

1. Написать скрипт, который будет при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую backup в вашем домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор zip, bzip2 или tar. Способ использования команд архивации узнать изучив справку.
2. Написать пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов.
3. Написать командный файл — аналог команды ls (без использования самой этой команды и команды dir). Требуется, что бы он выдавал информацию о нужном каталоге и выводил информацию о возможностях доступа к файлам этого каталога.
4. Написать командный файл, который получает в качестве аргумента командной строки формат файла (.txt, .doc, .jpg, .pdf и т.д.) и вычисляет количество таких файлов в указанной директории. Путь к директории также передается в виде аргумента командной строки.

Лабораторная работа № 12. Программирование в командном процессоре ОС UNIX. Ветвления и циклы

1. Используя команды getoptс grep написать командный файл, который анализирует командную строку с ключами:
 - -inputfile — прочитать данные из указанного файла;
 - -outputfile — вывести данные в указанный файл;
 - -ршаблон — указать шаблон для поиска;

- -C — различать большие и малые буквы;
 - -n — выдавать номера строк; а затем ищет в указанном файле нужные строки, определяемые ключом -r.
2. Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в о коде завершения в оболочку. Командный файл должен вызывать эту программу и проанализировав с помощью команды `$?`. Код возврата, выдает сообщение о том какое число было введено.
 3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N (например, 1.tmp, 2.tmp, 3.tmp,4.tmp и т.д.). Число файлов, которые необходимо создать передается в аргумента командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).
 4. Написать командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду `find`).

Лабораторная работа № 13. Программирование в командном процессоре ОС UNIX. Расширенное программирование

1. Написать командный файл, реализующий упрощенный механизм семафоров. Командный файл должен в течение некоторого времени `t1` дожидаться освобождения ресурса, выдавая об этом сообщение, а дождавшись его освобождения использовать его в течение некоторого времени `t2 < t1`, также выдавая информацию о том, что ресурс используется соответствующим командным файлом (процессом). Запустить командный файл в одном виртуальном терминале в фоновом режиме, перенаправив его вывод в другой (`> /dev/tty#`, где `#` — номер терминала куда перенаправляется вывод), в котором также запущен этот файл, но не фоновом, а в привилегированном режиме. Доработать программу так, чтобы имелась возможность взаимодействия трех и более процессов.
2. Реализовать команду `man` с помощью командного файла. Изучите содержимое каталога `/usr/share/man/man1`. В нем находятся архивы текстовых файлов содержащих справку по большинству установленных в системе программ и команд. Каждый архив можно открыть командой `less` сразу же просмотрев содержимое справки. Командный файл должен получать в виде аргумента командной строки название команды и в виде результата выдавать справку об этой команде или сообщение об отсутствии справки, если соответствующего файла нет в каталоге `man1`
3. Используя встроенную переменную `$RANDOM` напишите командный файл, генерирующий случайную последовательность букв латинского алфавита. Учтите, `$RANDOM` что выдает псевдослучайные числа в диапазоне от 0 до 32767.

Лабораторная работа № 14. Средства, применяемые при разработке программного обеспечения в ОС типа UNIX/Linux

1. В домашнем каталоге создайте подкаталог `~/work/os/lab_prog`.
2. Создайте в нём файлы: `calculate.h`, `calculate.c`, `main.c`. Это будет примитивнейший калькулятор, способный складывать, вычитать, умножать и делить, возводить число в степень, брать квадратный корень, вычислять `sin`, `cos`, `tan`. При запуске он будет запрашивать первое число, операцию, второе число. После этого программа выведет результат и остановится.
3. Выполните компиляцию программы посредством

```
gcc:gcc -c calculate.c
gcc -c main.c
gcc calculate.o main.o -o calcul -lm
```

4. При необходимости исправьте синтаксические ошибки.
5. Создайте Makefile
6. С помощью gdb выполните отладку программы calcul
7. С помощью утилиты splint попробуйте проанализировать коды файлов calculate.c и main.c.

Лабораторная работа № 15. Именованные каналы

Изучите приведённые в тексте программы server.c и client.c. Взяв данные примеры за образец, напишите аналогичные программы, внося следующие изменения: 1. Работает не 1 клиент, а несколько (например, два). 2. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). Используйте функцию sleep() для приостановки работы клиента. 3. Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Используйте функцию clock() для определения времени работы сервера. Что будет в случае, если сервер завершит работу, не закрыв канал?

Методические указания и шкала оценок

Порядок выполнения лабораторной работы заключается в следующем:

- Ознакомиться с разделами методических указаний к лабораторной работе.
- Выполнить задания лабораторной работы.
- Подготовить отчёт.

В качестве ответа на лабораторную работу в ТУИС необходимо загрузить:

1. Ссылку на скринкаст с выполнением лабораторной работы (на *youtube*).
2. Ссылку на скринкаст с презентацией лабораторной работы (на *youtube*).
3. Ссылку на репозиторий на *Github* (Для лабораторных работ №2-№15).
4. Следует загрузить отдельными файлами (не общим архивом):
 - Для лабораторных работ №1-№2
 - отчёт в docx;
 - отчёт в pdf;
 - презентацию в pdf.
 - Для лабораторных работ №3-№15
 - отчёт в markdown;
 - отчёт в docx (сделанный из markdown);
 - отчёт в pdf (сделанный из markdown);
 - архив с исходными материалами markdown (текстовые файлы, скриншоты и т. д.);
 - презентацию в pdf (сделанная из markdown);
 - презентацию в markdown.

Критерии оценки

Технические элементы отчёта лабораторной работы

- Скринкаст выполнения лабораторной работы
 - Ссылка на скринкаст выполнения лабораторной работы
 - Элементы скринкаста
 - Изображение рабочего стола с записью процесса выполнения лабораторной работы
 - Изображение выполняющего лабораторную работу с камеры компьютера (обычно в углу экрана)
 - Комментарии голосом, записанные в процессе выполнения лабораторной работы

- Отчёт о выполнении лабораторной работы
 - Форматы отчёта
 - DOCX (Для Лабораторных работ №1 и №2)
 - Markdown (Для Лабораторных работ №3 - №15)
 - PDF (полученный из Markdown) (Для Лабораторных работ №3 - №15)
 - DOCX (полученный из Markdown) (Для Лабораторных работ №3 - №15)
 - Структура отчёта
 - данные о работе (тема, дисциплина), ФИО автора и преподавателя;
 - цели и задачи;
 - объект и предмет исследования;
 - условные обозначения и термины;
 - список иллюстраций и таблиц;
 - теоретические вводные данные;
 - техническое оснащение и выбранные методы проведения работы;
 - полученные результаты;
 - анализ результатов;
 - заключение и выводы.
- Библиографическая информация
- Презентация по лабораторной работе
 - Размер презентации — 5-10 слайдов
 - Структура презентации
 - Представление выступающего (Who is this guy)
 - Прагматика выполнения лабораторной работы (Зачем)
 - Цель выполнения лабораторной работы
 - Задачи выполнения лабораторной работы
 - Результаты выполнения лабораторной работы
- Скринкаст презентации лабораторной работы

Соглашения об именовании

При выполнении работ следует придерживаться следующих правил именовании: - пользователь внутри виртуальной машины должен иметь имя, совпадающее с учётной записью студента, выполняющего лабораторную работу. - Имя хоста вашей виртуальной машины должно совпадать с учётной записью студента, выполняющего лабораторную работу. - Имя виртуальной машины должно совпадать с учётной записью студента, выполняющего лабораторную работу. - В дисплейных классах вы можете посмотреть имя вашей учётной записи, набрав в терминале команду: `id -un`. - При установке на своей технике необходимо использовать имя вашей учётной записи дисплейных классов (Транслитерированные первые буквы имени и отчества, плюс транслитерированная фамилия). Например, если студента зовут Остап Сулейманович Бендер, то его учётная запись имеет вид `osbender`. - Идентификатор пользователя должен быть ясно различим на скриншотах в отчётах.

Технические критерии выполнения лабораторной работы

Критерий	0	1	2
Скринкаст	Отсутствует скринкаст + голос выполнения работы	Скринкаст + голос без фиксирования вебкамерой	Скринкаст + голос с фиксированием вебкамерой
Отчёт (*)	Нет отчёта в markdown	Есть отчёт в markdown (только текст)	Есть отчёт в markdown и архив с материалами (изображения)

Форматы (*)	Нет отчётов в других форматах	Отчёт в docx (из markdown)	Отчёт в docx и в pdf (из markdown)
Структура	Не соответствует структуре отчёта	Частично соответствует структуре отчёта	Полностью соответствует структуре отчёта
Подписи	Скриншоты не подписаны	Не все скриншоты подписаны	Все скриншоты подписаны
Ссылки	На скриншоты нет ссылок	Не на все скриншоты есть ссылки	На все скриншоты есть ссылки
Полнота	Не все этапы работы описаны	Все этапы работы описаны	Все этапы работы описаны + подробное теоретическое введение
Библиография	Нет библиографии	Есть библиография, но ссылки не проставлены	Есть библиография с проставленными ссылками в тексте
Презентация	Презентация работы отсутствует	Есть презентация работы	Есть презентация работы и скринкаст
Оформление (*)	Презентация сделана не в Markdown	Презентация сделана в Markdown	Markdown + научный стиль
Git (**)	Работа не выложена на git	Работа выложена на хостинг git	Github + общепринятые коммиты + семантические версии + changelog

(*) - Критерии для Лабораторных работ №3 - №15

(**) - Критерии для Лабораторных работ №2 - №15

Содержательная часть работы

Критерий	0	1	2
Корректность выполнения	Выполнено не верно	Незначительные ошибки	Выполнено без ошибок
Анализ результатов	Не выполнен	Выполнен не в полном объёме	Выполнен в полном объёме
Запись выполнения работы	Запись отсутствует	Запись неполная или не всё пояснено	Запись полная, пояснения полные

Критерии выполнения соглашения об именовании лабораторной работы

Критерий	0	5	10
Именованье	Отсутствует именованье	Не везде выполнено соглашение об именовании	Везде выполнено соглашение об именовании