

**Федеральное государственное автономное образовательное учреждение
высшего образования
«Санкт-Петербургский политехнический университет Петра Великого»**

УТВЕРЖДАЮ
Директор ИКНК
_____ Д.П. Зегжда
«17» июня 2024 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

«Операционные системы»

Разработчик

Высшая школа компьютерных технологий и информационных систем

Направление (специальность)
подготовки

09.03.02 Информационные системы и технологии

Наименование ООП

09.03.02_02 Информационные системы и технологии

Квалификация (степень)
выпускника

бакалавр

Образовательный стандарт

СУОС

Форма обучения

Очная

СОГЛАСОВАНО

Соответствует СУОС

Руководитель ОП

Утверждена протоколом заседания

_____ А.А. Ефремов

высшей школы "ВШКТиИС"

«26» марта 2024 г.

от «26» марта 2024 г. № 1

РПД разработал:

Доцент, к.т.н. Я.А. Селиверстов

1. Цели и планируемые результаты изучения дисциплины

Цели освоения дисциплины

1. Понимание истории развития и этапов становления индустрии разработки операционных систем.
2. Осознание принципов работы операционных систем: Системы счисления (2-я, 8-я, 16-я).
Понимание базовых компонентов, архитектуры и функциональности ОС.
3. Знание основ программирования на языке ассемблер: Структура языка ассемблера.
Архитектурное окружение, регистры. Передача параметров через стек. Синтаксис команд (Intel, AT&T). Задачи на программирование
4. Изучение управления процессами и потоками: Понимание механизмов создания, планирования и синхронизации процессов.
5. Научиться управлению памятью: Изучение классификации видов памяти, понимание арифметики указателей. Освоение методов выделения и освобождения памяти, а также работы с виртуальной памятью.
6. Знание системного ввода-вывода: Изучение работы с устройствами ввода-вывода и интерфейсами.
7. Изучение файловых систем: Понимание структуры, организации и управления файлами в ОС.
8. Понимание основных концепций многозадачности: Изучение различных моделей многозадачности, включая кооперативную и предвосхищающую. Изучение механизмов управления процессами и планирования задач. Освоение методов синхронизации процессов и предотвращения состояний гонки. Понимание аномалий многопоточных программ. Анализ влияния многозадачности на производительность и эффективность операционных систем.
9. Изучение основных концепций межпроцессного взаимодействия (IPC): Понимание механизмов IPC, такие как файлы, пайпы, сигналы, очереди сообщений, семафоры, разделяемая память и сокеты.
10. Изучение концепций кроссплатформенности в контексте операционных систем:
Понимание принципов виртуализации и её значения для современных вычислительных систем. Изучение различных подходов и технологий виртуализации, включая гипервизоры и контейнеры. Освоение практических навыков настройки и управления виртуальными средами. Управление образами и контейнерами (Docker).
11. Управление пакетами и репозиториями в Linux. Понимание основ сетевой безопасности в Linux: Репозитории и управление репозиториями. Подключение репозиториев. Управление пакетами.
12. Анализ безопасности и защиты данных: Знакомство с методами обеспечения безопасности и контроля доступа в операционных системах.

13. Практические навыки работы с ОС: Развитие умений использовать различные операционные системы, настраивать среду и решать практические задачи.
14. Исследование современных тенденций: Ознакомление с актуальными разработками и трендами в области операционных систем.

Результаты обучения выпускника

Код	Результат обучения (компетенция) выпускника ООП
ОПК-5	Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем
ИД-1 ОПК-5	Применяет методы работы в операционных системах
ОПК-7	Способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем
ИД-1 ОПК-7	Применяет теорию операционных систем

Планируемые результаты изучения дисциплины

знания:

- Знает перечень основных модулей ОС и решаемые ими задачи, базовые алгоритмы функционирования основных компонент ОС

умения:

- Умеет работать в различных ОС

навыки:

- Владеет навыками установки и настройки различных ОС

2. Место дисциплины в структуре ООП

В учебном плане дисциплина «Операционные системы» относится к модулю «Модуль цифровых компетенций».

Изучение дисциплины базируется на результатах освоения следующих дисциплин:

- Введение в профессиональную деятельность
- Дискретная математика
- Алгоритмизация и программирование

3. Распределение трудоёмкости освоения дисциплины по видам учебной работы и формы текущего контроля и промежуточной аттестации

3.1. Виды учебной работы

Виды учебной работы	Трудоемкость по семестрам
	Очная форма
Лекционные занятия	30
Практические занятия	14
Самостоятельная работа	73
Часы на контроль	16
Промежуточная аттестация (экзамен)	11
Общая трудоемкость освоения дисциплины	144, ач
	4, зет

3.2. Формы текущего контроля и промежуточной аттестации

Формы текущего контроля и промежуточной аттестации	Количество по семестрам
	Очная форма
Текущий контроль	
Расчетно-графические работы, шт.	1
Промежуточная аттестация	
Экзамены, шт.	1

4. Содержание и результаты обучения

4.1 Разделы дисциплины и виды учебной работы

№ раздела	Разделы дисциплины, мероприятия текущего контроля	Очная форма		
		Лек, ач	Пр, ач	СР, ач
1.	История развития и этапы становления индустрии разработки операционных систем.	2	0	2

2.	Принципы работы операционных систем: Системы счисления (2-я, 8-я, 16-я). Базовые компоненты, архитектуры и функциональности ОС. Уровни вычислительной системы. Принципы Фон-Неймана. Принципы Гарвардской архитектуры.	4	1	6
3.	Основы программирования на языке ассемблер: Структура языка ассемблера. Архитектурное окружение, регистры. Передача параметров через стек. Синтаксис команд (Intel, AT&T). Задачи на программирование.	4	2	6
4.	Процессы и потоки: Понятие процесса и потока. Механизмы создания, планирования и синхронизации процессов. Системные вызовы. Механизмы прерываний. Представление процесса в различных операционных системах	4	2	6
5.	Организация памяти: Классификация видов памяти, арифметика указателей. Методы выделения и освобождения памяти. Виртуальная память. Сегментация. Страницчная организация памяти. Простой подход к аллокации памяти. Buddy аллокатор. SLAB аллокатор.	4	1	6
6.	Межпроцессное взаимодействие. Назначение и виды IPC. Пайпы, сигналы, очереди сообщений, семафоры, разделяемая память и сокеты.	2	1	6
7.	Планирование и многозадачность. Потоки исполнения и многопоточность. Переключение потоков. Кооперативная и вытесняющая многозадачности. Планирование и критерии планирования. Реалистичное планирование.	2	1	6
8.	Освоение методов синхронизации процессов и предотвращения состояний гонки. Понимание аномалий многопоточных программ. Deadlock-и и средства борьбы с ними. Анализ влияния многозадачности на производительность и эффективность операционных систем. Анализ процессов и средства мониторинга (утилиты) в LINUX.	2	1	6
9.	Файловые системы. Принципы организации файловых систем. Разновидности файловых систем. Интересные особенности: жёсткие и символические ссылки. Файловые системы в GNU/Linux. SSH-клиент. Принципы работы с жёсткими дисками. Устройство жесткого диска. CHS Large. LBA. Запись информации. Интерфейсы. Рэйд. Драйвер символного устройства.	2	2	8

10.	Управление пакетами и репозиториями. Основы сетевой безопасности. Репозитории и управление репозиториями. Подключение репозиториев. Управление пакетами. Основы сетевой безопасности.	2	1	6
11.	Виртуализация. Принципы виртуализации и её значения для современных вычислительных систем. Технологии виртуализации, включая гипервизоры и контейнеры. Linux: Введение в Docker. Установка docker. Обзор docker. Управление образами и контейнерами. Управление сетями в docker. Обзор docker-compose.	2	2	6
Итого по видам учебной работы:		30	14	73
Экзамены, ач				25
Часы на контроль, ач				16
Промежуточная аттестация (экзамен)				11
Общая трудоёмкость освоения: ач / зет				144 / 4

4.2. Содержание разделов и результаты изучения дисциплины

Раздел дисциплины	Содержание
1. История развития и этапы становления индустрии разработки операционных систем.	<p>0. Предпосылки создания</p> <ul style="list-style-type: none">- Механический период;- Механо-электрический период;- Электронный период. <p>1. Начальные этапы (1950-е годы):</p> <ul style="list-style-type: none">- Разработка первых операционных систем началась с создания больших вычислительных машин. Эти системы были адаптированы под конкретные задачи и не имели понятия многопоточности. <p>2. Появление многозадачности (1960-е годы):</p> <ul style="list-style-type: none">- Появление ОС с возможностью выполнения нескольких задач одновременно. Разработка таких ОС как CTSS и MULTICS, которые стали основой для последующих систем. <p>3. Эра мини- и персональных компьютеров (1970-е годы):</p> <ul style="list-style-type: none">- С появлением мини-компьютеров и ПК были разработаны более доступные операционные системы, такие как UNIX. UNIX внедрил концепции многопользовательского доступа и защищенности. <p>4. Стандартизация и распространение (1980-е годы):</p> <ul style="list-style-type: none">- Разработка интерфейсов (например, GUI) и стандартов, таких как POSIX, способствовала популяризации UNIX-подобных систем. Появляется DOS - первая широко распространенная ОС для ПК. <p>5. Революция Windows (1990-е годы):</p> <ul style="list-style-type: none">- С выходом Microsoft Windows операционные системы стали более интуитивными, завоевывая рынок. Windows обеспечила графический интерфейс, который сделал использование ПК популярным. <p>6. Linux и открытые операционные системы (1990-е - 2000-е годы):</p> <ul style="list-style-type: none">- Появление Linux как бесплатной альтернативы проприетарным ОС, что вызвало рост сообщества разработчиков. Это стало основой для многих дистрибутивов Linux. <p>7. Современные тенденции (2010-е - настоящее время):</p> <ul style="list-style-type: none">- Развитие облачных технологий, мобильных ОС (iOS, Android) и виртуализации. Операционные системы становятся все более специализированными, обучаясь адаптироваться под различные устройства и платформы. <p>8. Квантовый период. Основы квантовых вычислительных систем.</p>

<p>2. Принципы работы операционных систем: Системы счисления (2-я, 8-я, 16-я).</p> <p>Базовые компоненты, архитектуры и функциональности ОС. Уровни вычислительной системы.</p> <p>Принципы Фон-Неймана.</p> <p>Принципы Гарвардской архитектуры.</p>	<p>1. Системы счисления (2-я, 8-я, 16-я):</p> <ul style="list-style-type: none"> - Операционные системы работают с различными системами счисления, чтобы представлять данные и инструкции. - Двоичная система (2-я): Использует две цифры (0 и 1) и является основной для компьютерной архитектуры. - Восьмеричная система (8-я): Основана на восьми цифрах (0-7) и используется для более удобного представления двоичных данных. - Шестнадцатеричная система (16-я): Использует шестнадцать символов (0-9 и A-F) и позволяет компактно представлять большие двоичные числа, что особенно полезно в адресации памяти и написании программ. <p>2. Базовые компоненты, архитектуры и функциональности ОС:</p> <ul style="list-style-type: none"> - Основные компоненты ОС включают ядро (управление ресурсами и планирование задач), драйверы устройств (взаимодействие с аппаратурой), системные библиотеки (функционал для приложений) и пользовательский интерфейс. - Архитектуры ОС могут быть монолитными, микроядерными или распределенными, каждая из которых имеет свои преимущества и недостатки в управлении ресурсами и производительности. - Функциональности ОС охватывают управление памятью, файловую систему, обработку прерываний, управление процессами и многозадачность. <p>3. Уровни вычислительной системы:</p> <ul style="list-style-type: none"> - Вычислительная система обычно разбивается на несколько уровней: <ul style="list-style-type: none"> - Аппаратный уровень: включает физические компоненты компьютера, такие как процессор, память и устройства ввода-вывода. - Уровень операционной системы: отвечает за управление аппаратными ресурсами и предоставляет программный интерфейс для приложений. - Уровень приложений: включает все пользовательские программы и приложения, которые используют ресурсы компьютера через ОС. <p>4. Принципы Фон-Неймана:</p> <ul style="list-style-type: none"> - Эти принципы описывают архитектуру цифровых вычислительных систем, состоящую из центрального процессора (ЦП), памяти и устройств ввода-вывода. - Основная идея заключается в хранении программ и данных в одной памяти, что позволяет компьютеру извлекать и выполнять инструкции последовательно. - Принципы Фон-Неймана легли в основу большинства современных компьютеров и их операционных систем, определяя, как осуществляется выполнение программ и управление данными.
---	--

	<p>3. Основы программирования на языке ассемблера: Структура языка ассемблера.</p> <p>Архитектурное окружение, регистры. Передача параметров через стек. Синтаксис команд (Intel, AT&T). Задачи на программирование.</p> <ul style="list-style-type: none"> 1. Структура языка ассемблера: <ul style="list-style-type: none"> - Определение ассемблера как низкоуровневого языка программирования. - Элементы программы: инструкции, операнды, директивы. - Разделение кода на секции: кодовая, данных и стека. 2. Архитектурное окружение, регистры: <ul style="list-style-type: none"> - Введение в архитектуры процессоров (например, x86, ARM). - Описание регистров: общего назначения и специализированных (например, регистры состояния). - Роль регистров в оперативной памяти и их влияние на производительность. 3. Передача параметров через стек: <ul style="list-style-type: none"> - Принципы работы стека: стековая память и стековые операции. - Методы передачи параметров: порядок передачи и доступ к параметрам из функций. - Примеры использования стека для вызова функций и возврата значений. 4. Синтаксис команд (Intel, AT&T): <ul style="list-style-type: none"> - Сравнение синтаксиса Intel и AT&T. - Объяснение основных конструкций и форматов написания инструкций. - Примеры распространенных команд для обеих синтаксических систем. 5. Задачи на программирование: <ul style="list-style-type: none"> - Примеры типичных задач на ассемблере, включая арифметические операции, управление потоками, работа с массивами. - Практические упражнения для закрепления знаний: написание программ, работающих с регистрами и стеком. - Анализ выполненных задач и оптимизация кода.
--	--

<p>4. Процессы и потоки: Понятие процесса и потока. Механизмы создания, планирования и синхронизации процессов. Системные вызовы. Механизмы прерываний. Представление процесса в различных операционных системах</p>	<p>1. Понятие процесса и потока:</p> <ul style="list-style-type: none"> - Определение процесса: независимая программа, выполняющаяся в своей области памяти. - Определение потока: легковесный процесс, обладающий общей памятью с другими потоками того же процесса. - Различия между процессами и потоками: производительность, управление и изоляция. <p>2. Механизмы создания, планирования и синхронизации процессов:</p> <ul style="list-style-type: none"> - Создание процессов: системные вызовы fork, exec и их особенности. - Планирование процессов: алгоритмы планирования (FCFS, SJF, Round Robin) и их влияние на производительность. - Синхронизация процессов: механизмы (мьютексы, семафоры, барьеры) для предотвращения гонок данных. <p>3. Системные вызовы:</p> <ul style="list-style-type: none"> - Определение системных вызовов как интерфейса между приложениями и операционной системой. - Примеры системных вызовов для управления процессами: wait, kill, getpid. - Особенности обработки системных вызовов в различных операционных системах. <p>4. Механизмы прерываний:</p> <ul style="list-style-type: none"> - Определение прерываний и их роль в управлении процессами. - Программные и аппаратные прерывания: различия и примеры. - Обработка прерываний и их влияние на планирование процессов. <p>5. Представление процесса в различных операционных системах:</p> <ul style="list-style-type: none"> - Структуры данных для описания процессов (например, таблицы процессов, структуры контекста). - Различия в представлении процессов в Windows, Linux и других ОС. - Уровни абстракции процессов: пользовательские и ядерные уровни, виртуализация процессов.
---	--

	<p>1. Классификация видов памяти:</p> <ul style="list-style-type: none"> - Основные виды памяти: оперативная (RAM), постоянная (ROM), кэш-память. - Мгновенная, вторичная и виртуальная память: функции и примеры использования. - Разделение памяти на физическую и логическую. <p>2. Арифметика указателей:</p> <ul style="list-style-type: none"> - Понятийный аппарат указателей: определение и роль в управлении памятью. - Операции с указателями: арифметические операции, инкремент, декремент. - Применение указателей в динамическом выделении памяти. <p>3. Методы выделения и освобождения памяти:</p> <ul style="list-style-type: none"> - Статическое, динамическое и автоматическое выделение памяти. - Алгоритмы выделения памяти: первую подходящую, оставшуюся (best-fit) и худшую (worst-fit) стратегии. - Освобождение памяти: системные вызовы и механизмы управления. <p>4. Виртуальная память:</p> <ul style="list-style-type: none"> - Определение виртуальной памяти и её преимущества для приложений. - Механизмы, такие как страничная и сегментная виртуализация. - Обработка страниц и замещение: алгоритмы (LRU, FIFO и т.д.). <p>5. Сегментация:</p> <ul style="list-style-type: none"> - Определение сегментации как метода управления памятью. - Структура сегментов: код, данные, стек. - Преимущества и недостатки сегментации по сравнению со страничной организацией. <p>6. Страницчная организация памяти:</p> <ul style="list-style-type: none"> - Определение страниц и их роль в системах управления памятью. - Описание таблиц страниц и механизма трансляции адресов. - Применение страницочной организации для оптимизации работы оперативной памяти. <p>7. Простой подход к аллокации памяти:</p> <ul style="list-style-type: none"> - Описание базовых подходов к выделению и освобождению памяти. - Проблемы фрагментации: внешняя и внутренняя. - Методики устранения фрагментации. <p>8. Buddy аллокатор:</p> <ul style="list-style-type: none"> - Определение и принцип работы Buddy аллокатора. - Структура разделения блоков памяти на «друзья» (buddies). - Преимущества Buddy аллокатора: эффективность и простота. <p>9. SLAB аллокатор:</p> <ul style="list-style-type: none"> - Определение SLAB аллокатора и его использование в системах. - Организация кэшей для объектов одного типа. - Преимущества SLAB аллокатора: уменьшение фрагментации и
--	---

<p>6. Межпроцессное взаимодействие. Назначение и виды IPC. Пайпы, сигналы, очереди сообщений, семафоры, разделяемая память и сокеты.</p>	<p>1. Назначение IPC:</p> <ul style="list-style-type: none"> - Определение межпроцессного взаимодействия (IPC) и его значение в многозадачных системах. - Проблемы, которые решает IPC: обмен данными, синхронизация процессов и совместный доступ к ресурсам. <p>2. Виды IPC:</p> <ul style="list-style-type: none"> - Обзор основных методов IPC и их использование в приложениях. - Классификация методов по способу взаимодействия: прямое и косвенное. <p>3. Пайпы:</p> <ul style="list-style-type: none"> - Определение пайпов и их назначение в IPC. - Однонаправленные и двунаправленные пайпы: особенности использования. - Примеры применения пайпов для передачи данных между процессами. <p>4. Сигналы:</p> <ul style="list-style-type: none"> - Определение сигналов как средства межпроцессного взаимодействия. - Примеры распространённых сигналов и их обработка. - Синхронизация процессов с помощью сигналов: управление выполнением. <p>5. Очереди сообщений:</p> <ul style="list-style-type: none"> - Определение и структура очередей сообщений. - Методы отправки и получения сообщений. - Преимущества очередей сообщений: асинхронность и устойчивость к сбоям. <p>6. Semaphore:</p> <ul style="list-style-type: none"> - Объяснение семафоров как средств синхронизации. - Типы семафоров: бинарные и счётные. - Примеры применения семафоров для управления доступом к ресурсам. <p>7. Разделяемая память:</p> <ul style="list-style-type: none"> - Определение разделяемой памяти и её преимущества для IPC. - Методы синхронизации доступа к разделяемой памяти. - Преимущества и недостатки использования разделяемой памяти. <p>8. Сокеты:</p> <ul style="list-style-type: none"> - Определение сокетов и их использование для межпроцессного взаимодействия. - Серверные и клиентские сокеты: принципы работы и структуры. - Применение сокетов для сетевого взаимодействия между процессами. <p>9. Сравнение методов IPC:</p> <ul style="list-style-type: none"> - Сравнение вышеуказанных методов IPC по критериям производительности, сложности реализации и удобству использования. - Выбор подходящего метода в зависимости от требований
---	---

7. Планирование и многозадачность. Потоки исполнения и многопоточность. Переключение потоков. Кооперативная и вытесняющая многозадачности. Планирование и критерии планирования. Реалистичное планирование.

1. Планирование:
 - Определение планирования как процесса распределения процессорного времени между задачами.
 - Значение эффективного планирования для производительности системы.
2. Многозадачность:
 - Определение многозадачности и её роль в современных операционных системах.
 - Различие между задачами и потоками исполнения.
3. Потоки исполнения и многопоточность:
 - Объяснение концепции потоков исполнения как единиц выполнения в рамках процесса.
 - Определение многопоточности и её преимущества: более эффективное использование ресурсов и параллелизм.
4. Переключение потоков:
 - Процесс переключения между потоками исполнения.
 - Издержки на переключение: время и ресурсы, затрачиваемые на сохранение состояния потоков.
 - Алгоритмы и механизмы для выполнения переключения потоков.
5. Кооперативная многозадачность:
 - Определение кооперативной многозадачности.
 - Принципы работы: процессы самостоятельно передают управление.
 - Преимущества и недостатки: простота реализации против риска зависания.
6. Вытесняющая многозадачность:
 - Определение вытесняющей многозадачности.
 - Принципы работы: ОС управляет временем выполнения процессов и может прервать их.
 - Преимущества и недостатки: более высокий уровень контроля против дополнительных затрат на управление.
7. Планирование и критерии планирования:
 - Основные алгоритмы планирования: FCFS, SJF, RR и их принципы работы.
 - Критерии планирования: время ожидания, время выполнения, отклонение от ожиданий и другие метрики производительности.
8. Реалистичное планирование:
 - Проблемы, возникающие при идеальном планировании в реальных условиях.
 - Учет факторов: приоритеты, долгосрочные и краткосрочные задачи.
 - Стратегии для повышения эффективности планирования в реальных системах.
9. Заключение:
 - Роль планирования и многозадачности в повышении производительности системы.

8. Освоение методов синхронизации процессов и предотвращения состояний гонки. Понимание аномалий многопоточных программ. Deadlock-и и средства борьбы с ними. Анализ влияния многозадачности на производительность и эффективность операционных систем. Анализ процессов и средства мониторинга (утилиты) в LINUX.

1. Методы синхронизации процессов:
 - Определение синхронизации как механизма управления доступом к общим ресурсам.
 - Основные методы: мьютексы, семафоры, условные переменные.
2. Предотвращение состояний гонки:
 - Понимание состояний гонки как ситуации, когда результаты выполнения процессов зависят от неопределенного порядка их выполнения.
 - Стратегии предотвращения: критические секции и блокировки.
3. Аномалии многопоточных программ:
 - Объяснение аномалий, таких как неправильная последовательность выполнения и другие ошибки, возникающие при недостаточной синхронизации.
 - Примеры типичных ошибок и способы их обнаружения.
4. Deadlock-и:
 - Определение дедлоков как состояний, при которых процессы блокируются в ожидании друг друга.
 - Условия, способствующие возникновению дедлоков: взаимное исключение, удержание и ожидание, отсутствие потерь ресурсов, циклическое ожидание.
5. Средства борьбы с дедлоками:
 - Стратегии предотвращения, обнаружения и разрешения дедлоков.
 - Примеры алгоритмов, таких как алгоритм Банкарда и методы временного отката.
6. Анализ влияния многозадачности на производительность:
 - Как многозадачность влияет на скорость выполнения задач и ресурсоемкость.
 - Понимание компромиссов между параллелизмом и сложностью управления ресурсами.
7. Эффективность операционных систем:
 - Как многозадачность, синхронизация и предотвращение дедлоков способствуют общей эффективности ОС.
 - Оценка производительности и типичные метрики, используемые для анализа ос.
8. Анализ процессов и средства мониторинга в LINUX:
 - Основные утилиты для мониторинга процессов: top, htop, ps, vmstat.
 - Инструменты для анализа производительности: perf, strace.
 - Способы использования этих инструментов для диагностики проблем и оптимизации производительности.
9. Заключение:
 - Значение методов синхронизации и предотвращения состояний гонки в разработке надежных многопоточных приложений.
 - Перспективы улучшения синхронизации и мониторинга в современных ОС.

9. Файловые системы.
Принципы организации файловых систем.
Разновидности файловых систем. Интересные особенности: жёсткие и символические ссылки.
Файловые системы в GNU/Linux. SSH-клиент. Принципы работы с жёсткими дисками.
Устройство жесткого диска. CHS Large. LBA. Запись информации. Интерфейсы.
Рэйд. Драйвер символьного устройства.

1. Принципы организации файловых систем:
 - Основные компоненты файловой системы: структуры данных, метаданные и методы управления.
 - Как файловые системы организуют данные на физическом носителе и обеспечивают доступ к ним.
2. Разновидности файловых систем:
 - Описание различных типов файловых систем: FAT32, NTFS, ext2, ext3, ext4, HFS+, etc.
 - Выбор файловой системы в зависимости от требований к производительности, надёжности и совместимости.
3. Интересные особенности:
 - Объяснение жёстких и символьических ссылок:
 - Жёсткие ссылки: ссылки на один и тот же inode, позволяющие доступ к данным по различным именам.
 - Символьические ссылки: файл-указатель, который может ссылаться на другой файл или директорию.
4. Файловые системы в GNU/Linux:
 - Особенности и популярность файловых систем в GNU/Linux, таких как ext4 и Btrfs.
 - Поддержка прав доступа и управление пользователями в файловых системах Linux.
5. SSH-клиент:
 - Описание SSH-клиента как инструмента для безопасного удалённого доступа к серверам.
 - Принципы работы: аутентификация, шифрование данных и управление сессиями.
6. Принципы работы с жёсткими дисками:
 - Как операционные системы взаимодействуют с жёсткими дисками для чтения и записи данных.
 - Роли драйверов и файловых систем в этом процессе.
7. Устройство жесткого диска:
 - Структура жесткого диска: пластины, считающие/записывающие головки, моторы и контроллеры.
 - Описание процессов, связанных с хранением и доступом к данным.
8. CHS Large и LBA:
 - Объяснение методов адресации секторов:
 - CHS (Cylinder-Head-Sector) — традиционный метод, использующий цилиндры, головки и сектора.
 - LBA (Logical Block Addressing) — более современный метод, который упрощает адресацию.
9. Запись информации:
 - Процессы записи данных на жесткий диск, включая алгоритмы и оптимизацию.
 - Методы проверки целостности записанных данных.

10. Управление пакетами и репозиториями. Основы сетевой безопасности. Репозитории и управление репозиториями.

Подключение репозиториев.

Управление пакетами. Основы сетевой безопасности.

1. Основы сетевой безопасности:
 - Введение в концепции сетевой безопасности.
 - Объяснение угроз, таких как вирусы, мошенничество и атаки.
 - Принципы защиты данных и конфиденциальности в сети.
 2. Репозитории и управление репозиториями:
 - Определение понятия репозитория: хранилище пакетов программного обеспечения.
 - Разновидности репозиториев: официальные, сторонние и локальные.
 - Как управлять репозиториями, чтобы обеспечить доступ к необходимым пакетам.
 3. Подключение репозиториев:
 - Процесс добавления новых репозиториев в систему.
 - Настройки конфигурации и документация по каждому репозиторию.
 - Примеры добавления репозиториев в популярных дистрибутивах (например, Debian/Ubuntu, Fedora).
 4. Управление пакетами:
 - Введение в системы управления пакетами: APT, RPM, YUM, DNF.
 - Основные команды для установки, обновления и удаления пакетов.
 - Как управлять зависимостями пакетов для правильной работы ПО.
 5. Основы сетевой безопасности (повтор):
 - Разработка безопасной архитектуры сетей.
 - Использование брандмауэров, VPN и других инструментов для защиты данных.
 - Лучшие практики по обеспечению сетевой безопасности для пользователей и администраторов.
- Заключение:**
- Важность управления пакетами и репозиториями для обеспечения безопасности и стабильности системы.
 - Роль сетевой безопасности в защите данных и систем от угроз извне.

11. Виртуализация. Принципы виртуализации и её значения для современных вычислительных систем. Технологии виртуализации, включая гипервизоры и контейнеры. Linux: Введение в Docker. Установка docker. Обзор docker. Управление образами и контейнерами. Управление сетями в docker. Обзор docker-compose.

1. Принципы виртуализации и её значения для современных вычислительных систем:
 - Определение виртуализации: создание виртуальных версий физических ресурсов.
 - Значение виртуализации для оптимизации использования оборудования, повышения гибкости и масштабируемости систем.
 - Преимущества виртуализации, такие как снижение затрат на инфраструктуру и улучшение управления ресурсами.
 2. Технологии виртуализации, включая гипервизоры и контейнеры:
 - Различие между гипервизорами (виртуальные машины) и контейнерами.
 - Основные типы гипервизоров: тип 1 (Bare-metal) и тип 2 (Hosted).
 - Введение в контейнеризацию: преимущества легковесных, изолированных сред выполнения.
 3. Linux: Введение в Docker:
 - Обзор Docker как платформы для контейнеризации приложений.
 - Преимущества использования Docker для разработки и развертывания приложений.
 4. Установка Docker:
 - Пошаговая инструкция по установке Docker на различных дистрибутивах Linux.
 - Проверка успешности установки через командную строку.
 5. Обзор Docker:
 - Основные компоненты Docker: демоны, клиенты и реестр.
 - Объяснение архитектуры Docker и как она взаимодействует с системой.
 6. Управление образами и контейнерами:
 - Принципы работы с образами: создание, загрузка и управление.
 - Команды Docker для управления контейнерами: запуск, остановка, удаление и мониторинг.
 7. Управление сетями в Docker:
 - Обзор сетевых моделей Docker и их настройки.
 - Как создать и настроить пользовательские сети для контейнеров.
 8. Обзор docker-compose:
 - Введение в docker-compose как инструмент для управления многоконтейнерными Docker-приложениями.
 - Примеры использования docker-compose для описания и развертывания приложений с несколькими контейнерами.
- Заключение:
- Важность виртуализации и контейнеризации для современных вычислительных систем и их влияние на разработку, эксплуатацию и управление инфраструктурой.

5. Образовательные технологии

1. Интерактивные лекции: - Использование мультимедийных презентаций для визуализации сложных концепций. - Встраивание опросов для активного вовлечения студентов.
2. Лабораторные работы: - Практическое освоение операционных систем через hands-on задания. - Установка и настройка различных операционных систем (Linux, Windows) на виртуальных машинах.
3. Модульное обучение: - Разделение курса на модули, каждый из которых охватывает определенные темы, такие как архитектура ОС, управление памятью, файловые системы и др. - Каждый модуль включает теоретическую часть, практическое задание и проверку знаний.
4. Форумы и дискуссионные группы: - Создание онлайн-форумов/групп для обсуждения тем курса, решения проблем и обмена знаниями.
5. Виртуальные симуляторы: - Использование симуляторов для демонстрации работы операционных систем и процессов, таких как планирование задач и управление памятью. - Моделирование системных вызовов и их влияние на работу ОС.
6. Онлайн-курсы по направлению ОС: - Применение системы поощрений за прохождение открытых онлайн курсов по направлению ОС.
7. Оценка и обратная связь: - Регулярное тестирование и оценка знаний через онлайн-тесты и практические задания.- Обратная связь по результатам работ и экзаменов для понимания сильных и слабых сторон студентов.

6. Лабораторный практикум

Не предусмотрено

7. Практические занятия

№ раздела	Наименование практических занятий (семинаров)	Трудоемкость, ач
		Очная форма
1.	Общая версия Linux. Уровень 1. Установка операционной системы и знакомство с ней. Выбор дистрибутива. Сравнение Debian-подобных и RedHat-подобных дистрибутивов. Обзор главного окна инсталлятора Разметка диска. Завершение установки ОС Утилита sudo, административные действия в ОС. Настройка сети, используя утилиты ip и nmcli. Изменение или настройка сети с использованием утилиты ip. Изменение или настройка сети с использованием утилит ifconfig и route из пакета net-tools. Изменение или настройка сети с использованием утилиты nmcli. Обновление системы. Установка дополнений VM в случае необходимости. Установка OpenSSH Server и файлового менеджера Midnight Commander. Знакомство с интерфейсом (терминал). Навигация по файловой системе и основные операции с файлами и каталогами. Знакомство с текстовыми редакторами. Текстовый редактор vi/Vim. Текстовый редактор nano. Практическое задание: создание/перемещение/удаление каталогов/пользователей.	1
2.	Пользователи. Управление пользователями и группами. Типы пользователей в Linux. UID. GID. Суперпользователь (root). Системные пользователи (пользователи-демоны, технологические пользователи). Обычные пользователи. Управление пользователями. Создание пользователей и групп. Добавление пользователей. Примеры использования команды и параметров. Добавление группы. Изменение параметров пользователя. Изменение владельца и группы владельца файлов и каталогов. Утилиты sudo, su	1
3.	Linux. Загрузка операционной системы. Процесс. Управление процессами. Атрибуты процессов (PID, PPID, UID, SMD). Управление процессами (systemctl). Стандартные потоки (STDIN, STDOUT, STDERR). Мониторинг процессов и состояния компьютера (ps, top, df). Потоки ввода/вывода. Конвейер (pipeline). Сигналы.	1

4.	Устройство файловой системы Linux. Понятия файла и каталога. Типы файлов в Linux Inode и каталог. Жёсткие и символические ссылки. Права доступа к файлам и каталогам. Буквенная запись. Численная запись. Специальные биты.	1
5.	Основы ассемблера. Основы синтаксиса Ассемблера. Сегменты памяти. Регистры (Registers). Системные вызовы. Режимы адресации. Переменные. Константы. Арифметические инструкции. Логические инструкции .Условия. Петли. Числа. Строки. Массивы. Процедуры. Рекурсия. Макросы. Управление файлами. Управление памятью.	2
6.	Введение в скрипты bash. Планировщики задач crontab и at.	1
7.	Анализ процессов в Linux (ps и pstree).	1
8.	Межпроцессное взаимодействие (IPC): именованные и неименованные каналы, сигналы.	1
9.	Организация памяти. Страницчная организация памяти - преобразование логических адресов в физические. Реализация динамического аллокатора памяти. SLAB аллокатор - реализация кеширующего аллокатора.	1
10.	Планирование и многозадачность. Реализовать планировщик, использующий алгоритм Round Robin.	1
11.	Управление пакетами и репозиториями. Основы сетевой безопасности. Управление пакетами через утилиты apt, dpkg, snap. Основы сетевой безопасности и журналирование событий операционной системы- утилиты netfilter и iptables.	1
12.	Установка Docker. Обзор и структура Docker. Управление образами и контейнерами. Управление сетями в Docker. Docker Compose.	1
13.	Написать драйвер символьного устройства.	1
Итого часов		14

8. Организация и учебно-методическое обеспечение самостоятельной работы

Организация самостоятельной работы по дисциплине "Операционные системы" играет важную роль в успешном освоении материала. Вот некоторые рекомендации по организации и учебно-методическому обеспечению самостоятельной работы:

1. Планирование времени: Создайте расписание самостоятельной работы, учитывая свои занятости и обязанности. Разбейте материал на небольшие блоки и распределите время на изучение каждого блока.

2. Выбор источников информации: Определите основные учебники, лекции, онлайн-курсы и другие ресурсы, которые помогут вам изучить материал. Выберите наиболее подходящие для вас источники.

3. Постановка целей: Определите конкретные цели и задачи для каждой сессии самостоятельной работы. Это поможет вам ориентироваться и следить за прогрессом.

4. Активное изучение материала: Помимо чтения учебников, активно работайте с материалом. Решайте задачи, выполняйте практические работы, делайте конспекты и т.д.

5. Самопроверка: После изучения каждой темы проверяйте свои знания с помощью тестов, задач или самопроверочных вопросов. Это поможет закрепить материал.

6. Групповая работа: Обсуждайте материал с коллегами или создавайте группы для совместного изучения. Обсуждение темы с другими студентами может помочь лучше усвоить материал.

7. Лабораторные работы и проекты: Если это предусмотрено программой, уделите достаточно времени на выполнение лабораторных работ и проектов. Практическое применение знаний поможет лучше понять материал.

8. Обратная связь: Постоянно анализируйте свой прогресс и результаты самостоятельной работы. При необходимости корректируйте свой подход к изучению материала.

Примерное распределение времени самостоятельной работы студентов

Вид самостоятельной работы	Примерная трудоемкость, ач
	Очная форма
Текущая СР	
работа с лекционным материалом, с учебной литературой	22
опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	2
самостоятельное изучение разделов дисциплины	10
выполнение домашних заданий, домашних контрольных работ	16
подготовка к лабораторным работам, к практическим и семинарским занятиям	2
подготовка к контрольным работам, коллоквиумам	2
Итого текущей СР:	54
Творческая проблемно-ориентированная СР	
выполнение расчётно-графических работ	0
выполнение курсового проекта или курсовой работы	0
поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме	4
работа над междисциплинарным проектом	0
исследовательская работа, участие в конференциях, семинарах, олимпиадах	2
анализ данных по заданной теме, выполнение расчётов, составление схем и моделей на основе собранных данных	4
Итого творческой СР:	10
Общая трудоемкость СР:	73

9. Учебно-методическое обеспечение дисциплины

9.1. Адрес сайта курса

<https://drive.google.com/drive/folders/1Y45dJ1GP1vuXvmC8mOQWo7guClhe0P3S>

9.2. Рекомендуемая литература

Основная литература

№	Автор, название, место издания, издательство, год (годы) издания	Год изд.	Источник
1	Таненбаум Э., Гороховский Ю., Шинтяков Д. Архитектура компьютера: Москва [и др.]: Питер, 2010.	2010	ИБК СПбПУ
2	Коликова Т.В. Операционные системы, 2012. URL: http://elib.spbstu.ru/dl/2688.pdf	2012	ЭБ СПбПУ

Дополнительная литература

№	Автор, название, место издания, издательство, год (годы) издания	Год изд.	Источник
1	Душутина Е.В. Системное программное обеспечение. Межпроцессные взаимодействия в операционных системах: Санкт-Петербург: Изд-во Политехн. ун-та, 2016. URL: http://elib.spbstu.ru/dl/2/s17-72.pdf	2016	ЭБ СПбПУ
2	Дубовцев А.В. Tango. Операционная система из будущего: СПб.: БХВ-Петербург, 2011.	2011	ИБК СПбПУ
3	Душутина Е.В. Системное программное обеспечение. Практические вопросы разработки системных приложений: Санкт-Петербург: Изд-во Политехн. ун-та, 2016. URL: http://elib.spbstu.ru/dl/2/s17-71.pdf	2016	ЭБ СПбПУ

Ресурсы Интернета

1. Архитектура операционной системы (на примере Linux), осень 2017 – YouTube: <https://www.youtube.com/playlist?list=PLlb7e2G7aSpT4SMpYl6uVYi13k0k9CBiq>
2. Программа курса · Операционные системы · Stepik: <https://stepik.org/course/1780/syllabus>
3. Программа курса · Введение в Linux · Stepik: <https://stepik.org/course/73/syllabus>
4. Программа курса · Введение в архитектуру ЭВМ. Элементы операционных систем. · Stepik: <https://stepik.org/course/253/syllabus>
5. Документация Intel: <https://www.intel.com/content/www/us/en/support/ru-banner-inside.html>
6. Документация Linux: <https://git.kernel.org>
7. Linux Utilities: <https://github.com/Tinram/Linux-Utilities?ysclid=l7mw0ab8r8597744208>
8. Эндрю Таненбаума «Современные операционные системы», 4-е изд. — СПб.: Питер, 2015. — 1120 с.: https://vk.com/wall-43363264_368910?ysclid=lyurzcv465369487704

9. Лучшие книги о Linux (<https://losst.pro/luchshie-knigi-o-linux?ysclid=l7mvkc3d0894257680#1>)

9.3. Технические средства обеспечения дисциплины

1. Компьютерные комплексы:

- Персональные компьютеры с установленными различными операционными системами (Linux, Windows, macOS) для практических занятий.
- Серверы для освоения серверных операционных систем и сетевых технологий.

2. Виртуализация:

- Использование программного обеспечения для виртуализации, такого как VMware, VirtualBox или KVM, для создания и тестирования нескольких ОС на одном хосте.
- Обеспечение изолированной среды для экспериментов и исследований.

3. Лабораторное оборудование:

- Сетевое оборудование (маршрутизаторы, коммутаторы) для изучения сетевых аспектов операционных систем.
- Устройства для управления умными домами и IoT для практики с встраиваемыми ОС.

4. Системы управления версиями:

- Git и другие системы для управления исходным кодом, позволяющие студентам работать над проектами в команде и отслеживать изменения в коде.

5. Программные инструменты:

- IDE и текстовые редакторы (например, Visual Studio Code, Eclipse) для написания и отладки программ на различных языках.
- Командные интерпретаторы и оболочки (bash, PowerShell) для практики работы с командной строкой.

6. Эмуляторы и симуляторы:

- Эмуляторы операционных систем для обучения работе с различными платформами без необходимости физического оборудования.
- Симуляторы сетевых протоколов и процессов для визуализации работы ОС.

7. Лабораторные стенды:

- Подготовленные стенды с различными конфигурациями ОС для проведения сравнительных испытаний и анализа производительности.

8. Облачные платформы:

- Доступ к облачным сервисам (например, AWS, Google Cloud) для изучения облачных ОС и развертывания приложений в облачной среде.

Заключение:

- Использование разнообразных технических средств обеспечивает комплексный подход к обучению дисциплине "Операционные системы", давая студентам практические навыки и углубленные знания.

10. Материально-техническое обеспечение дисциплины

1. Компьютерная техника:

- Современные персональные компьютеры или ноутбуки, оснащенные достаточной производительностью для работы с различными ОС.

2. Сетевое оборудование:

- Коммутаторы и маршрутизаторы для изучения сетевых технологий и взаимодействия операционных систем в сети.
- Модемы и точки доступа для практики в настройке сетевого оборудования.

3. Программное обеспечение:

- Лицензионные версии различных операционных систем (Linux, Windows Server, macOS) для установки на учебные устройства.
- Инструменты виртуализации (например, VMware, VirtualBox), позволяющие создавать несколько виртуальных машин на одном оборудовании.

4. Лабораторное оборудование:

- Специальные стенды для практического обучения, включающие в себя оборудование для настройки и обслуживания ОС.
- Учебные наборы для работы с микроконтроллерами и встраиваемыми системами.

5. Мебель и пространство:

- Удобные рабочие места, включая столы и стулья, для комфортного обучения студентов.
- Аудитории или лаборатории, оборудованные проектором и экраном для лекций и демонстраций.

6. Облачные ресурсы:

- Доступ к облачным платформам (например, Microsoft Azure, Google Cloud), что позволяет студентам работать с операционными системами в облачной среде.

7. Учебные материалы:

- Печатные и электронные учебники, методические пособия и лабораторные работы, а также доступ к онлайн-ресурсам и видеолекциям.

8. Программные средства для разработки:

- Компьютерные программы и интегрированные среды разработки (IDE) для практики программирования на различных языках.

Заключение:

- Комплексное материально-техническое обеспечение дисциплины "Операционные системы" способствует качественному обучению, позволяет студентам приобретать практические навыки и готовит их к реальным задачам в области информационных технологий.

11. Критерии оценивания и оценочные средства

11.1. Критерии оценивания

Для дисциплины «Операционные системы» формой аттестации является экзамен. Дисциплина реализуется с применением системы индивидуальных достижений.

Текущий контроль успеваемости

Максимальное значение персонального суммарного результата обучения (ПСРО) по приведенной шкале - 100 баллов

Максимальное количество баллов приведенной шкалы по результатам прохождения двух точек контроля - 80 баллов.

Подробное описание правил проведения текущего контроля с указанием баллов по каждому контрольному мероприятию и критериев выставления оценки размещается в СДО в навигационном курсе дисциплины.

Промежуточная аттестация по дисциплине

Максимальное количество баллов по результатам проведения аттестационного испытания в период промежуточной аттестации – 20 баллов приведенной шкалы.

Промежуточная аттестация по дисциплине проводится в соответствии с расписанием.

Критерии оценивания качества освоения дисциплины "Операционные системы":

1. Теоретические знания:

- Оценка знаний основных концепций, принципов и архитектуры операционных систем.
- Умение объяснять функции и предназначение различных компонентов ОС.

2. Практические навыки:

- Умение устанавливать, настраивать и администрировать операционные системы.
- Проведение лабораторных работ и выполнение практических задач.

3. Курсовая работа/проект:

- Качество выполнения курсового проекта или исследовательской работы по теме, связанной с операционными системами.

- Оригинальность и практическая значимость выполненный работы.

4. Участие в обсуждениях и семинарах:

- Активность и качество участия в групповых обсуждениях, семинарах и практических занятиях.

5. Экзаменационные результаты:

- Оценка по экзамену (письменному, устному или компьютерному) как итоговому тесту на усвоение материала.

Алгоритм выведения итоговой оценки:

1. Сбор оценок:

- Определите оценку за каждую категорию (теория, практика, курсовая работа, участие, экзамены) на основе выставленных баллов.

2. Калькуляция общей суммы баллов:

- Сложите все полученные баллы. Пример:

- Теоретические знания: 75

- Практические навыки: 125

- Доклад: 25

- Посещение лекций: 70

- Экзамен: 90

- Общая сумма: $75 + 65 + 85 + 70 + 90 = 385$

3. Определение итогового балла:

- Разделите общую сумму баллов на количество оценок.

- Пример: $385 / 5 = 77$.

4. Классификация итоговой оценки:

- Используйте следующие границы для классификации итоговой оценки:

- Неудовлетворительно: 0 - 49
- Удовлетворительно: 50 - 69
- Хорошо: 70 - 89
- Отлично: 90 - 100
- Итоговая оценка по результатам примера будет "Хорошо" (77).

Заключение:

- Применение данных критериев и алгоритма позволяет объективно оценить качество усвоения дисциплины "Операционные системы" и обеспечивает прозрачность процесса оценивания.

Результаты промежуточной аттестации, определяются на основе баллов, набранных в рамках применения, СИД

Баллы по приведенной шкале в рамках применения СИД (ПСРО+ ПА)	Оценка по результатам промежуточной аттестации
	Экзамен/диф.зачет/зачет
0 - 60 баллов	Неудовлетворительно/не зачтено
61 - 75 баллов	Удовлетворительно/зачтено
76 - 89 баллов	Хорошо/зачтено
90 и более	Отлично/зачтено

11.2. Оценочные средства

Оценочные средства по дисциплине представлены в фонде оценочных средств, который является неотъемлемой частью основной образовательной программы и размещается в электронной информационно-образовательной среде СПбПУ на портале etk.spbstu.ru

12. Методические рекомендации по организации изучения дисциплины

Изучение дисциплины "Операционные системы" требует систематического подхода и хорошей организации учебного процесса. Вот несколько методических рекомендаций, которые могут помочь вам успешно освоить этот предмет:

1. Понимание основных концепций: Начните с изучения основных концепций операционных систем, таких как процессы, потоки управления, память, файловая система и др. Это поможет вам лучше понять функционирование операционных систем.

2. Использование учебников и онлайн-ресурсов: Используйте учебники, лекции и онлайн-ресурсы для более глубокого понимания материала. Рекомендуемые учебники, такие как "Operating System Concepts" от Abraham Silberschatz и др., могут быть полезны.

3. Практические навыки: Постарайтесь проводить практические занятия, работая с различными операционными системами. Это поможет вам лучше понять принципы работы ОС на практике.

4. Решение задач и выполнение лабораторных работ: Регулярно решайте задачи по теме и выполняйте лабораторные работы. Это поможет вам закрепить теоретические знания и улучшить понимание материала.

5. Общение с преподавателем и коллегами: Если у вас возникают вопросы или затруднения, не стесняйтесь обращаться к преподавателю или обсуждать материал с коллегами. Общение может помочь вам лучше усвоить материал.

6. Создание конспектов и резюме: Ведите конспекты лекций и создавайте краткие резюме по каждой теме. Это поможет вам систематизировать информацию и легче повторять материал перед экзаменом.

7. Тестирование знаний: Регулярно проверяйте свои знания с помощью тестов и самопроверок. Это поможет вам выявить слабые места и сконцентрироваться на них.

Следуя этим методическим рекомендациям и уделяя достаточно времени изучению дисциплины "Операционные системы", вы сможете успешно освоить этот предмет. Удачи в учебе!

13. Адаптация рабочей программы для лиц с ОВЗ

Адаптированная программа разрабатывается при наличии заявления со стороны обучающегося (родителей, законных представителей) и медицинских показаний (рекомендациями психолого-медико-педагогической комиссии). Для инвалидов адаптированная образовательная программа разрабатывается в соответствии с индивидуальной программой реабилитации.