

**ГОУ ВПО Российско-Армянский (Славянский)
университет**

Утверждено
Директор Института математики и информатики
Арамян Р.Г.



«21» мая 2025, протокол №9.1

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ

Наименование дисциплины: Операционные системы

Автор: Петросян Давид Арменович

Направление подготовки: 01.03.02 Прикладная математика и информатика
Наименование образовательной программы 01.03.02 Прикладная
математика и информатика

1. АННОТАЦИЯ

1.1. Краткое описание содержания данной дисциплины;

Курс «Операционные системы» предоставляет студентам глубокое понимание внутренней структуры и принципов работы современных операционных систем. Он охватывает ключевые компоненты и механизмы, лежащие в основе их функционирования, а также методы их реализации. Основное внимание уделяется фундаментальным концепциям и практическим аспектам, необходимым для анализа, проектирования и разработки системного программного обеспечения.

Дисциплина начинается с изучения процессов начальной загрузки системы и роли BIOS/UEFI. Затем студенты погружаются в детали организации файловых систем, включая концепцию виртуальных файловых систем и конкретные реализации, такие как Ext4. Значительное внимание уделяется управлению памятью, рассматриваются такие механизмы, как страничная организация, сегментация и свопинг. Отдельный блок посвящен планированию процессов, где разбираются различные алгоритмы планирования и механизмы переключения контекста. Курс также затрагивает современные технологии, такие как виртуализация и эмуляция, а также принципы работы контейнерных технологий, давая представление о том, как эти концепции реализованы и используются в современных системах.

1.2. Трудоемкость в академических кредитах и часах, формы итогового контроля (экзамен/зачет);

Курс «Операционные системы» осваивается в течение одного семестра на 2 курсе бакалавриата (4 семестр).

Виды учебной работы	В акад. часах
Общее количество часов	64
Лекция	32
Практика	32
Итоговый контроль	Диф.зачет

1.3. Взаимосвязь дисциплины с другими дисциплинами учебного плана специальности (направления)

«Введение в C/C++» (или аналогичные курсы по основам программирования на C/C++): Курс предполагает уверенное владение языками C и C++, включая работу с указателями, структурами данных и базовыми алгоритмами.

«Архитектура компьютера» (или аналогичные курсы по архитектуре ЭВМ и низкоуровневому программированию): Эти дисциплины предоставляют необходимую базу для понимания взаимодействия программ с аппаратным обеспечением на низком уровне и принципов работы системных вызовов.

1.4. Результаты освоения программы дисциплины:

Код компетенции	Наименование компетенции	Код индикатора достижения компетенций	Наименование индикатора достижений компетенций
ПК-7	способностью к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения	7.1	Знать методы и технологии разработки и применения системного и прикладного программного обеспечения
		7.2	Разрабатывать и применять алгоритмические и программные решения в области системного и прикладного программного обеспечения
		7.3	Владеть способностью разрабатывать и применять алгоритмические и программные решения в области системного и прикладного программного обеспечения

УК-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	2.1	Знает подходы в постановке задач для достижения поставленной цели, обладает знаниями в выборе оптимальных способов их решения
		2.2	Умеет, исходя из действующих правовых норм, имеющихся ресурсов и ограничений, выбирать оптимальные способы решения задач в профессиональной области для достижения поставленной цели
		2.3	Владеет навыками определения круга профессиональных задач в рамках поставленной цели; выбором оптимальных способов их решения с учетом действующих правовых норм и имеющихся ресурсов
ОПК-3	Способен применять и модифицировать математические модели для решения задач в области профессиональной деятельности	3.1	Демонстрирует умение анализировать, модифицировать и использовать математические модели в области профессиональной деятельности

		3.2	Умеет выбирать методы исследования математических моделей; строить и исследовать математические модели, применять и модифицировать их для решения задач в области профессиональной деятельности
		3.3	Владеет навыками применения математического аппарата к исследуемым математическим моделям

2. УЧЕБНАЯ ПРОГРАММА

2.1. Цели и задачи дисциплины

Цель дисциплины «Операционные системы» — сформировать у студентов всестороннее понимание архитектуры, принципов функционирования и внутренних механизмов современных операционных систем, а также развить навыки анализа и проектирования системных решений.

Для достижения этой цели ставятся следующие задачи:

- Изучить процессы начальной загрузки компьютерных систем и роль BIOS/UEFI.
- Понять принципы организации и работы файловых систем, включая концепцию виртуальных файловых систем и особенности конкретных реализаций (например, Ext4).
- Освоить ключевые механизмы управления памятью, такие как страничная организация, сегментация и свопинг.
- Разобрать различные алгоритмы планирования процессов и понять механизм переключения контекста.
- Изучить концепции виртуализации и эмуляции, а также их применение в современных вычислительных системах.

- Получить представление о принципах работы контейнерных технологий и их роли в разработке и развертывании приложений.
- Развить навыки анализа производительности и безопасности операционных систем.

2.2. Трудоемкость дисциплины и виды учебной работы (в академических часах и зачетных единицах)

Виды учебной работы	Всего, в акад. часах	Распределение по семестрам					
		4 сем	— сем	— сем	— сем.	— сем	— сем.
1	2	3	4	5	6	7	8
1.Общая трудоемкость изучения дисциплины по семестрам, в т. ч.:	144						
1.1. Аудиторные занятия, в т. ч.:	64	64					
1.1.1.Лекции	32	32					
1.1.2.Практические занятия, в т. ч.	32	32					
1.2. Самостоятельная работа, в т. ч.:	53	53					
1.3. Контроль	27	27					
Итоговый контроль (Экзамен, Зачет, диф. зачет - указать)		Диф.зачет					

2.3. Содержание дисциплины

2.3.1. Тематический план и трудоемкость аудиторных занятий (модули, разделы дисциплины и виды занятий) по рабочему учебному плану

Разделы и темы дисциплины	Всего (ак. часов)	Лекции (ак. часов)	Практ. Занятия (ак. часов)	Семинары (ак. часов)	Лабор. (ак. часов)
1	2=3+4+5+6 +7	3	4	5	6
Тема1: Введение в операционные системы. Обзор архитектуры	4	2	2		
Тема 2: Начальная загрузка системы и BIOS/UEFI	4	2	2		
Тема 3: Управление процессами	4	2	2		
Тема 4: Планирование процессов: Алгоритмы и механизмы.	4	2	2		

Тема 5: Управление памятью: Основы и непрерывное выделение.	4	2	2		
Тема 6: Управление памятью: Сегментация.	4	2	2		
Тема 7: Управление памятью: Страничная организация (Paging).	4	2	2		
Тема 8: Управление памятью: Виртуальная память и свопинг.	4	2	2		
Тема 9: Файловые системы: Обзор и общие концепции	4	2	2		
Тема 10: Виртуальные файловые системы (VFS)	4	2	2		
Тема 11: Реализация файловой системы Ext4	4	2	2		
Тема 12: Ввод-вывод и управление устройствами.	6	3	3		
Тема 13: Виртуализация и эмуляция.	6	3	3		
Тема 14: Контейнерные технологии.	8	4	4		
ИТОГО	64	32	32		

2.3.2. Краткое содержание разделов дисциплины в виде тематического плана

Тема 1. (Введение в операционные системы. Обзор архитектуры)

Общее представление об операционных системах, их функциях, типах и основных компонентах, а также их месте в компьютерной архитектуре.

Тема 2. (Начальная загрузка системы и BIOS/UEFI)

Изучение последовательности загрузки компьютера, роли базовой системы ввода-вывода (BIOS) и унифицированного расширяемого интерфейса прошивки (UEFI).

Тема 3. (Управление процессами)

Рассмотрение концепции процесса, его жизненного цикла, а также механизмов создания, завершения и управления процессами в ОС.

Тема 4. (Планирование процессов: Алгоритмы и механизмы.)

Изучение различных алгоритмов планирования (FIFO, Round Robin, приоритетное и др.) и детальный разбор механизма переключения контекста.

Тема 5. (Управление памятью: Основы и непрерывное выделение)

Введение в управление памятью, включая принципы адресного пространства и методы выделения непрерывных блоков памяти.

Тема 6. (Управление памятью: Сегментация.)

Изучение механизма сегментации как способа организации памяти, позволяющего программам работать с логическими блоками данных.

Тема 7. (Управление памятью: Страничная организация (Paging).)

Детальное рассмотрение страничной организации памяти, включая таблицы страниц, механизмы трансляции адресов и защиту памяти.

Тема 8. (Управление памятью: Виртуальная память и свопинг)

Понятие виртуальной памяти, её преимущества, а также механизмы замещения страниц (свопинг) и управления кешем.

Тема 9. (Файловые системы: Обзор и общие концепции)

Обзор архитектуры файловых систем, включая иерархию, дисковые структуры и принципы хранения файлов

Тема 10. (Виртуальные файловые системы (VFS))

Изучение концепции VFS как уровня абстракции, позволяющего операционной системе работать с различными типами файловых систем единообразно.

Тема 11. (Реализация файловой системы Ext4.)

Детальное рассмотрение одной из популярных реализаций файловых систем, её структуры, особенностей и механизмов работы.

Тема 12. (Ввод-вывод и управление устройствами.)

Изучение принципов организации подсистемы ввода-вывода, драйверов устройств и механизмов взаимодействия ОС с периферийными устройствами.

Тема 13. (Виртуализация и эмуляция.)

Рассмотрение концепций виртуализации (аппаратной и программной) и эмуляции, их применений и принципов работы.

Тема 14. (Контейнерные технологии.)

Изучение принципов работы контейнеров (например, Docker), их отличий от виртуальных машин и преимуществ для развертывания приложений.

2.4. Модульная структура дисциплины с распределением весов по формам контролей

Формы контролей	Вес формы (форм) текущего контроля в результирующей оценке текущего контроля (по модулям)		Вес формы промежуточного контроля в итоговой оценке промежуточного контроля		Вес итоговой оценки промежуточного контроля в результирующей оценке промежуточных контролей		Вес итоговой оценки промежуточного контроля в результирующей оценке промежуточных контролей (семестровой оценке)		Веса результирующей оценки промежуточных контролей и оценки итогового контроля в результирующей оценке итогового контроля
	М1 ¹	М2	М1	М2	М1	М2			
Вид учебной работы/контроля									
Контрольная работа <i>(при наличии)</i>									
Устный опрос <i>(при наличии)</i>									
Тест <i>(при наличии)</i>									
Лабораторные работы <i>(при наличии)</i>									
Письменные домашние задания <i>(при наличии)</i>									
Реферат <i>(при наличии)</i>									
Эссе <i>(при наличии)</i>									
Проект <i>(при наличии)</i>									
Другие формы <i>(при наличии)</i>									
Веса результирующих оценок текущих контролей в итоговых оценках промежуточных контролей						0.4			
Веса оценок промежуточных контролей в итоговых оценках промежуточных контролей						0.6			
Вес итоговой оценки 1-го промежуточного контроля в результирующей оценке промежуточных контролей									

¹ Учебный Модуль

Вес итоговой оценки 2-го промежуточного контроля в результирующей оценке промежуточных контролей								
Вес результирующей оценки промежуточных контролей в результирующей оценке итогового контроля								0.4
Вес итогового контроля (Экзамен/зачет) в результирующей оценке итогового контроля								0.6
	$\Sigma = 1$	$\Sigma = 1$	$\Sigma = 1$	$\Sigma = 1$	$\Sigma = 1$	$\Sigma = 1$	$\Sigma = 1$	$\Sigma = 1$

3. Теоретический блок

3.1. Материалы по теоретической части курса

1. Andrew S. Tanenbaum, Herbert Bos, 2015 Modern Operating Systems
2. Michael Kerrisk, The Linux Programming Interface
3. Robert Love, LINUX System Programming
4. Robert Love, Linux Kernel Development, Third Edition
5. Peter Jay Salzman, Michael Burian, Ori Pomerantz, The Linux Kernel Module Programming Guide
6. Programming via Operating Systems, Vahram Martirosyan, Feb 2022
7. Operating Systems: Design and Implementation, Andrew S. Tanenbaum Albert S. Woodhull, Jan 1997
8. Structured Computer Organization, Andrew S. Tanenbaum, Jan
9. Operating system design: the XINU approach, D. Comer Timothy V. Fossum, Jan
10. Linux Kernel in a nutshell, Greg Kroah-Hartman, Jan 2006

4. Фонды оценочных средств

4.1. Планы практических и семинарских занятий

Темы практических/семинарских занятий:

1. Работа с файловой системой и правами доступа.
2. Управление процессами в Linux.
3. Использование перенаправлений, конвейеров и стандартных потоков.
4. Основы написания Bash-скриптов.
5. Отладка программ в командной строке (gdb, strace).
6. Использование cron и systemd для автоматизации задач.
7. Введение в файловые системы и структура Ext4.
8. Работа с утилитами мониторинга ресурсов (top, htop, iotop).
9. Обзор контейнеризации и запуск Docker-контейнеров.

Формы: разбор задач, работа в терминале, мини-кейсы, обсуждение сценариев, командная работа.

4.2. Планы лабораторных работ и практикумов

Примерные лабораторные задания:

1. Навигация по файловой системе. Базовые команды (`ls`, `cd`, `cp`, `mv`, `rm`, `chmod`, `chown`).
2. Создание Bash-скриптов: автоматизация повседневных задач.
3. Обработка текстовых файлов с использованием `grep`, `sed`, `awk`.
4. Создание, управление и завершение процессов.
5. Использование `stop` для планирования задач.
6. Написание утилиты на C с применением системных вызовов (`fork()`, `exec()`, `wait()`).
7. Работа с файловыми дескрипторами и перенаправлением вывода.
8. Контейнеризация: установка и запуск Docker, сборка простого контейнера.

4.3. Материалы по практической части курса

4.3.2. Учебные справочники

- William Shotts. *The Linux Command Line*
- Michael Kerrisk. *Linux Programming Interface*
- `man-pages` — встроенная справка команд Linux

4.3.3. Задачники (практикумы)

- Практикум: Bash-скрипты и автоматизация (авторские материалы)
- Сборник задач по системному программированию (внутренний)
- Виртуальные лаборатории по Docker и виртуализации

4.3.4. Наглядно-иллюстративные материалы

- Презентации по архитектуре Linux, управлению процессами и памяти
- Схемы загрузки системы и взаимодействия ядра с процессами
- Скриншоты и видео с демонстрацией команд и утилит

4.3.5. Другие виды материалов

- Онлайн-песочницы: <https://replit.com>, <https://katacoda.com>, <https://linuxzoo.net>
- GitHub-репозитории с исходным кодом лабораторных работ
- Docker Hub-образы и примеры

4.4. Вопросы и задания для самостоятельной работы студентов

Примеры заданий:

- Написать Bash-скрипт, выполняющий резервное копирование домашней директории.
- Настроить выполнение задачи по расписанию через cron.
- Изучить ключевые команды для работы с файлами и процессами.
- Написать отчет по использованию инструментов `htop`, `lsuf`, `iotop`.
- Проанализировать структуру Ext4 в тестовой файловой системе.

4.5. Тематика рефератов, эссе и других форм самостоятельных работ

- Сравнительный анализ файловых систем Linux (Ext4, Btrfs, XFS)
- Архитектура ядра Linux: модульность и расширяемость
- Безопасность в Linux: модели управления доступом
- Развитие контейнеризации: от chroot до Docker и LXC
- Управление памятью в современных ОС: подходы и реализация

4.6. Образцы вариантов контрольных работ, тестов и/или других форм текущих и промежуточных контролей

- **Вариант контрольной работы:**
 1. Объясните различие между hard link и symbolic link.
 2. Напишите скрипт, проверяющий наличие заданного файла и логирующий результат.
 3. Опишите, как работает команда `ps aux | grep` и что означает каждая часть вывода.
- **Промежуточное тестирование:**

Тест с множественным выбором (20 вопросов) по темам: процессы, память, файловые системы.

4.7. Перечень экзаменационных вопросов

1. Основные функции операционных систем.
2. Алгоритмы планирования процессов.
3. Механизмы управления виртуальной памятью.
4. Работа с файловыми системами в Linux.
5. Концепция VFS и реализация Ext4.
6. Взаимодействие с оборудованием через подсистему ввода-вывода.
7. Архитектура контейнерных технологий (Docker).
8. Особенности системных вызовов `fork()`, `exec()`, `wait()`.

4.8. Образцы экзаменационных билетов

Билет №5:

1. Опишите процесс загрузки системы Linux (BIOS/UEFI, загрузчик, ядро).
2. Напишите и объясните Bash-скрипт для архивации логов.
3. Чем отличается сегментация от страничной организации памяти?

4.9. Образцы экзаменационных практических заданий

- Создайте и запустите скрипт, копирующий файлы из одной директории в другую и логирующий действия.
- Реализуйте на языке C программу, создающую дочерний процесс и передающую данные через pipe.
- Настройте выполнение скрипта с помощью cron на каждые 5 минут.

4.10. Банк тестовых заданий для самоконтроля

Примеры тестовых вопросов:

- Команда `chmod 755 file.sh` изменяет:
 - а) Владелец не может редактировать файл
 - б) Все могут выполнять файл
 - в) Только root может запускать
 - г) Никто не может читать файл**Ответ:** б
- Что делает `ps -ef | grep ssh`?
Ответ: Выводит все процессы, в имени которых содержится `ssh`.
- Какая команда используется для отображения текущего каталога?
Ответ: `pwd`

4.11. Методики решения и ответы к образцам тестовых заданий

- **Вопрос:** Как перенаправить `stderr` в файл?
Решение: `command 2> error.log`
Пояснение: `2>` указывает на перенаправление потока ошибок.
- **Вопрос:** Что означает `chmod +x script.sh`?
Решение: Делает файл исполняемым для всех пользователей.
Контекст: `x` — разрешение на выполнение; `+` — добавление разрешения.

5. Методический блок

5.1. Методика преподавания

Преподавание данной учебной дисциплины строится на сочетании лекционных занятий, практических семинаров и самостоятельной работы студентов.

На лекциях излагаются основные концепции программирования в среде Linux, включая работу с командной строкой, системными вызовами, управлением процессами, файловой системой и инструментами автоматизации. Рассматриваются как фундаментальные аспекты операционной системы, так и прикладные темы, такие как скриптинг на Bash, работа с компиляторами и отладчиками, взаимодействие с системными журналами.

Особое внимание уделяется практической направленности курса: студенты учатся писать и отлаживать программы в реальной среде, осваивают инструменты командной строки и получают навыки, необходимые для разработки и администрирования в Unix-подобных системах.