

**Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»**

УТВЕРЖДЕНО

**Директор физтех-школы
радиотехники и компьютерных
технологий**

Д.А. Гаврилов

	Рабочая программа дисциплины (модуля)
по дисциплине:	Структура и алгоритмы ядра операционной системы
по направлению:	Прикладные математика и физика
профиль подготовки:	Радиотехника и компьютерные технологии Физтех-школа Радиотехники и Компьютерных Технологий кафедра информатики и вычислительной техники
курс:	4
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 7 (осенний) - Дифференцированный зачет

Аудиторных часов: 30 всего, в том числе:

лекции: 0 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 15 час.

Всего часов: 45, всего зач. ед.: 1

Программу составил: А.В. Фёдоров, ассистент

Программа обсуждена на заседании кафедры информатики и вычислительной техники 30.08.2023

Аннотация

Курс "Структура и алгоритмы ядра операционной системы" предназначен для изучения основных структур и алгоритмов ядра современной операционной системы. Изучение происходит на примере ядра ОС Linux. Студенты изучают такие механизмы ядра как управление оперативной памятью, многозадачный режим работы, алгоритмы планирования процессов, объектно-ориентированную модель устройств и драйверов, подсистему ввода-вывода. Курс проходит в формате семинарских занятий. Для успешного прохождения данного курса необходимо посещение и конспектирование семинаров и своевременное выполнение практических работ, а также самостоятельная работа с дополнительной литературой.

1. Цели и задачи

Цель дисциплины

- ознакомить слушателей со структурой и алгоритмами ядра современной операционной системы (ОС), изучение происходит на примере реализации ядра Linux. Рассматриваются такие базовые механизмы ядра как управления оперативной памятью, организация многозадачного режима работы, алгоритмы планирования процессов, подсистема ввода-вывода, организация планирования в подсистеме блочных устройств, объектно-ориентированная модель устройств и драйверов.

Задачи дисциплины

- формирование системного восприятия устройства ядра современной операционной системы и понимания основных подсистем ядра, их базовых структур, интерфейсов и алгоритмов, а также особенностей программной реализации;
- освоение студентами базовых принципов организации взаимодействия операционной системы с пользователем и аппаратным обеспечением.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1 Анализирует задачу, выделяя этапы ее решения, действия по решению задачи
	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.3 Рассматривает различные варианты решения задачи, оценивает их преимущества и недостатки
	УК-1.4 Грамотно, логично, аргументированно формирует собственные суждения и оценки
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук, и использовать их в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности
	ОПК-4.4 Владеет навыками работы с компьютером и компьютерными сетями с целью получения, хранения и обработки научной (технической, технологической) информации
ПК-4 Способен критически оценивать	ПК-4.1 Знает численные порядки величин, характерных для соответствующей профессиональной области

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- Состав и назначение основных подсистем ядра ОС Linux: подсистема ввода вывода, виртуальная память, процессы и планировщик.
- Основные принципы организации процессов и потоков в ядре ОС Linux.
- Принципы организации DDM (device-driver model) модели ядра, устройство интерфейса sysfs и понятие kobject.
- Основные алгоритмы планирования процессов и их характеристики, основные особенности работы ОС в режиме реального времени.
- Устройство интерфейса системных вызовов, понимать организацию взаимодействия пользовательских программ с ядром ОС через ABI.
- Устройство виртуальной файловой системы VFS и файлового интерфейса Unix.
- Основные алгоритмы планирования дисковых операций блочной подсистемы, их характеристики, особенности и область применения.
- Принципы организации файловых систем, основные виды современных файловых систем.
- Основы внутреннего устройства современных файловых систем ext3, ext4, btrfs
- Принципы организации виртуальной памяти, страничная модель. Поддержка виртуальной памяти со стороны ЦПУ.
- Особенности инициализации оборудования материнской платы из ядра ОС Linux для встраиваемых систем.
- Основные виды виртуализации: гипервизоры 1-го и 2-го типов, виртуализация на уровне ядра ОС.
- Понимать основные концепции построения UNIX-подобных операционных систем.

уметь:

- Программировать простейший модуль ядра, организующий взаимодействие с пользователем через файловый интерфейс.
- Программировать пользовательское приложение взаимодействующее с ядром через системные вызовы open, read, write, ioctl.
- Производить конфигурирование ядра Linux и его компиляцию под заданные требования.
- Программировать тестовые нагрузки на различные подсистемы ядра.
- Производить тестирования различных подсистем ядра (виртуальная память, подсистема блочного ввода-вывода, сетевой стек) как с помощью штатных средств ОС, так и с помощью самостоятельно разработанных тестов.
- Использовать средства разработки (vim, ctags, cscope, git, IDE) для самостоятельного изучения кода ядра ОС Linux.

владеть:

- Технологией разработки ядра модулей ядра ОС Linux.
- Технологией кросс-компиляции ядра ОС Linux под различные аппаратные платформы.
- Стандартными средствами тестирования и анализа производительности ядра ОС: unixbench, LTP, fio, pktgen, iperf, top, iotop, vmstats.

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

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа

1	Многопоточный и асинхронный режим работы ядра ОС. Процессы, потоки, прерывания и исключительные ситуации . Синхронизация		6		2
2	Файловые системы, блочная подсистема ввода-вывода . Системы хранения данных		8		4
3	Виртуальная память		6		3
4	Подсистема драйверов и device driver model		8		4
5	Виртуализация и облачные технологии		2		2
Итого часов			30		15
Подготовка к экзамену		0 час.			
Общая трудоёмкость		45 час., 1 зач.ед.			

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 7 (Осенний)

1. Многопоточный и асинхронный режим работы ядра ОС. Процессы, потоки, прерывания и исключительные ситуации . Синхронизация

1. Введение. История развития ОС. Различные виды ОС, их особенности зависящие от области применения.
2. Процессы, дескриптор процесса - структура `task_struct`, состояния процесса и его жизненный цикл. Системные вызовы `fork` и `exec`, дерево процессов. Процессы и потоки, системный вызов `clone`, `copy-on-write`.
3. Многопоточная обработка в ядре Linux, проблема синхронизации. Прimitives синхронизации, и их аппаратная поддержка (`cmpxchg`). Взаимоблакировки и тупиковые ситуации, состояние гонки.
4. Многозадачный режим работы и разделение времени, проблемы многозадачности. Алгоритмы планирования `O(1)`, `CFS`, `BF-scheduler`, проблема балансировки нагрузки. Планирование в режиме реального времени: `FIFO` и `RR`.
5. Прерывания и исключительные ситуации. Обработка прерывания в ядре Linux, взаимодействие с контроллерами прерываний.

2. Файловые системы, блочная подсистема ввода-вывода . Системы хранения данных

1. Виртуальная файловая система - `VFS`, файловый интерфейс и `unix-way`. Основные структуры `VFS`, понятие файла, директории. Ссылки и виды ссылок. Атрибуты и дискреционная модель доступа.
2. Блочная система ввода вывода. Структуры `buffer-head` и `bio`, буферизация ввода-вывода и дисковый кэш. Планирования ввода-вывода, очередь запросов основные алгоритмы планирования: `Linux Elevator`, `Deadline`, `Noop`, `CFQ`, `AS (anticipation heuristics)`. Альтернативная реализация блочной подсистемы — `blk-mq`. Производительность дисковой подсистемы.
3. Файловые системы: `proc`, `sys`, `debugfs`, `ext3/ext4`, `btrfs`
4. Современные системы хранения данных (СХД), `software-defined СХД`. Дисковые массивы, технологии `RAID` и `Fiber Channel (FC)`. `ZFS`, `Ceph`

3. Виртуальная память

1. Виртуальная память, страничная модель. Физические и виртуальные адреса, трансляция адресов и аппаратная поддержка. Аллокаторы памяти в ядре Linux.

2. Интерфейс системных вызовов. ABI, формат ELF, разделяемые библиотеки и динамический загрузчик.

4. Подсистема драйверов и device driver model

Device-Driver Model (DDM) – объектно-ориентированная модель организации подсистемы устройств и драйверов ядра linux. Блочные и символьные устройства. Загружаемые модули. Интерфейс sysfs и kobject.

5. Виртуализация и облачные технологии

Технология виртуализации. Гипервизоры 1-го и 2-го типов, виртуализация на уровне ядра ОС. Cgroups, namespaces, контейнеры LXC. Облачные технологии на примере OpenStack.

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Учебная аудитория, оснащенная компьютером и мультимедийным оборудованием (проектор, звуковая система).

6. Перечень рекомендуемой литературы

Основная литература

1. Современные операционные системы [Текст] : [учеб. пособие для вузов] / Э. Таненбаум ; [пер. с англ. Н. Вильчинский, А. Лашкевич] .— 3-е изд. — СПб. : Питер, 2015 .— 1120 с
2. Архитектура компьютера [Текст] : [учеб. пособие для вузов] / Э. Таненбаум, Т. Остин ; [пер. с англ. Е. Матвеев] .— 6-е изд. — СПб. : Питер, 2014 .— 816 с

Дополнительная литература

1. Язык программирования C [Текст] : [учеб. пособие для вузов] / Б. Керниган, Д. Ритчи ; пер. с англ. и ред. В. Л. Бродового .— 2-е изд., перераб. и доп. — М. : Вильямс, 2006, 2007, 2009, 2010, 2012, 2013, 2015 .— 304 с.
2. Современные операционные системы [Текст], [учеб. пособие для вузов] / Э. Таненбаум, Х. Бос ; [пер. с англ.]. -СПб., Питер, 2017

7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

1. Linux Device Drivers 3rd edition, Jonathan Corbet, Alessandro Rubini, and Greg Kroah-Hartman, <https://lwn.net/Kernel/LDD3/>
2. Understanding The Linux Virtual Memory Manager, Mel Gorman, 2007 <https://www.kernel.org/doc/gorman/pdf/understand.pdf>
3. Linux Kernel on-line documentation <https://www.kernel.org/doc/html/latest/index.html>
4. <http://kernel.org>
5. <http://lwn.net>

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

На занятиях используются мультимедийные технологии, включая демонстрацию презентаций.

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Студент, изучающий дисциплину, должен с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике.

В результате изучения дисциплины студент должен знать основные определения, понятия.

Успешное освоение курса требует напряжённой самостоятельной работы студента. В программе курса приведено минимально необходимое время для работы студента над темой. Самостоятельная работа включает в себя:

- чтение и конспектирование рекомендованной литературы;
- проработку учебного материала (по конспектам, учебной и научной литературе);
- подготовку к дифференцированному зачету.

Руководство и контроль за самостоятельной работой студента осуществляется в форме индивидуальных консультаций.

Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов, следует обращаться за консультациями к преподавателю.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Прикладные математика и физика
профиль подготовки:	Радиотехника и компьютерные технологии Физтех-школа Радиотехники и Компьютерных Технологий кафедра информатики и вычислительной техники
курс:	4
квалификация:	бакалавр
Семестр, формы промежуточной аттестации: 7 (осенний) - Дифференцированный зачет	
Разработчик:	А.В. Фёдоров, ассистент

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1 Анализирует задачу, выделяя этапы ее решения, действия по решению задачи
	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.3 Рассматривает различные варианты решения задачи, оценивает их преимущества и недостатки
	УК-1.4 Грамотно, логично, аргументированно формирует собственные суждения и оценки
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук, и использовать их в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности
	ОПК-4.4 Владеет навыками работы с компьютером и компьютерными сетями с целью получения, хранения и обработки научной (технической, технологической) информации
ПК-4 Способен критически оценивать применимость используемых методик и методов	ПК-4.1 Знает численные порядки величин, характерных для соответствующей профессиональной области
	ПК-4.3 Способен обосновать причинно-следственные отношения используемых понятий и моделей

2. Показатели оценивания компетенций

В результате изучения дисциплины «Структура и алгоритмы ядра операционной системы» обучающийся должен:

знать:

- Состав и назначение основных подсистем ядра ОС Linux: подсистема ввода вывода, виртуальная память, процессы и планировщик.
- Основные принципы организации процессов и потоков в ядре ОС Linux.
- Принципы организации DDM (device-driver model) модели ядра, устройство интерфейса sysfs и понятие kobject.
- Основные алгоритмы планирования процессов и их характеристики, основные особенности работы ОС в режиме реального времени.
- Устройство интерфейса системных вызовов, понимать организацию взаимодействия пользовательских программ с ядром ОС через ABI.
- Устройство виртуальной файловой системы VFS и файлового интерфейса Unix.
- Основные алгоритмы планирования дисковых операций блочной подсистемы, их характеристики, особенности и область применения.
- Принципы организации файловых систем, основные виды современных файловых систем.
- Основы внутреннего устройства современных файловых систем ext3, ext4, btrfs
- Принципы организации виртуальной памяти, страничная модель. Поддержка виртуальной памяти со стороны ЦПУ.
- Особенности инициализации оборудования материнской платы из ядра ОС Linux для встраиваемых систем.
- Основные виды виртуализации: гипервизоры 1-го и 2-го типов, виртуализация на уровне ядра ОС.
- Понимать основные концепции построения UNIX-подобных операционных систем.

уметь:

- Программировать простейший модуль ядра, организующий взаимодействие с пользователем через файловый интерфейс.
- Программировать пользовательское приложение взаимодействующее с ядром через системные вызовы open, read, write, ioctl.
- Производить конфигурирование ядра Linux и его компиляцию под заданные требования.
- Программировать тестовые нагрузки на различные подсистемы ядра.
- Производить тестирования различных подсистем ядра (виртуальная память, подсистема блочного ввода-вывода, сетевой стек) как с помощью штатных средств ОС, так и с помощью самостоятельно разработанных тестов.
- Использовать средства разработки (vim, ctags, cscope, git, IDE) для самостоятельного изучения кода ядра ОС Linux.

владеть:

- Технологией разработки ядра модулей ядра ОС Linux.
- Технологией кросс-компиляции ядра ОС Linux под различные аппаратные платформы.
- Стандартными средствами тестирования и анализа производительности ядра ОС: unixbench, LTP, fio, pktgen, iperf, top, iotop, vmstats.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

1. Понятие процессов и потоков в Linux. Структура task_struct, жизненный цикл процесса.
2. Системные вызовы fork, exec, clone. Принцип copy-on-write.
3. Примитивы синхронизации: spinlock, mutex.
4. spinlock - аппаратная поддержка примитивов синхронизации.
5. Планировщики ядра linux - O(1), CFS, FIFO, RR
6. Прерывания в ядре linux
7. Виртуальная файловая система VFS, понятие файла и директории
8. Ссылки. Атрибуты и дискреционная модель доступа.
9. Блочная система ввода вывода, алгоритмы планирования блочных операций.
10. Файловые системы proc, sys, ext4, btrfs
11. Виртуальная память, страничная модель.
12. Аллокаторы памяти в ядре Linux.
13. Интерфейс системных вызовов.
14. Подсистема драйверов и device driver model
15. Виртуализация. Подсистема драйверов и device driver model

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

1. Понятие процесса, дескриптор процесса. Жизненный цикл процесса, дерево процессов, системные вызовы `fork` и `exec`.
2. Потоки: `pthread` и `kthread`, системный вызов `clone`, `copy-on-write`.
3. Примитивы синхронизации, и их аппаратная поддержка (`cmpxchg`). Взаимоблокировки и тупиковые ситуации, состояние гонки.
4. Многозадачный режим работы и разделение времени, проблемы многозадачности. Алгоритмы планирования $O(1)$, CFS, BF-scheduler, проблема балансировки нагрузки.
5. Планирование в режиме реального времени: FIFO и RR.
6. Прерывания и исключительные ситуации. Обработка прерывания в ядре Linux, взаимодействие с контроллерами прерываний. Верхняя и нижняя половины обработчика прерываний.
7. Виртуальная файловая система. Основы структуры VFS, понятие файла, директории. Ссылки и виды ссылок. Атрибуты и дискреционная модель доступа.
8. Блочная система ввода вывода. Структуры `buffer-head` и `bio`, буферизация ввода-вывода и дисковый кэш.
9. Планирования ввода-вывода, очередь запросов основные алгоритмы планирования: Linux Elevator, Deadline, Noop, CFQ, AS (anticipation heuristics).
10. Альтернативная реализация блочной подсистемы — `blk-mq`. Производительность дисковой подсистемы.
11. Файловые системы: `proc`, `sys`, `debugfs`, `ext3/ext4`, `btrfs`.
12. Современные системы хранения данных (СХД), software-defined СХД. Дисковые массивы, технологии RAID и Fiber Channel (FC). ZFS, Ceph.
13. Виртуальная память, страничная модель. Физические и виртуальные адреса, трансляция адресов и аппаратная поддержка.
14. Аллокаторы памяти в ядре Linux — SLUB и Buddy.
15. Интерфейс системных вызовов. ABI, программное соглашение.
16. Формат ELF, разделяемые библиотеки и динамический загрузчик. Статическая и динамическая линковка.
17. Device-Driver Model (DDM) – объектно-ориентированная модель организации подсистемы устройств и драйверов ядра Linux. Блочные и символьные устройства, шины, адаптеры, конечные устройства.
18. Загружаемые модули. Интерфейс `sysfs` и `kobject`. Взаимодействие через файловый интерфейс, системный вызов `ioctl`.
19. Технология виртуализации. Гипервизоры 1-го и 2-го типов, виртуализация на уровне ядра ОС.
20. Cgroups, namespaces, контейнеры LXC.
21. Облачные технологии на примере OpenStack.

Билет 1.

1. Понятие процесса, дескриптор процесса. Жизненный цикл процесса, дерево процессов, системные вызовы `fork` и `exec`.
2. Облачные технологии на примере OpenStack.

Билет 2.

1. Потоки: `pthread` и `kthread`, системный вызов `clone`, `copy-on-write`.
2. Cgroups, namespaces, контейнеры LXC.

Критерии оценивания

Оценка отлично (10) выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, проявляющему интерес к данной предметной области, продемонстрировавшему умение уверенно и творчески применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (9) выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (8) выставляется студенту, показавшему систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочетами.

Оценка хорошо (7) выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но недостаточно грамотно обосновывает полученные результаты.

Оценка хорошо (6) выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности.

Оценка хорошо (5) выставляется студенту, если он в основном знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач достаточно большое количество неточностей.

Оценка удовлетворительно (4) выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он освоил основные разделы учебной программы, необходимые для дальнейшего обучения, и может применять полученные знания по образцу в стандартной ситуации.

Оценка удовлетворительно (3) выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, допускающему ошибки в формулировках базовых понятий, нарушения логической последовательности в изложении программного материала, слабо владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и с трудом применяет полученные знания даже в стандартной ситуации.

Оценка неудовлетворительно (2) выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных принципов и не умеет использовать полученные знания при решении типовых задач.

Оценка неудовлетворительно (1) выставляется студенту, который не знает основного содержания учебной программы дисциплины, допускает грубейшие ошибки в формулировках базовых понятий дисциплины и вообще не имеет навыков решения типовых практических задач.

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Дифференцированный зачет проводится в устной форме.

При проведении зачета обучающемуся предоставляется 20 минут на подготовку. Опрос обучающегося проводится в течение 40 минут.

Во время проведения дифференцированного зачета обучающиеся могут пользоваться программой дисциплины.