

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

**УТВЕРЖДЕНО**

**Директор высшей школы  
программной инженерии  
А.В. Малеев**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Низкоуровневое программирование
<b>по направлению:</b>	Программная инженерия
<b>профиль подготовки:</b>	Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии
<b>курс:</b>	2
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 3 (осенний) - Экзамен

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 18 час.

Подготовка к экзамену: 30 час.

Всего часов: 108, всего зач. ед.: 3

Программу составил: И.С. Мещерин, старший преподаватель

Программа обсуждена на заседании высшей школы программной инженерии 14.03.2025

## Аннотация

Данная дисциплина является адаптацией под учебный план ИВТ МФТИ дисциплины "Operating System Engineering" американского университета MIT.

В рамках данной дисциплины будет изучено внутреннее устройство операционных систем на примере учебно-исследовательской системы XV6.

Будет рассказано об функционировании отдельных компонент ядра. На практических занятиях требуется воспроизвести из исходных текстов работоспособную систему XV6 для работы в эмуляторе, и исследовать её, модифицируя отдельные подсистемы.

## 1. Цели и задачи

### Цель дисциплины

- познакомить студентов с внутренним устройством операционных систем.

### Задачи дисциплины

- разобрать внутреннее устройство отдельных компонент ядра и стандартных библиотек операционной системы.

## 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-1.4 Грамотно, логично, аргументированно формирует собственные суждения и оценки
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.1 Способен использовать информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности
	ОПК-2.2 Владеет навыками создания программного обеспечения для ЭВМ и систем различной архитектуры
ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.2 Способен оценивать концепции и атрибуты качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роль людей, процессов, методов, инструментов и технологий обеспечения качества
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.2 Умеет применять языки программирования для решения прикладных задач
	ОПК-6.3 Знает методы тестирования программного кода на ошибки и способен проводить тестирование на различных уровнях (модульное, интеграционное, системное)
	ОПК-6.4 Имеет навыки программирования и тестирования программных продуктов
ПК-3 Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение	ПК-3.1 Различает синтаксис языков программирования, особенности программирования на этих языках, стандартные библиотеки языков программирования
	ПК-3.2 Умеет выбирать языки программирования для написания программного кода с учетом технического задания
	ПК-3.3 Умеет излагать основные принципы построения и виды архитектуры программного обеспечения, методы и средства проектирования программного обеспечения, методологии разработки программного обеспечения и технологии программирования

	ПК-3.4 Знает, как определять оптимальные методы и средства проектирования программного обеспечения и структур данных
ПК-5 Способен проектировать, разрабатывать, внедрять, сопровождать и снимать с эксплуатации информационные системы	ПК-5.3 Умеет разрабатывать прототип информационных систем в соответствии с требованиями и проводить его тестирование для проверки корректности архитектурных решений

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- внутреннее устройство ядра;
- внутреннее устройство стандартной библиотеки.

уметь:

- разрабатывать и отлаживать код под эмуляторами;
- модифицировать код ядра операционной системы.

владеть:

- навыками кросс-компиляции и работы с эмуляторами;
- навыками исследования исходного кода.

### 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	Инструменты виртуализации и разработки	2	2		2
2	Таблицы страниц памяти	4	4		2
3	Прерывания и системные вызовы	4	4		2
4	Аппаратные прерывания и драйверы устройств	4	4		2
5	Блокировки	4	4		2
6	Планирование задач	4	4		2
7	Устройство файловой системы	4	4		2
8	Организация операционных систем	4	4		4
Итого часов		30	30		18
Подготовка к экзамену		30 час.			
Общая трудоёмкость		108 час., 3 зач.ед.			

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

Семестр: 3 (Осенний)

##### 1. Инструменты виртуализации и разработки

Виртуальные машины: с полностью программной эмуляцией, и с использованием аппаратного ускорения. Кросс-компиляция кода под произвольные платформы. Работа с отдачиком в виртуальной машине qemu.

## 2. Таблицы страниц памяти

Аппаратные средства поддержки страничной модели памяти. Адресное пространство ядра. Выделение физической памяти. Адресное пространство процесса. Реализация системного вызова `sbrk`.

## 3. Прерывания и системные вызовы

Вектор прерываний. Обработка аппаратных прерываний. Программные прерывания. Их использование для реализации механизма системных вызовов. Программные прерывания во время выполнения кода ядра. Обработка прерываний Page-Fault для реализации совместного использования страниц памяти.

## 4. Аппаратные прерывания и драйверы устройств

Реализация драйвера консольного ввода-вывода через UART с использованием прерываний. Прерывания по таймеру. Их использование для реализации механизма многозадачности. Проблема одновременной обработки нескольких прерываний.

## 5. Блокировки

Проблема гонки состояний. Объекты блокировки и их использование. Проблема взаимной блокировки. Взаимная блокировка во время прерывания. Модели памяти и влияние оптимизаций компилятора на поведение многопоточного кода. Короткие по времени блокировки.

## 6. Планирование задач

Кооперативная и вытесняющая многозадачность. Переключение процессов и его реализация в ядре. Алгоритмы планирования задач и приоритеты процессов. Реализация запуска, ожидания и удаления процессов. Реализация механизма неименованных каналов.

## 7. Устройство файловой системы

Уровень организации файловых систем. Уровень кэширования и его реализация. Уровень журналирования и его реализация. Уровень отдельных файлов и каталогов. Уровень файловых дескрипторов.

## 8. Организация операционных систем

Уровни абстракций аппаратных ресурсов. Режим пользователя, супервизора и понятие системных вызовов. Основные компоненты ядра. Основные компоненты системы XV6.

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

Учебная аудитория, оснащенная компьютерами с OS Linux для каждого студента, либо с предустановленной системой виртуализации.

## 6. Перечень рекомендуемой литературы

Основная литература

1. Чистый код : создание, анализ и рефакторинг / Р. Мартин. — Санкт-Петербург, Питер, 2019.— URL: <https://ibooks.ru/bookshelf/361844/reading> (дата обращения: 24.11.2020). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)

2. Современные операционные системы, [учеб. пособие для вузов] / Э. Таненбаум, Х. Бос . — Санкт-Петербург, Питер, 2019.— URL: <https://ibooks.ru/bookshelf/364626/reading> (дата обращения: 24.11.2020). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)

Рекомендуемая литература для самостоятельного изучения:

1. Low-Level Programming. 2017. Igor Zhirkov

2. Handbook of Applied Cryptography. 1997. Alfred J. Menezes, Paul C. van Oorschot, & Scott A. Vanstone.

Дополнительная литература

Рекомендуемая литература для самостоятельного изучения:

1. Linux и Go. Эффективное низкоуровневое программирование. Цилурик Олег Иванович

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

1. Учебное пособие университета MIT по аналогичной дисциплине (на английском языке):

<https://pdos.csail.mit.edu/6.828/2020/xv6/book-riscv-rev1.pdf>

2. <https://web.archive.org/web/20150822173159/http://www.wasm.ru/wault>

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

Стандартные средства разработки, входящие в состав ОС Linux.

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

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:

– проработку учебного материала (по конспектам лекций, учебной и научной литературе), подготовку ответов на вопросы, предназначенных для самостоятельного изучения.

Промежуточный контроль знаний проводится в виде реализации лабораторных работ и их защит.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Программная инженерия
<b>профиль подготовки:</b>	Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии
<b>курс:</b>	<u>2</u>
<b>квалификация:</b>	бакалавр
Семестр, формы промежуточной аттестации: 3 (осенний) - Экзамен	
<b>Разработчик:</b>	И.С. Мещерин, старший преподаватель

## 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-1.4 Грамотно, логично, аргументированно формирует собственные суждения и оценки
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.1 Способен использовать информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности
	ОПК-2.2 Владеет навыками создания программного обеспечения для ЭВМ и систем различной архитектуры
ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.2 Способен оценивать концепции и атрибуты качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роль людей, процессов, методов, инструментов и технологий обеспечения качества
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.2 Умеет применять языки программирования для решения прикладных задач
	ОПК-6.3 Знает методы тестирования программного кода на ошибки и способен проводить тестирование на различных уровнях (модульное, интеграционное, системное)
	ОПК-6.4 Имеет навыки программирования и тестирования программных продуктов
ПК-3 Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение	ПК-3.1 Различает синтаксис языков программирования, особенности программирования на этих языках, стандартные библиотеки языков программирования
	ПК-3.2 Умеет выбирать языки программирования для написания программного кода с учетом технического задания
	ПК-3.3 Умеет излагать основные принципы построения и виды архитектуры программного обеспечения, методы и средства проектирования программного обеспечения, методологии разработки программного обеспечения и технологии программирования
	ПК-3.4 Знает, как определять оптимальные методы и средства проектирования программного обеспечения и структур данных
ПК-5 Способен проектировать, разрабатывать, внедрять, сопровождать и снимать с эксплуатации информационные системы	ПК-5.3 Умеет разрабатывать прототип информационных систем в соответствии с требованиями и проводить его тестирование для проверки корректности архитектурных решений

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Низкоуровневое программирование» обучающийся должен:

### знать:

- внутреннее устройство ядра;
- внутреннее устройство стандартной библиотеки.

### уметь:

- разрабатывать и отлаживать код под эмуляторами;
- модифицировать код ядра операционной системы.

### владеть:

- навыками кросс-компиляции и работы с эмуляторами;
- навыками исследования исходного кода.

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

1. Выполнение арифметических операций в шестнадцатеричной системе счисления.
2. Разработка программ на языке ассемблера в интегрированной среде Microsoft Visual Studio. Структура программы на языке ассемблера.
3. Регистры. Организация памяти и режимы адресации. Команды пересылки данных.
4. Арифметические команды целочисленного устройства микропроцессора.
5. Двоично-десятичная арифметика
6. Команды логических операций.
7. Команды передачи управления.
8. Команды сдвигов.
9. Подпрограммы и вызов процедур.
10. Цепочечные команды.
11. Реализация для XV6 функциональность сбора статистической информации о свободной памяти и количестве запущенных процессов.
12. Реализация системного вызова `brk`. Не забывайте про необходимость освобождения выделенных страниц памяти.
13. Реализация переключения между потоками в планировщике заданий.
14. Модифицирование кода стандартной файловой системы XV6 таким образом, чтобы увеличить максимально допустимый размер файла.

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

1. Режимы работы процессора: пользовательский и привилегированный. Системные вызовы
2. Организация ядра. Структура кода операционной системы XV6
3. Процессы. Реализацию запуска процесса в операционной системе
4. Аппаратная поддержка страничной адресации
5. Адресное пространство ядра и адресное пространство процесса
6. Реализация выделения памяти для ядра и для процесса
7. Прерывания. Обработка ядром системных вызовов
8. Реализация драйверов устройств на примере текстового терминала
9. Проблема конкурентности в драйверах устройств
10. Проблема взаимной блокировки. Упорядочиванием блокировок
11. Переключения контекстов в планировщике задач
12. Реализация простоя в планировщике задач
13. Уровни абстракции в реализации файловых систем
14. Буферизация ввода-вывода в реализации файловой системы
15. Устройство файловой системы на уровне указателей на файлы

Примеры билетов:

Билет 1.

1. Стадии сборки в C++.
2. Кэши процессора и их применение.

Билет 2.

1. Инструкции безусловного и условного перехода в ассемблере.
2. Системные вызовы `fork` и `exec`.

Критерии оценивания  
отлично



10 всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;

9 систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений;

8 глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, правильное обоснование принятых решений;

хорошо

7 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;

6 знает материал, грамотно излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

5 знает основной материал, грамотно излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач неточности;

удовлетворительно

4 фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

3 характер знаний достаточен для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

неудовлетворительно

2 не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет правильно использовать полученные знания при решении типовых практических задач.

1 не знает формулировок основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

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

Итоговый контроль проводится в формате устного и (или) письменного экзамена.

Время отведенное на экзамен: 4 академических часа.