

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

**УТВЕРЖДЕНО**

**Директор физтех-школы  
прикладной математики и  
информатики**

**А.М. Райгородский**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Архитектура компьютеров и операционные системы
<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Математическое моделирование и компьютерные технологии Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
<b>курс:</b>	2
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 3 (осенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 75 час.

Всего часов: 135, всего зач. ед.: 3

Программу составил: О.Н. Ивченко, заведующий кафедрой

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 12.02.2024

## Аннотация

Курс посвящен низкоуровневым аспектам разработки программного обеспечения для UNIX-подобных операционных систем, а также отработки навыков написания программ и их тестирования в предельных ситуациях. В рамках данной дисциплины будут немного затронуты программирование на языках ассемблера под архитектуры компьютеров ARM (32 бит) и x86, в объеме, минимально необходимом для понимания таких аспектов, как работа с памятью, соглашения о вызовах, и способы системных вызовов. После прохождения тем про язык ассемблера, оставшаяся часть курса будет посвящена изучению системных вызовов для работы с памятью, файлами, процессами. Особое внимание будет уделено механизмам межпроцессных взаимодействий: сигнала, каналам, разделяемой памяти, и сетевому взаимодействию.

Заключительным этапом всего курса является дифференцированный зачет, целью которого является проверка знаний студентов по теории и выявление практических навыков, полученных при выполнении практических заданий.

### 1. Цели и задачи

#### Цель дисциплины

- познакомить студентов с базовыми принципами организации внутренней организации компьютерных систем, с базовыми принципами организации операционных систем, а также абстракций и интерфейсов, которые предоставляются программисту для взаимодействия с операционной системой.

#### Задачи дисциплины

- задача дисциплины заключается в демонстрации базовых принципов на примере операционных систем семейства UNIX и, частично, Windows.

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
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-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
	ОПК-2.3 Знает основные требования информационной безопасности

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

В результате освоения дисциплины обучающиеся должны

знать:

- основы работы в UNIX-подобных системах;
- основы низкоуровневого программирования;
- основы машинного кода, языков ассемблера;
- различные пути повышения производительности программы;
- основы сетевого взаимодействия;
- основы устройства сетей.

уметь:

- создавать многопоточные и межсетевые программы на языках Си и Ассемблер;
- работать в UNIX-подобных средах;
- создавать программы на языках Си и Ассемблер без использования высокоуровневых библиотек.

владеть:

- навыками ведения простейших программных проектов в системах контроля версий.

#### 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	Цифровые электронно-вычислительные машины. Основные концепции и история развития.	4	4		10
2	Аппаратура современного компьютера. Архитектура x86: история развития, аппаратная реализация и язык ассемблера.	5	5		10
3	Инструментальные средства разработки и отладки программ в ОС Linux. Инструментарий GNU.	4	4		10
4	Семейство операционных систем Linux: базовые сервисы и использование POSIX API.	4	4		10
5	Операционная система Linux: развертывание и работа с помощью интерфейса командной строки.	5	5		10
6	Операционные системы: основы разработки и история развития.	4	4		10
7	Семейство операционных систем Linux: архитектура и программирование на уровне ядра ОС.	4	4		15
Итого часов		30	30		75
Подготовка к экзамену		0 час.			
Общая трудоёмкость		135 час., 3 зач.ед.			

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

Семестр: 3 (Осенний)

1. Цифровые электронно-вычислительные машины. Основные концепции и история развития.

Вычисления. Модели вычислений. Системы счисления. Кодирование информации. Понятие

компьютера. Цифровые электронно-вычислительные машины. Платформы и парадигмы построения компьютеров. История развития. Дискретные элементы цифровых электронных схем. Логические элементы. Интегральные схемы. Передача сигналов и хранение информации. Информационные шины. История развития цифровой вычислительной техники. Общие методы повышения эффективности вычислений. Закон Мура. Конвейеризация и распараллеливание. Программное обеспечение: ассемблеры, языки программирования высокого уровня, Unix, BSD, DOS, Windows, MacOS, GNU, Minix, Linux, iOS, Android, Windows Mobile.

2. Аппаратура современного компьютера. Архитектура x86: история развития, аппаратная реализация и язык ассемблера.

Краткий практический обзор, без которого объяснение происходящего в современном цифровом компьютере было бы невозможно: использование логических элементов для построения. Двоичной арифметики и памяти, регистры, адресация памяти, стек, вызов подпрограмм, прерывания, защита памяти, виртуальная память. Иллюстрация на примере архитектур компании Intel. Гонка производительности RISC и CISC, кешы, суперскалярность, предсказания ветвлений, многоядерность, микропрограммирование, альтернативы Intel. Эволюция архитектур компании Intel от 8008 до Intel 64. Языки ассемблера для архитектур IA-32 и Intel 64. Инструментальные средства программиста. Организация программы. Наборы команд. Описание данных. Вызов подпрограмм. Использование сервисов BIOS.

3. Инструментальные средства разработки и отладки программ в ОС Linux. Инструментарий GNU.

IX API: основные компоненты и версии. Получение информации о системе. Основные утилиты системного программиста. Объекты ядра ОС. Пользовательский режим и режим ядра. Основные инструменты для диагностики и мониторинга ОС. Процессы и потоки. Создание и жизненный цикл процесса. Переменные окружения процесса. Приоритет процесса. Основной поток процесса. Приоритет потока. Диспетчеризация потоков. Управление памятью. Виртуальная память и логическое адресное пространство процесса. Структура логического адресного пространства. Работа с виртуальной памятью. Работа с кучами. Работа со стеком. Файлы, отображаемые в память. Загрузка исполняемых файлов и разделяемых библиотек. Локальные файловые системы. Системы EXT2, EXT3 и EXT4. Атрибуты файлов. Создание и удаление файлов и директорий (папок в локальной файловой системе). Поиск файлов. Синхронизация потоков и доступ к ресурсам. Критические сессии. Синхронизация с использованием объектов ядра. Семафоры и мьютексы. Особенности синхронизации в многопроцессорных системах. Межпроцессная коммуникация. Сигналы. Каналы. Сокеты. Прочие механизмы и варианты их использования.

4. Семейство операционных систем Linux: базовые сервисы и использование POSIX API.

Локальные файловые системы. Системы EXT2, EXT3 и EXT4. Атрибуты файлов. Создание и удаление файлов и директорий (папок в локальной файловой системе). Поиск файлов. Синхронизация потоков и доступ к ресурсам. Критические сессии. Синхронизация с использованием объектов ядра. Семафоры и мьютексы. Особенности синхронизации в многопроцессорных системах. Межпроцессная коммуникация. Сигналы. Каналы. Сокеты. Прочие механизмы и варианты их использования.

5. Операционная система Linux: развертывание и работа с помощью интерфейса командной строки.

Цели создания и история развития операционных систем (ОС). Решаемые задачи и

требования к ним. Классификация ОС. ОС семейства Unix и Linux. Загрузка, ядро системы и процессы пользователя. Управление памятью, процессами, вводом/выводом, сетевым взаимодействием, устройствами. Системные вызовы, интерфейсы программирования API/ABI, драйверы/модули ядра, динамическая компоновка программ, файловые системы, сетевые соединения. Эффективность, надёжность и безопасность ОС. ОС для настольных компьютеров и серверов: требования и особенности. ОС реального времени. Особенности блокировки ресурсов, журналирования и восстановления после сбоев. Управление пользователями, аутентификация, авторизация, квотирование. ОС и сеть Интернет. Распределённые ОС.

## 6. Операционные системы: основы разработки и история развития.

Эффективность, надёжность и безопасность ОС. ОС для настольных компьютеров и серверов: требования и особенности. ОС реального времени. Особенности блокировки ресурсов, журналирования и восстановления после сбоев. Управление пользователями, аутентификация, авторизация, квотирование. ОС и сеть Интернет. Распределённые ОС.

## 7. Семейство операционных систем Linux: архитектура и программирование на уровне ядра ОС.

История развития ОС семейства Linux. Разрядность (32, 64 бита) и архитектуры процессоров. POSIX API: основные компоненты и версии. Получение информации о системе. Основные утилиты системного программиста. Объекты ядра ОС. Пользовательский режим и режим ядра. Основные инструменты для диагностики и мониторинга ОС. Процессы и потоки. Создание и жизненный цикл процесса. Переменные окружения процесса. Приоритет процесса. Основной поток процесса. Приоритет потока. Диспетчеризация потоков. Управление памятью. Виртуальная память и логическое адресное пространство процесса.

Структура логического адресного пространства. Работа с виртуальной памятью. Работа с кучами. Работа со стеком. Файлы, отображаемые в память. Загрузка исполняемых файлов и разделяемых библиотек. Локальные файловые системы. Системы EXT2, EXT3 и EXT4. Атрибуты файлов. Создание и удаление файлов и директорий (папок в локальной файловой системе). Поиск файлов. Синхронизация потоков и доступ к ресурсам. Критические сессии. Синхронизация с использованием объектов ядра. Семафоры и мьютексы. Особенности синхронизации в многопроцессорных системах. Межпроцессная коммуникация. Сигналы. Каналы. Сокеты. Прочие механизмы и варианты их использования.

## 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, мультимедийным оборудованием.

## 6. Перечень рекомендуемой литературы

### Основная литература

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)

Архитектура компьютера [учеб. пособие для вузов] / Э. Таненбаум, Т. Остин. — Санкт-Петербург, Питер, 2020. — URL: <https://ibooks.ru/bookshelf/361850/reading> (дата обращения: 26.11.2020). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)

### Дополнительная литература

Операционная система UNIX [учебное пособие для вузов] / А. М. Робачевский, С. А. Немнюгин, О. Л. Стесик. — Санкт-Петербург, БХВ-Петербург, 2015. — URL: <https://ibooks.ru/bookshelf/18515/reading> (дата обращения: 26.11.2020). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)

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

Не используются

**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.

На лекционных занятиях используются мультимедийные технологии, включая демонстрацию презентаций.

Для контроля и коррекции знаний обучающиеся могут использовать компьютерное тестирование.

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

В программе курса приведено минимально необходимое время для работы студента над темой. Успешное освоение дисциплины требует:

- посещения студентом всех видов аудиторных занятий;
- ведения конспекта в ходе лекционных занятий;
- качественной самостоятельной подготовки к практическим занятиям, активной работы на них;
- активной самостоятельной и аудиторной работы студента;
- своевременной сдачи преподавателю заданий по аудиторным видам работ.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Математическое моделирование и компьютерные технологии Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
<b>курс:</b>	2
<b>квалификация:</b>	бакалавр
Семестр, формы промежуточной аттестации: 3 (осенний) - Дифференцированный зачет	
<b>Разработчик:</b>	О.Н. Ивченко, заведующий кафедрой

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
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-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
	ОПК-2.3 Знает основные требования информационной безопасности

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Архитектура компьютеров и операционные системы» обучающийся должен:

### знать:

- основы работы в UNIX-подобных системах;
- основы низкоуровневого программирования;
- основы машинного кода, языков ассемблера;
- различные пути повышения производительности программы;
- основы сетевого взаимодействия;
- основы устройства сетей.

### уметь:

- создавать многопоточные и межсетевые программы на языках Си и Ассемблер;
- работать в UNIX-подобных средах;
- создавать программы на языках Си и Ассемблер без использования высокоуровневых библиотек.

### владеть:

- навыками ведения простейших программных проектов в системах контроля версий.

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

1. Система Linux, виртуальная машина.
2. Инструменты для написания.
3. Компиляции и отладки программ.
4. Командный интерпретатор bash.
5. Написание shell-скриптов.
6. Введение в язык Си.
7. Современный диалект языка Си (стандарт 2011 года).
8. Размещение данных в памяти.
9. Выравнивание данных.
10. Структуры и объединения.
11. Указатели на функции.

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



1. Стек вызовов и вызов функций на ARM.
2. Представление вещественных чисел IEEE754.
3. Программные прерывания и системные вызовы.
4. Ассемблер x86\_64. Архитектура CISC v.s. RISC. gdb и objdump. Соглашения о вызовах x86\_64.
5. Выравнивание данных и векторные инструкции SSE/AVX.
6. Системные вызовы через int 0x80 и vdso (sysenter/syscall).
7. Файловые дескрипторы, open, read и write.
8. Системные вызовы POSIX для работы со временем: time, localtime, и пр. Проблема потокобезопасности.
9. Системные вызовы stat, access, readdir.
10. Отображение ELF файла на память; системный вызов mmap.
11. Позиционно-независимый код и dlopen/dlsym.
12. Системные вызовы fork, exec, exit.

#### Критерии оценивания

Оценка "отлично" (10 баллов) 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 баллов) 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 баллов) выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочётами.

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

Оценка "хорошо" (6 баллов) выставляется студенту, если он твёрдо 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 баллов) 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 балла) 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 балла) 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 балла) 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 балл) 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 может проводиться по итогам текущей успеваемости и сдачи заданий и других видов работ, предусмотренных программой дисциплины и (или) путем организации специального опроса, проводимого в устной и (или) письменной форме.

При проведении устного дифференцированного зачета обучающемуся предоставляется 30 минут на подготовку. Опрос обучающегося по билету не должен превышать одного астрономического часа.

Во время проведения дифференцированного зачета обучающиеся могут пользоваться программой дисциплины, а также справочной литературой, конспектами лекций или другими материалами.