

**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>	Прикладная математика и информатика
<b>профиль подготовки:</b>	Информатика
	Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
<b>курс:</b>	2
<b>квалификация:</b>	бакалавр

Семестры, формы промежуточной аттестации:

3 (осенний) - Дифференцированный зачет

4 (весенний) - Дифференцированный зачет

Аудиторных часов: 120 всего, в том числе:

лекции: 60 час.

семинары: 60 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 150 час.

Всего часов: 270, всего зач. ед.: 6

Количество контрольных работ, заданий: 3

Программу составил: О.Н. Ивченко, заведующий кафедрой

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 10.04.2023

## 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 затронуто 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.

### 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-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
	ОПК-2.3 Знает основные требования информационной безопасности
ОПК-5 Способен участвовать в проведении фундаментальных и прикладных исследований и разработок, самостоятельно осваивать новые теоретические, в том числе, математические методы исследований и работать на современной экспериментальной научно-исследовательской, измерительно-аналитической и технологической аппаратуре	ОПК-5.1 Способен решать поставленные задачи в области теоретических и экспериментальных исследований и разработок
	ОПК-5.2 Обладает способностью к освоению новых знаний на основе изучения литературы, научных статей и других источников
	ОПК-5.3 Способен к профессиональной эксплуатации современной экспериментальной научно-исследовательской (измерительно-аналитической и технологической) аппаратуры

### 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	Низкоуровневые конструкции языка Си	10	10		38
2	Архитектура 32 и 64-разрядных систем ARM и x86_64	20	20		37
3	Системные вызовы и низкоуровневые функции операционных систем	18	18		38
4	Низкоуровневое сетевое программирование	12	12		37
Итого часов		60	60		150
Подготовка к экзамену		0 час.			
Общая трудоёмкость		270 час., 6 зач.ед.			

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

Семестр: 3 (Осенний)

###### 1. Низкоуровневые конструкции языка Си

Введение в язык Си. Современный диалект языка Си (стандарт 2011 года). Отличия от C++, размещение данных в памяти, выравнивание данных, структуры и объединения, указатели на функции. Представление целых чисел. Обратный дополнительный код, битовые операции. Знаковые и беззнаковые числа. Undefined Behaviour.

###### 2. Архитектура 32 и 64-разрядных систем ARM и x86\_64

Язык ассемблера ARM, базовые инструкции. Стек вызовов и вызов функций на ARM. Представление вещественных чисел IEEE754. Программные прерывания и системные вызовы. Ассемблер x86\_64. Архитектура CISC v.s. RISC. gdb и objdump. Соглашения о вызовах x86\_64. Выравнивание данных и векторные инструкции SSE/AVX.

Семестр: 4 (Весенний)

###### 3. Системные вызовы и низкоуровневые функции операционных систем

Файловые дескрипторы, open, read и write. Системные вызовы POSIX для работы со временем: time, localtime, и пр. Проблема потокобезопасности. Системные вызовы stat, access, readdir. Отображение ELF файла на память; системный вызов mmap. Позиционно-независимый код и dlopen/dlsym. Системные вызовы fork, exec, exit, pipe, mkfifo, dup2 и межпроцессное взаимодействие. mmap и POSIX shm в качестве межпроцессного взаимодействия. Сигналы BSD и UNIX System V. Файловые дескрипторы signalfd и timerfd; механизм epoll. Posix Threads, мьютексы, семафоры и atomic. Условные переменные.

#### 4. Низкоуровневое сетевое программирование

Сокеты UNIX в качестве межпроцессного взаимодействия. Сокеты TCP/IP. Сетевое взаимодействие. Прикладной уровень OSI. Протокол HTTP/1.1. Механизм epoll/kqueue для обработки TCP/IP. Сообщения UDP. Представительский уровень OSI. Шифрование с использованием Open/LibreSSL.

### 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. 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)
2. 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

1. 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.

### 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.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Прикладная математика и информатика
<b>профиль подготовки:</b>	Информатика Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
<b>курс:</b>	2
<b>квалификация:</b>	бакалавр

Семестры, формы промежуточной аттестации:

- 3 (осенний) - Дифференцированный зачет
- 4 (весенний) - Дифференцированный зачет

**Разработчик:** О.Н. Ивченко, заведующий кафедрой

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
	ОПК-2.3 Знает основные требования информационной безопасности
ОПК-5 Способен участвовать в проведении фундаментальных и прикладных исследований и разработок, самостоятельно осваивать новые теоретические, в том числе, математические методы исследований и работать на современной экспериментальной научно-исследовательской, измерительно-аналитической и технологической аппаратуре	ОПК-5.1 Способен решать поставленные задачи в области теоретических и экспериментальных исследований и разработок
	ОПК-5.2 Обладает способностью к освоению новых знаний на основе изучения литературы, научных статей и других источников
	ОПК-5.3 Способен к профессиональной эксплуатации современной экспериментальной научно-исследовательской (измерительно-аналитической и технологической) аппаратуры

## 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.

1. Представление целых чисел.
2. Обратный дополнительный код, битовые операции.
3. Знаковые и беззнаковые числа.
4. Undefined Behaviour.
5. Язык ассемблера AVR, базовые конструкции.
6. Работа с регистрами и с памятью.
7. Битовые операции на языке Си и ассемблере AVR.
8. Кодирование команд.
9. Представление целых чисел.
10. Знаковые и беззнаковые числа.
11. Флаги переноса, длинная арифметика.
12. Стек вызовов и прерывания.
13. Язык ассемблера ARM, базовые инструкции.

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

3 семестр:

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

4 семестр:

1. pipe, mkfifo, dup2 и межпроцессное взаимодействие
2. mmap и POSIX shm в качестве межпроцессного взаимодействия
3. Сигналы BSD и UNIX System V
4. Файловые дескрипторы signalfd и timerfd; механизм epoll
5. Posix Threads, мьютексы, семафоры и atomic
6. Условные переменные
7. Сокеты UNIX в качестве межпроцессного взаимодействия
8. Сокеты TCP/IP. Сетевое взаимодействие
9. Прикладной уровень OSI. Протокол HTTP/1.1
10. Механизм epoll/kqueue для обработки TCP/IP
11. Сообщения UDP
12. Представительский уровень OSI. Шифрование с использованием Open/LibreSSL.

#### **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**

При проведении дифференцированного зачёта обучающемуся предоставляется 30 минут на подготовку.

Во время проведения дифференцированного зачета обучающиеся могут пользоваться программой дисциплины, а также входящей в unix-систему справочной документацией.