

**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>	Прикладная математика, компьютерные науки и инженерия 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: 0 час.

лабораторные занятия: 60 час.

Самостоятельная работа: 150 час.

Всего часов: 270, всего зач. ед.: 6

Количество контрольных работ, заданий: 3

Программу составил: В.В. Яковлев, канд. физ.-мат. наук, заведующий кафедрой

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 04.06.2020

## 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
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
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-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

### 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. Компьютерные сети: принципы, технологии, протоколы [Текст] / В. Г. Олифер, Н. А. Олифер - СПб. Питер, 2016
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. Операционная система UNIX [Текст] : учеб. пособие для вузов / А. М. Робачевский .— СПб. : БХВ-Петербург, 2000, 2002, 2003 .— 656 с.

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

#### Дополнительная литература:

Современные операционные системы [Текст] / Э. Таненбаум, Х. Бос ; [пер. с англ.] - СПб. Питер, 2016

### 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 письменных опросов.

Зачет выставляется на основе работы на семинаре и выполнения домашних работ, либо, в случае пересдачи комиссии, выполнения задания и его защиты комиссии.

Оценка за зачет выставляется из соотношения: 30% за теоретическую часть, и 70% - за практическую.

Оценка за теоретическую часть - это среднее арифметическое, полученное из оценок на письменных контрольных работах, по итогам каждого из разделов дисциплины.

Оценка за практическую часть - это оценка за выполнение семинарских и домашних заданий, с учетом сроков сдачи.

Внимание: неудовлетворительная оценка за каждую из частей является БЛОКИРУЮЩЕЙ, то есть, в случае неудовлетворительной оценки за теоретическую либо практическую часть, итоговая оценка - неудовлетворительно.

Внимание: выполнение и сдача задач, разбираемых на семинарских занятиях, и задач домашнего задания, помеченных как “обязательные” (как правило, по одной задаче в неделю) является обязательным условием получения положительной оценки.

Все промежуточные расчеты оценки выполняются с точностью до второго знака после точки, итоговая оценка выставляется по 10-балльной шкале, с округлением по стандартным арифметическим правилам.

#### Дополнительная литература

Брайант Р.Э., О’Халларон Д.Р. Компьютерные системы: архитектура и программирование. СПб.: БХВ, 2005. 1186 с.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Прикладная математика и информатика
<b>профиль подготовки:</b>	Прикладная математика, компьютерные науки и инженерия 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:** В.В. Яковлев, канд. физ.-мат. наук, заведующий кафедрой

## 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
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-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

## 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. Реализуйте на языке ассемблера x86 (IA-32) или x86-64 функцию с сигнатурой:

```
extern double my_sin(double x)
```

которая вычисляет значение  $\sin(x)$ .

Запрещено использовать встроенные тригонометрические инструкции.

Для вычислений используйте известный вам из курса Математического анализа способ разложения функции в ряд. Точность результата должна быть максимально возможной для типа данных double.

2. Программе в аргументе командной строки передается имя файла с бинарными данными в Little-Endian.

Файл хранит внутри себя односвязный список элементов:

```
struct Item {
```

```
int value;
uint32_t next_pointer;
};
```

Поле value хранит значение элемента списка, поле next\_pointer - позицию в файле (в байтах), указывающую на следующий элемент. Признаком последнего элемента является значение next\_pointer, равное 0.

Расположение первого элемента списка (если он существует) - строго в нулевой позиции в файле, расположение остальных - случайным образом.

Выведите на экран значения элементов в списке в текстовом представлении.

Используйте отображение содержимого файла на память.

3. Программе задается единственный аргумент - номер TCP-порта.

Необходимо принимать входящие соединения на TCP/IPv4 для сервера localhost, читать данные от клиентов в текстовом виде, и отправлять их обратно в текстовом виде, заменяя все строчные буквы на заглавные. Все обрабатываемые символы - из кодировки ASCII.

Одновременных подключений может быть много. Использовать несколько потоков или процессов запрещено.

Сервер должен корректно завершать работу при получении сигнала SIGTERM.

Указание: используйте неблокирующий ввод-вывод.

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

1. Система Linux, виртуальная машина. Инструменты для написания, компиляции и отладки программ.
2. Командный интерпретатор bash, написание shell-скриптов. Введение в язык Си.
3. Современный диалект языка Си (стандарт 2011 года). Отличия от C++, размещение данных в памяти, выравнивание данных, структуры и объединения, указатели на функции.
4. Представление целых чисел. Обратный дополнительный код, битовые операции. Знаковые и беззнаковые числа. Undefined Behaviour.
5. Язык ассемблера AVR, базовые конструкции; работа с регистрами и с памятью
6. Битовые операции на языке Си и ассемблере AVR; кодирование команд
7. Представление целых чисел, знаковые и беззнаковые числа, флаги переноса, длинная арифметика
8. Стек вызовов и прерывания
9. Язык ассемблера ARM, базовые инструкции.
10. Стек вызовов и вызов функций на ARM
11. Представление вещественных чисел IEEE754
12. Программные прерывания и системные вызовы
13. Ассемблер x86\_64. Архитектура CISC v.s. RISC. gdb и objdump. Соглашения о вызовах x86\_64
14. Выравнивание данных и векторные инструкции SSE/AVX
15. Системные вызовы через int 0x80 и vdso (sysenter/syscall)
16. Файловые дескрипторы, open, read и write.
17. Системные вызовы POSIX для работы со временем: time, localtime, и пр. Проблема потокобезопасности.
18. Системные вызовы stat, access, readdir.
19. Отображение ELF файла на память; системный вызов mmap
20. Позиционно-независимый код и dlopen/dlsym
21. Системные вызовы fork, exec, exit
22. pipe, mkfifo, dup2 и межпроцессное взаимодействие
23. mmap и POSIX shm в качестве межпроцессного взаимодействия
24. Сигналы BSD и UNIX System V
25. Файловые дескрипторы signalfd и timerfd; механизм epoll
26. Posix Threads, мьютексы, семафоры и atomic
27. Условные переменные



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

Примеры билетов (первая часть, третий семестр):

1. Стек вызовов и вызов функций на ARM. Файловые дескрипторы, open, read и write.
2. Стек вызовов и прерывания. Отображение ELF файла на память; системный вызов mmap.
3. Выравнивание данных и векторные инструкции SSE/AVX. Позиционно-независимый код и dlopen/dlsym.

Примеры билетов (вторая часть, четвертый семестр):

1. pipe, mkfifo, dup2 и межпроцессное взаимодействие. Сообщения UDP.
2. Сокеты TCP/IP. Сетевое взаимодействие. Системные вызовы fork, exec, exit.
3. Сигналы BSD и UNIX System V. Механизм epoll/kqueue для обработки TCP/IP.

#### 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

Дифференцированный зачет выставляется на основе работы на семинаре и выполнения домашних работ, либо, в случае пересдачи комиссии, выполнения задания и его защиты комиссии.

Оценка за дифференцированный зачет выставляется из соотношения: 30% за теоретическую часть при ответе на билет, и 70% - за практическую по итогам выполненных домашних заданий.

Внимание: неудовлетворительная оценка за каждую из частей является БЛОКИРУЮЩЕЙ, то есть, в случае неудовлетворительной оценки за теоретическую либо практическую часть, итоговая оценка - неудовлетворительно.