

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

УТВЕРЖДЕНО

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

	Рабочая программа дисциплины (модуля)
по дисциплине:	Разработка операционных систем
по направлению:	Программная инженерия
профиль подготовки:	Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии МФТИ - Яндекс
курс:	2
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 4 (весенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 0 час.

лабораторные занятия: 30 час.

Самостоятельная работа: 48 час.

Всего часов: 108, всего зач. ед.: 3

Количество контрольных работ, заданий: 2

Программу составил: А.В. Малеев, заведующий кафедрой

Программа обсуждена на заседании высшей школы программной инженерии МФТИ - Яндекс 07.06.2022

Аннотация

Данная дисциплина была разработана для обучения студентов базовым аспектам разработки программного обеспечения для UNIX-подобных операционных систем, а также отработки навыков написания программ и их тестирования в предельных ситуациях.

1. Цели и задачи

Цель дисциплины

- обучение студентов базовым аспектам разработки программного обеспечения для UNIX-подобных операционных систем, а также отработки навыков написания программ и их тестирования в предельных ситуациях.

Задачи дисциплины

- демонстрация 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-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1 Формулирует совокупность взаимосвязанных задач в рамках поставленной цели работы, обеспечивающих ее достижение. Определяет ожидаемые результаты решения поставленных задач
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-4 Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	ОПК-4.1 Знает основные правила оформления технической документации на различных стадиях жизненного цикла информационной системы
	ОПК-4.2 Владеет на практике методологией составления научно-технических отчетов (проектов)
ОПК-8 Способен осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий	ОПК-8.3 Умеет оптимизировать и проводить рефакторинг существующего кода для улучшения производительности и поддержки
ПК-5 Способен проектировать, разрабатывать, внедрять, сопровождать и снимать с эксплуатации информационные системы	ПК-5.1 Умеет описывать архитектуру, устройство и функционирование информационных систем
	ПК-5.2 Умеет определять оптимальные методы и инструменты разработки, внедрения и адаптации прикладного программного обеспечения информационных систем
	ПК-5.3 Умеет разрабатывать прототип информационных систем в соответствии с требованиями и проводить его тестирование для проверки корректности архитектурных решений
	ПК-5.4 Имеет практический опыт разработки, внедрения и адаптации прикладного программного обеспечения информационных систем в различных программных средах

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	16
2	Системные вызовы и низкоуровневые функции операционных систем	10		10	16
3	Низкоуровневое сетевое программирование	10		10	16
Итого часов		30		30	48
Подготовка к экзамену		0 час.			
Общая трудоёмкость		108 час., 3 зач.ед.			

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

Семестр: 4 (Весенний)

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. 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.

3. 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)

компьютер, мультимедийное оборудование (проектор).

6. Перечень рекомендуемой литературы

Основная литература

1. Защищенный код для Windows Vista, Электрон. версия печ. публикации / М. Ховард, Д. Лебланк. — Москва, Русская редакция, 2008
2. Основы операционных систем, [учебное пособие] / В. Е. Карпов, К. А. Коньков. — Москва, ИНТУИТ, 2016.— URL: <https://e.lanbook.com/book/100311> (дата обращения: 26.01.2021). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)

Дополнительная литература

Список литературы для самостоятельного изучения:

Архитектура компьютера [Текст] : [учеб. пособие для вузов] / Э. Таненбаум, Т. Остин ; [пер. с англ. Е. Матвеев] .— 6-е изд. — СПб. : Питер, 2014 .— 816 с
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)

Отсутствует

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

На лекциях используется компьютер и мультимедийное оборудование (проектор, звуковая система)

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

Программа курса в разделе «самостоятельная работа» обозначает 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, доказательство отдельных утверждений, свойств,
- выполнение домашних заданий.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Программная инженерия
профиль подготовки:	Разработка программно-информационных систем высшая школа программной инженерии МФТИ - Яндекс высшая школа программной инженерии
курс:	2
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 4 (весенний) - Дифференцированный зачет

Разработчик: А.В. Малеев, заведующий кафедрой

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1 Формулирует совокупность взаимосвязанных задач в рамках поставленной цели работы, обеспечивающих ее достижение. Определяет ожидаемые результаты решения поставленных задач
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-4 Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	ОПК-4.1 Знает основные правила оформления технической документации на различных стадиях жизненного цикла информационной системы
	ОПК-4.2 Владеет на практике методологией составления научно-технических отчетов (проектов)
ОПК-8 Способен осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий	ОПК-8.3 Умеет оптимизировать и проводить рефакторинг существующего кода для улучшения производительности и поддержки
ПК-5 Способен проектировать, разрабатывать, внедрять, сопровождать и снимать с эксплуатации информационные системы	ПК-5.1 Умеет описывать архитектуру, устройство и функционирование информационных систем
	ПК-5.2 Умеет определять оптимальные методы и инструменты разработки, внедрения и адаптации прикладного программного обеспечения информационных систем
	ПК-5.3 Умеет разрабатывать прототип информационных систем в соответствии с требованиями и проводить его тестирование для проверки корректности архитектурных решений
	ПК-5.4 Имеет практический опыт разработки, внедрения и адаптации прикладного программного обеспечения информационных систем в различных программных средах

2. Показатели оценивания компетенций

В результате изучения дисциплины «Разработка операционных систем» 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. Основные этапы развития и классификация программного обеспечения ЭВМ. Структура и функции системного, инструментального и прикладного программного обеспечения.
2. Эволюция, назначение и типы операционных систем.
3. Структура ядра операционной системы и его функции. Утилиты, системные обрабатывающие программы и библиотеки.
4. Функциональные компоненты операционной системы автономного компьютера. Организация взаимодействия прикладных программ с операционной системой через функции API.
5. Виды архитектур сетевых операционных систем. Коммуникационные протоколы. Сетевые службы и сетевые сервисы.
6. Средства аппаратной поддержки операционных систем.
7. Сущность концепции микроядерной архитектуры, ее достоинства и недостатки. Макроядерные операционные системы.
8. Реализация прикладных программных сред. Стандартизация системных функций и процедур. Стандарты POSIX.
9. Классификация ресурсов вычислительной системы и возможности их разделения. Понятие вычислительного процесса.
10. Мультипрограммный режим обработки данных. Критерии эффективности функционирования вычислительных систем.
11. Понятие потока и его отличие от понятия процесса. Граф состояний вычислительного процесса в многозадачной среде.
12. Характеристика основных стратегий планирования и диспетчеризации процессов в мультипрограммных системах.
13. Дисциплины планирования, основанные на квантовании. Диаграмма состояний потока в системах с квантованием времени.
14. Дисциплины планирования, основанные на приоритетах. Абсолютные, относительные и динамические приоритеты.
15. Мультипрограммная обработка данных на основе прерываний. Внешние, внутренние и программные прерывания.

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

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.

Критерии оценивания

- оценка «отлично (10)» 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 (9)» выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений
- оценка «отлично (8)» выставляется студенту, показавшему всесторонние систематизированные, глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, и правильное обоснование принятых решений
- оценка «хорошо (7)» 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 (6)» выставляется студенту, если он 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 знает материал, и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;
- 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 владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;
- оценка «удовлетворительно (3)» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет фрагментарно основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;
- оценка «неудовлетворительно (2)» выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач
- оценка «неудовлетворительно (1)» выставляется студенту, который не знает формулировок основных понятий дисциплины

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

Дифференцированный зачет проводится 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.