

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

УТВЕРЖДЕНО

**Директор института-заместитель
директора ФАКТ**

М.А. Кудров

	Рабочая программа дисциплины (модуля)
по дисциплине:	Архитектура компьютеров
по направлению:	Информатика и вычислительная техника
профиль подготовки:	Программная инженерия передовая инженерная школа радиолокации, радионавигации и программной инженерии кафедра технологий проектирования сложных технических систем
курс:	1
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 2 (весенний) - Дифференцированный зачет

Аудиторных часов: 75 всего, в том числе:

лекции: 30 час.

семинары: 45 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 60 час.

Всего часов: 135, всего зач. ед.: 3

Количество контрольных работ, заданий: 2

Программу составил: В.К. Хохлов, канд. техн. наук, доцент

Программа обсуждена на заседании кафедры технологий проектирования сложных технических систем
07.02.2024

Аннотация

Целью курса является знакомство обучающихся с архитектурой компьютеров и операционными системами. Студенты научатся работать в графическом интерфейсе Unix, в командном интерпретаторе, приобретут опыт взаимодействия с системными вызовами из программ на Си. В рамках курса будет рассмотрено написание shell-скриптов, программы sed и awk, общие концепции: процесс и планирование, кооперация процессов, управление памятью и файловые системы.

1. Цели и задачи

Цель дисциплины

- получение базовых знаний об организации компьютеров и операционных систем, разделении обязанностей между аппаратным обеспечением и ядром операционной системы. Рассмотрение концепций современных операционных систем производится на примере операционной системы Unix. Рассматриваются пользовательский интерфейс Unix, программирование на языке Unix Shell, использование системных вызовов для взаимодействия с ядром в программах на языке Си.

Задачи дисциплины

- изучение основных концепций и принципов проектирования операционных систем. Рассмотрение взаимодействия ядра операционной системы с аппаратным обеспечением современных компьютеров;
- рассмотрение реализации основных концепций современных ОС на примере Unix (понятия процесс, планировщик процессов файл и др.);
- знакомство с командной оболочкой Unix Shell на уровне пользователя и программиста. Выполнение лабораторных работ по написанию Shell-скриптов. Выполнение лабораторных работ на других скриптовых языках, в том числе, sed и AWK;
- изучение основных системных вызовов Unix. Программирование на языке Си с использованием системных вызовов.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели
	ПК-1.3 Способен применять теоретические и (или) экспериментальные методы исследований к конкретной научной задаче и интерпретировать полученные результаты
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации
	ПК-2.2 Способен планировать и проводить научные исследования самостоятельно или в качестве члена (руководителя) малого научного коллектива
	ПК-2.3 Способен проводить апробацию результатов научно-исследовательской работы посредством публикации научных статей и участия в конференциях

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

- основные компоненты ОС общего назначения, необходимые для её функционирования;
- основные команды, необходимые для уверенной работы в Unix Shell на уровне пользователя;
- управляющие операторы и управляющие конструкции Unix Shell, необходимые для написания shell-скриптов;

уметь:

- работать в командной оболочке Unix Shell, писать скрипты для Unix Shell, писать программы на языке Си с использованием системных вызовов ОС Unix.

владеть:

- приёмами программирования на скриптовых языках на примере Unix Shell, awk и sed.

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	Введение. Структура ЭВМ	4	7		10
2	Представление информации в памяти компьютера	4	7		10
3	Архитектура процессора	4	7		10
4	Иерархия памяти	4	8		10
5	Машинное представление программ	8	8		10
6	Оптимизация программ	6	8		10
Итого часов		30	45		60
Подготовка к экзамену		0 час.			
Общая трудоёмкость		135 час., 3 зач.ед.			

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

Семестр: 2 (Весенний)

1. Введение. Структура ЭВМ

Уровни абстрактного представления ЭВМ, язык Ассемблера и машинные команды среди них. Элементы и контекст машинного представления информации. Трансляция и интерпретация программ и команд. Краткое описание устройств ЭВМ и схема их взаимодействия. Структура центрального процессора (ЦП). Регистры, арифметико-логическое устройство, устройство управления. Схема работы ЭВМ. Кэширование и иерархия устройств хранения. Оперативная память ЭВМ. Ячейки, адреса, машинные слова, разряды, биты. Двоичное представление информации в ЭВМ, причины выбора такого представления. Взаимодействие ЭВМ друг с другом. Одновременность и параллельность.

2. Представление информации в памяти компьютера

Двоичная система счисления. Шестнадцатеричная нотация. Слова и размеры данных. Представления целых чисел в форме с фиксированной точкой (представление беззнаковых чисел, представление знаковых чисел в прямом и дополнительном кодах). Особенности сложения и вычитания целых чисел. Флаги. Представление вещественных чисел в форме с плавающей точкой. Размещение числовых данных в памяти. Двоично-десятичные числа. Представление нечисловой информации.

3. Архитектура процессора

Архитектура системы команд X86. Способы задания операндов. Система команд как важнейшая характеристика ЭВМ. Разнообразие систем команд в реальных ЭВМ (CISC, RISC и др.). Понятие цифрового конструирования и язык управления аппаратурой. Последовательная реализация X86. Основные принципы конвейеризации. Конвейерная реализация X86.

4. Иерархия памяти

Технологии хранения данных. Локальность. Иерархия видов памяти и принцип кэширования. Кэш-память. Создание кэш-ориентированных программ. Влияние кэш-памяти на производительность.

5. Машинное представление программ

Кодирование программ. Форматы данных. Обращение к данным. Арифметические и битовые операции. Команды управления. Процедуры. Массивы. Неоднородные конструкции данных. Указатели. Использование отладчика. Некорректные ссылки и переполнение буфера. 64-битное расширение IA-32. Программы с плавающей точкой.

6. Оптимизация программ

Возможности и ограничения оптимизирующих компиляторов. Измерение производительности программ. Исключение неэффективности циклов. Уменьшение количества вызовов процедур. Исключение ненужных ссылок в память. Понятие о современном процессоре. Разворачивание циклов. Увеличение степени параллелизма. Результат оптимизации кода. Ограничители производительности. Производительность памяти. Обнаружение и исключение мест потери производительности.

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

- персональные компьютеры;
- учебная аудитория;
- мультимедийный проектор;
- экран.

6. Перечень рекомендуемой литературы

Основная литература

1. Основы операционных систем [Текст] : Курс лекций : учеб. пособие для вузов / В. Е. Карпов, К. А. Коньков .— 2-е изд., доп. и испр. — М. : Интернет - Ун-т информац. технологий, 2009, 2011 .— 536 с.
2. Операционная система UNIX [Текст] : учеб. пособие для вузов / А. М. Робачевский, С. А. Немнюгин, О. Л. Стесик .— 2-е изд., перераб. и доп. — СПб. : БХВ-Петербург, 2005, 2007, 2010 .— 656 с.

Дополнительная литература

1. Программирование на языке ассемблера IBM PC [Текст] : [учеб. пособие для вузов] / В. Н. Пильщиков .— М. : ДИАЛОГ-МИФИ, 2005, 2010 .— 288 с.

2. Архитектура IBM PC и язык Ассемблера [Текст] : учеб. пособие для вузов / В. Я. Митницкий ; М-во образования Рос. Федерации, Моск. физ.-техн. ин-т (гос. ун-т) .— М. : МФТИ, 2000 .— 148 с.
3. Современные операционные системы [Текст] : [учеб. пособие для вузов] / Э. Таненбаум ; [пер. с англ. Н. Вильчинский, А. Лашкевич] .— 3-е изд. — СПб. : Питер, 2015 .— 1120 с

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

1. <http://www.citforum.ru/>
2. <http://www.opennet.ru/man.shtml> (оригиналы и переводы справочных руководств Unix).
3. <https://www.freebsd.org/cgi/man.cgi> (оригинальные MAN-страницы ОС FreeBSD)

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

удалённый Unix-сервер с помощью подключения по протоколу SSH. Dev-C++
—интегрированная среда разработки приложений.

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

- чтение и конспектирование рекомендованной литературы;
- проработку учебного материала (по конспектам лекций, учебной и научной литературе), подготовку ответов на вопросы, предназначенных для самостоятельного изучения;
- доказательство отдельных утверждений, свойств;
- решение задач, предлагаемых студентам на лекциях и практических занятиях;
- подготовку к практическим занятиям, коллоквиумам, дифференцированному зачету.

ПРИЛОЖЕНИЕ

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Информатика и вычислительная техника
профиль подготовки:	Программная инженерия передовая инженерная школа радиолокации, радионавигации и программной инженерии кафедра технологий проектирования сложных технических систем
курс:	1
квалификация:	бакалавр
Семестр, формы промежуточной аттестации: 2 (весенний) - Дифференцированный зачет	
Разработчик:	В.К. Хохлов, канд. техн. наук, доцент

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели
	ПК-1.3 Способен применять теоретические и (или) экспериментальные методы исследований к конкретной научной задаче и интерпретировать полученные результаты
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации
	ПК-2.2 Способен планировать и проводить научные исследования самостоятельно или в качестве члена (руководителя) малого научного коллектива
	ПК-2.3 Способен проводить апробацию результатов научно-исследовательской работы посредством публикации научных статей и участия в конференциях

2. Показатели оценивания компетенций

В результате изучения дисциплины «Архитектура компьютеров» обучающийся должен:

знать:

- основные компоненты ОС общего назначения, необходимые для её функционирования;
- основные команды, необходимые для уверенной работы в Unix Shell на уровне пользователя;
- управляющие операторы и управляющие конструкции Unix Shell, необходимые для написания shell-скриптов;

уметь:

- работать в командной оболочке Unix Shell, писать скрипты для Unix Shell, писать программы на языке Си с использованием системных вызовов ОС Unix.

владеть:

- приёмами программирования на скриптовых языках на примере Unix Shell, awk и sed.

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

Перечень вопросов к дифференцированному зачету:

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
13. pipe, mkfifo, dup2 и межпроцессное взаимодействие
14. mmap и POSIX shm в качестве межпроцессного взаимодействия
15. Сигналы BSD и UNIX System V
16. Файловые дескрипторы signalfd и timerfd; механизм epoll
17. Posix Threads, мьютексы, семафоры и atomic
18. Условные переменные
19. Сокеты UNIX качестве межпроцессного взаимодействия
20. Сокеты TCP/IP. Сетевое взаимодействие
21. Прикладной уровень OSI. Протокол HTTP/1.1
22. Механизм epoll/kqueue для обработки TCP/IP
23. Сообщения UDP
24. Представительский уровень OSI. Шифрование с использованием Open/LibreSSL.

Пример билета:

Билет №1

- 1.Стек вызовов и вызов функций на ARM
- 2.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.