



Национальный исследовательский университет «Высшая школа экономики»
Программа дисциплины «Архитектура компьютера и операционные системы» для направления
01.03.02 «Прикладная математика и информатика» подготовки бакалавра

**Федеральное государственное автономное образовательное учреждение
высшего образования
"Национальный исследовательский университет
"Высшая школа экономики"**

Факультет компьютерных наук
Департамент больших данных и информационного поиска

**Рабочая программа дисциплины
Архитектура компьютера и операционные системы**

для образовательной программы «Прикладная математика и информатика» направления
подготовки 01.03.02 Прикладная математика и информатика
уровень - бакалавр

Разработчик программы:
А.В. Чернов-доцент, к.ф.-м.н

Одобрена на заседании Департамента больших данных и информационного поиска
«__»_____2018г.

Руководитель Департамента больших данных и информационного поиска
В.В.Подольский _____

Утверждена Академическим советом образовательной программы
«__»_____2018 г., № протокола_____

Академический руководитель образовательной программы
А.С.Конущин _____

Москва, 2018

*Настоящая программа не может быть использована другими подразделениями университета и
другими вузами без разрешения подразделения-разработчика программы.*



1. Область применения и нормативные ссылки

Настоящая программа учебной дисциплины устанавливает минимальные требования к знаниям и умениям студента и определяет содержание и виды учебных занятий и отчетности. Программа предназначена для преподавателей, ведущих дисциплину «Архитектура компьютера и операционные системы», учебных ассистентов и студентов направления подготовки 01.03.02 «Прикладная математика и информатика», обучающихся по образовательной программе «Прикладная математика и информатика»

Программа учебной дисциплины разработана в соответствии с:

- Образовательным стандартом ФГАОУ ВО НИУ ВШЭ по направлению подготовки 01.03.02 Прикладная математика и информатика <https://www.hse.ru/data/2017/09/04/1321436546/2.03.02%20%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%B0%D1%8F%20%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0%20%D0%B8%20%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0.pdf> ;
- Образовательной программой «Прикладная математика и информатика», направление подготовки 01.03.02 «Прикладная математика и информатика»
- Объединенным учебным планом университета по образовательной программе «Прикладная математика и информатика», утверждённым в 2018 г.

2. Цели освоения дисциплины

2.1 Аннотация

Дисциплина «Архитектура и операционные системы» предназначена для подготовки бакалавров по направлению 01.03.02 – Прикладная математика и информатика.

Понимание устройства и принципов работы электронных вычислительных машин (ЭВМ), которые далее мы будем называть уже привычным термином «компьютер», а также основных принципов работы операционных систем является незаменимым для инженеров в сфере информационных технологий при:

- оценке проектных решений, особенно комплексных и затрагивающих различные уровни абстракции, например, выборе систем виртуализации и платформ развёртывания;
- проектировании новых систем для достижения компромисса стоимость-производительность-надежность;
- выявлении узких мест в работе существующих информационных систем;
- диагностики нетривиальных неполадок и ошибок;
- повышении производительности вычислений за счет более эффективного использования программных и аппаратных средств;
- построении инновационных программно-аппаратных систем: систем интеллектуального анализа данных, роботов, инструментов телеприсутствия и дополненной реальности.

2.2. Цели освоения

Цель дисциплины – помочь развитию незаменимого в профессиональной деятельности «инженерного чутья», которое рождается из понимания принципов работы компьютеров и деталей взаимодействия прикладного и системного программного обеспечения (ПО) с аппаратурой ЭВМ. Очевидно, что кроме теоретических оснований, дисциплина предусматривает изучение конкретных реализаций аппаратного обеспечения, операционных систем и т.п. Конкретно, будут рассмотрены следующие разделы.

1. Кодирование информации и форматы данных. Представление целочисленных, вещественных и символьных данных в компьютере.
2. Аппаратура современных компьютеров. Интегральные схемы.
3. Уровни абстракции – от микроархитектуры ядра центрального процессора до распределённых гетерогенных информационных систем. Роль базовых систем



ввода/вывода и операционных систем.

4. Наиболее популярные архитектуры процессоров общего назначения. Подробное рассмотрение архитектуры *x86* (от *8086/8088* до *AMD64/EM64T*).
5. Программирование на языке ассемблера для архитектуры IA32.
6. Базовые проектные решения при создании операционных систем для достижения высокой производительности, надёжности, информационной безопасности.
7. Инструментарий операционной системы Linux для системного программиста.
8. Основные сервисы операционных систем семейств *Linux* и их правильное использование.

2.3. Учебные задачи дисциплины

Дисциплина должна сформировать представление о проблемах разработки ПО на разных уровнях абстракции при оптимальном использовании возможностей аппаратного обеспечения. В результате изучения дисциплины «Архитектура компьютера и операционные системы» студенты должны:

- 1) знать наиболее важные проектные решения на всех уровнях абстракции, накопленные в процессе развития компьютерной техники, и основные параметры архитектуры *x86*, строение операционных систем общего назначения и базовые сервисы операционных систем семейств *Linux*;
- 2) понимать связь свойств аппаратного обеспечения современных цифровых компьютеров с проблемами системного программирования;
- 3) уметь применять язык ассемблера для платформ *x86* при создании системного ПО и использовать базовые сервисы операционных систем семейств *Linux*.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины

Дисциплина формирует следующие компетенции.

Компетенция	Код по ОС ВШЭ	Уровень формирования компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции	Форма контроля уровня сформированности компетенции
Способен вести исследовательскую деятельность, включая анализ проблем, постановку целей и задач, выделение объекта и предмета исследования, выбор способа и методов исследования, а также оценку его качества	УК-6	РБ	Способность находить и самостоятельно осваивать нужную информацию из общедоступных источников; умение формулировать и объяснять доказательства теорем в устной и письменной форме; умение оптимально распределять собственное время при работе над	Лекции, практические занятия, домашние задания	Письменные контрольные работы, устный опрос, экзамен



Компетенция	Код по ОС ВШЭ	Уровень формирования компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции	Форма контроля уровня сформированности компетенции
			поставленными задачами		
Способен формализовать и алгоритмизировать поставленную задачу.	ПК-4	РБ, СД	Умение формализовать условие задачи, требующей алгоритмического решения; умение разбить задачу на подзадачи; умение проводить анализ корректности и временной сложности алгоритмов	Лекции, практические занятия, домашние задания	Письменные контрольные работы, устный опрос, экзамен
Способен писать, оформлять, отлаживать и оптимизировать программный код.	ПК-5	РБ, СД	Способность программировать, отлаживать и тестировать алгоритмы и структуры данных	Лекции, практические занятия, домашние задания	Письменные контрольные работы, устный опрос, экзамен
Способен спроектировать и реализовать законченную программную систему с использованием готовых программных модулей и компонент.	ПК-6	РБ, СД	Способность проектировать законченную программную систему; умение находить в общедоступных источниках готовые программные модули и компоненты; способность провести интеграцию программной системы с готовыми программными модулями и компонентами	Лекции, практические занятия, домашние задания	Письменные контрольные работы, устный опрос, экзамен

Виды и задачи профессиональной деятельности

Научно-исследовательские	
Исследование и разработка математических моделей и методов, алгоритмов и программного обеспечения по тематике проводимых научно-исследовательских	НИД-3



проектов;	
Проектные и производственно-технологические	
Разработка архитектуры, алгоритмических и программных решений системного и прикладного программного обеспечения	ПД-3
Разработка программного и информационного обеспечения компьютерных систем, автоматизированных систем вычислительных комплексов, сервисов, операционных систем и распределенных баз данных	ПД-4
Изучение и использование различных языков программирования, алгоритмов, библиотек и пакетов программ при разработке программного обеспечения	ПД-5

4. Место дисциплины в структуре образовательной программы

Настоящая дисциплина относится к профессиональному циклу, базовой части (Б. Пр.Б.).

Изучение дисциплины «Архитектура компьютера и операционные системы» требует предварительных знаний в следующих областях: математическая логика, дискретная математика, основы информатики, программирование на языках высокого уровня (Си или Си++).

Дисциплина является пререквизитом для следующих дисциплин: Параллельные и распределенные вычисления, Компьютерные сети и базы данных, Безопасность компьютерных систем, Введение в управление робототехническими системами, Введение в программирование на Erlang, Методы и системы обработки больших данных, Алгоритмы для иерархий памяти.

5. Тематический план учебной дисциплины

№	Название темы	Всего часов по дисциплине	Аудиторные часы		Самостоятельная работа
			Лекции	Сем. и практика	
1	Цифровые электронно-вычислительные машины. Основные концепции и история развития.	12	6	0	6
2	Аппаратура современного компьютера. Архитектура x86: история развития, аппаратная реализация и язык ассемблера.	60	12	16	32
3	Операционная система Linux: развертывание и работа с помощью интерфейса командной строки	14	2	4	8
4	Инструментальные средства разработки и отладки программ в ОС Linux. Инструментарий GNU.	14	2	4	8
5	Представление целых, вещественных и строковых данных в компьютере	40	8	12	20
6	Операционные системы: основы разработки и история развития.	16	8	0	8
7	Семейство операционных систем Linux: базовые сервисы и использование <i>POSIX API</i> .	160	36	44	80
8	Семейство операционных систем Linux: архитектура и программирование на уровне ядра ОС.	26	10	4	12



Итого	342	84	84	174
--------------	------------	-----------	-----------	------------

6. Формы контроля знаний студентов и критерии оценки знаний, навыков

Текущий контроль – 9 контрольных работ (по 80 мин.) на компьютере, домашние работы, работа на семинарах.

Итоговый контроль – 1 экзамен (в конце третьего модуля изучения курса);

Итоговая оценка за курс (максимум 10 баллов) складывается из:

- оценки за экзамен (максимум 10 баллов), умножаемой на коэффициент 0.4;
- накопленной оценки (максимум 10 баллов), умножаемой на коэффициент 0.6;
- полученная сумма округляется к ближайшему целому.

Накопленная оценка $O_{\text{накоп}}$ вычисляется по следующей формуле:

$$O_{\text{накоп}} = 0.9 \cdot O_{\text{задач}} + 0.1 \cdot O_{\text{семинар}}$$

Где $O_{\text{задач}}$ – это оценка за выполнение контрольных и домашних работ, сданных в систему автоматического тестирования, от 0 до 10 баллов, а $O_{\text{семинар}}$ – оценка, выставляемая преподавателем на основании работы на семинарских занятиях, от 0 до 10 баллов.

Оценка $O_{\text{задач}}$ за выполнение контрольных и домашних работ, сданных в систему автоматического тестирования, определяется, исходя из суммы первичных баллов, полученных за каждую сданную задачу, сравнением с пороговыми значениями.

7. Содержание дисциплины

Тема 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*, *MacOS*, *GNU*, *Minix*, *Linux*, *iOS*, *Android*.

Тема 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*, кеши, 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.



Тема 3. Операционные системы: основы разработки и история развития

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.

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

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.

8. Оценочные средства для текущего контроля и аттестации студента

8.1. Оценочные средства для оценки качества освоения дисциплины в ходе текущего контроля Темы домашних работ

1. Представление целых чисел. Порядок байт Big-Endian и Little-Endian.
2. Представление вещественных чисел. Выполнение умножения и сложения вещественных чисел.
3. Представление строковых данных. Кодировка UTF-8.
4. Простейшие программы на ассемблере.
5. Организация стековых фреймов и позиционно-независимого кода.
6. Низкоуровневая работа с файлами средствами POSIX API.
7. Работа с файловой системой средствами POSIX API.
8. Работа с виртуальной памятью. Файлы, отображаемые в память.
9. Процессы. Создание процессов, запуск файлов на исполнение.
10. Неименованные и именованные каналы. Перенаправление ввода-вывода. Конвейер.
11. Сигналы.
12. Потоки. Создание, ожидание и удаление потоков средствами POSIX Threads.
13. Мьютексы и условные переменные.



14. Классические задачи синхронизации.
15. Сокеты. Мультиплексирование ввода-вывода.

Примеры задач, предлагаемых на контрольных работах

1. Составить подпрограмму на языке Ассемблера.
2. Составить программу на языке Си с использованием POSIX API.

Вопросы для оценки качества освоения дисциплины

1. Что такое «абстракция»?
2. Как Вы понимаете значение терминов «интерфейс» и «протокол»?
3. Сколько бит содержит байт? Почему?
4. Сколько бит содержит слово? Почему?
5. Что такое «обратный код» представления целых чисел со знаком?
6. Из чего состоит логическая схема?
7. Что такое прерывание?
8. Что такое шина? Какие они бывают?
9. Какие уровни виртуализации вы знаете?
10. Как может проявляться «закон дырявых абстракций»?
11. Что такое суперскалярность?
12. Как реализуется внеочередное выполнение инструкций центральным процессором?
13. Как связана длина конвейера центрального процессора с проблемой предсказания ветвлений?
14. Приведите примеры кэширования в аппаратуре вашего компьютера.
15. Как расшифровывается аббревиатура *BIOS*?
16. Когда появилась первая операционная система?
17. Как можно охарактеризовать строение ядра ОС Linux?
18. Чем процесс отличается от потока?
19. Что такое *PID*?
20. Кому и как присваиваются приоритеты? Как их можно "посмотреть" и поменять?
21. Что такое логическое адресное пространство? Каков его размер и от чего он зависит?
22. Чем логический адрес отличается от физического?
23. Что такое «виртуальная память»?
24. Как реализуется страничная адресация?
25. Что такое «режим ядра»?
26. Что такое «окружение [*environment*]» процесса? Как классифицируются переменные окружения?
27. Как изменить переменные окружения? При каких условиях это возможно?
28. Каковы основные особенности файловой системы *EXT4*?
29. Какие символы недопустимы в имени файла в файловой системе *EXT4*?
30. Перечислите основные атрибуты файла.
31. Что такое режим доступа к файлу?
32. При каких условиях файл может быть открыт двумя разными программами?
33. Что такое «маска файла»?
34. Как *ELF*-файл загружается для выполнения?
35. Как компоновщик собирает *ELF*-файл?
36. Чем статическая компоновка отличается от динамической?
37. Как происходит импорт/экспорт функций из *SO*?
38. Какие способы межпроцессной коммуникации вы знаете?
39. Чем именованный отображенный в память файл отличается от неименованного?

8.2. Примеры заданий промежуточной аттестации



Темы вопросов на экзамене

1. Поколения ЭВМ
2. Серия IBM/360
3. PDP-11
4. Компьютерные системы. Иерархии машин.
5. Управление ресурсами. ОС как менеджер ресурсов.
6. Аппаратные средства ЦП для организации многопроцессного режима.
7. Ядро. Основные функции ядра. Системные вызовы.
8. Шина. Архитектура общей шины.
9. Архитектура с раздельными шинами.
10. Физическая организация памяти.
11. Кэширование. Характеристики кэш-памяти.
12. Страничная виртуальная адресация.
13. Многоуровневая страничная адресация.
14. Проблема внешней и внутренней фрагментации.
15. Соглашения о вызовах.
16. Реализация системных вызовов.
17. Стандартизация Unix и Linux.
18. Структура дерева каталогов Linux.
19. Основные команды shell, перенаправление потоков.
20. 24. Файловые системы. Основы организации файлов.
21. Файловая система FAT.
22. Файловая система s5fs.
23. Файловые системы ext2, ext3, ext4.
24. Журналирование ФС, квотирование ФС.
25. Основные требования к ФС.
26. Системные вызовы работы с файлами.
27. Атомарность операций ФС.
28. Блокировки файлов.
29. Реализация работы с файлами в ядре ОС.
30. Получение и использование информации о файле в ФС.
31. Работа с каталогами. Работа с ФС.
32. Специальные файлы устройств. Типы устройств.
33. Алгоритмы планирования дисковых обменов.
34. RAID-системы.
35. Адресное пространство процесса. Лимиты адресного пространства.
36. Управление адресным пространством. Стратегии распределения памяти.
37. Файлы, отображаемые в память.
38. Механизм sору-on-write.
39. Механизмы загрузки программы на выполнение. Динамические библиотеки.
40. Выделение и учет физических страниц в ядре.
41. Алгоритмы замещения страниц.
42. Страничный демон.
43. Буферный и страничный кэши.
44. Процесс в UNIX. Атрибуты процесса. Идентификация процесса.
45. Создание процесса. Завершение работы процесса.
46. Ожидание завершения процесса.
47. Жизненный цикл процесса.
48. Замещение тела процесса.
49. Схема fork/exec.
50. Неименованные каналы.
51. Именованные каналы.
52. Сигналы, базовые средства работы с сигналами



53. Особенности обработки сигналов в System V и BSD, Linux
54. Особенности написания обработчиков сигналов
55. Безопасная обработка сигналов.
56. Нити. Модели поддержки нитей.
57. Создание, завершение работы нитей. Ожидание нитей.
58. Основные понятия параллельного программирования. Требования к механизмам взаимного исключения.
59. Базовые средства взаимного исключения. Семафоры, мониторы.
60. Мьютексы и нити.
61. Условные переменные, использование условных переменных.
62. Задача об обедающих философах.
63. Задача читателей и писателей.
64. Задача производителей и потребителей.
65. Задача о спящем парикмахере.
66. Группы процессов UNIX.
67. Процессы-демоны.
68. Пакетное планирование процессов.
69. Планирование разделения времени. Реализация планирования в Linux.
70. Планирование реального времени.
71. Инверсия приоритета. Методы борьбы с инверсией приоритетов.
72. Прерывания. Обработка прерываний.
73. Классификация ядер ОС.
74. ОС реального времени.
75. Сокеты. Взаимодействие без установления соединения.
76. Взаимодействие с установлением соединения.
77. Классификация архитектур процессов-серверов.
78. Мультиплексирование событий pselect.
79. Событийно ориентированные программы.
80. Трассировка процессов ptrace.
81. Загрузка ядра Linux
82. Инициализационные скрипты, уровни работы.
83. Управление консолями, getty.

9. Порядок формирования оценок по дисциплине

Контрольные работы выполняются в аудиториях под контролем преподавателей. Задания контрольных работ сдаются в виде программ в электронной форме строго в отведенное время. Во время семинаров выполняются тренировочные задания, которые сдаются, как правило, в виде программ в электронной форме. Домашние работы состоят из: «дорешиваний» задач контрольной работы, решения тренировочных задач к темам семинарских занятий, решения домашних задач. Домашние задания сдаются, как правило, в виде программ в электронной форме.

Домашние работы сдаются в срок, оговариваемый при выдаче соответствующих задач. При пропуске первого срока сдачи максимальный балл, получаемый за сдачу задачи, уменьшается на штраф за первую просрочку, при пропуске второго срока сдачи максимальный балл уменьшается на штраф за вторую просрочку. После пропуска финального срока сдачи, задача не принимается.

Если в результате автоматической проверки решения в программе будет выявлена ошибка, накладывается заранее оговоренный штраф за неправильное решение. Ошибка компиляции и ошибка нарушения стиля кодирования, выявляемая автоматически, как правило, не штрафуются.

Независимо от числа неправильных попыток сдачи, успешно решенная задача не может получить балл, ниже первого неснижаемого остатка. Независимо от срока сдачи в рамках времени, отведенного для решения задачи, успешно решенная задача не может получить балл, ниже второго



неснижаемого остатка. Штраф за сроки сдачи применяется после применения штрафа за неправильные попытки и первого неснижаемого остатка. Все параметры задачи объявляются при выдаче задачи. В таблице ниже приведены возможные параметры оценивания задачи:

Название параметра	Возможное значение	Комментарий
Максимальный балл (full_score)	100 баллов	
Штраф за ошибку компиляции	0 баллов	
Штраф за ошибку, выявленную при тестировании (run_penalty)	10 баллов	
Первый неснижаемый остаток (min_score_1)	50 баллов	
Первый срок сдачи	Не позже +14 суток от момента открытия задачи в системе приема заданий	
Штраф за пропуск первого срока сдачи	50 баллов	
Второй срок сдачи	Не позже +28 суток от момента открытия задачи	
Штраф за пропуск второго срока сдачи	100 баллов	
Второй неснижаемый остаток (min_score_2)	20 баллов	
Финальный срок сдачи	31 декабря 2017 г, 23:59:59	

Решение, успешно прошедшее автоматические тесты, подлежит обязательной проверке преподавателем. Преподаватель имеет право вернуть решение на доработку, после чего решение должно быть исправлено и пересдано на автоматическую проверку. Временем сдачи решения по отношению к штрафам за пропуски сроков сдачи считается время первой сдачи, которая успешно прошла автоматические тесты, даже если она при этом была отклонена преподавателем. Отклонение решения преподавателем не налагает дополнительных штрафов, кроме, возможно, штрафов за последующие решения, ошибка в которых была выявлена автоматическим тестированием.

Все баллы, полученные за сдачу задач, суммируются и составляют набранный первичный балл студента (user_score). Определяется максимальный балл (max_score), не превосходящий сумму максимальных баллов за все задачи. Так, некоторые задачи могут быть объявлены «бонусными», баллы, полученные за их сдачу, вносят вклад в сумму студента, но максимальный балл за бонусную задачу не входит в сумму максимальных баллов за все задачи. Процентная оценка вычисляется по формуле: $\text{round}(\text{user_score} * 100.0 / \text{max_score})$. В зависимости от процентной оценки по заранее объявленным порогам вычисляется частичный накопленный балл, принимающий значение от 0 до 10.

Преподаватель, ведущий семинарские занятия, для каждого студента выставляет семинарский балл, принимающий значения от 0 до 10. При этом может учитываться активность студента на семинаре, дополнительные работы, выполненные студентом и т. п. Сумма частичного накопленного балла с весом 0.9 и семинарского балла с весом 0.1 является полным накопленным баллом и входит с весом 0.6 в итоговую оценку по курсу.

10. Учебно-методическое и информационное обеспечение дисциплины

9.1. Базовые учебники

- Э. Таненбаум, Х. Бос. Современные операционные системы. 4-е издание. СПб.: Питер, 2015. [ISBN 978-5-496-01395-6](#)
- Р. Лав. Ядро Linux: описание процесса разработки. 3-е издание. М.: ООО И.Д. Вильямс,



2015. [ISBN 978-5-8459-1944-1](#)

3. Р. Э. Брайант, Д. Р. О'Халларон. Компьютерные системы: архитектура и программирование. СПб.: БХВ-Петербург, 2005. [ISBN 5-94157-433-9](#)
4. С. А. Раго, У. Р. Стивенс. UNIX. Профессиональное программирование. 3-е издание. М.: Символ-Плюс, 2014. [ISBN 978-5-93286-216-2](#)

9.2. Основная литература

1. Э. Таненбаум, Т. Остин. Архитектура компьютера. 6-е издание. СПб.: Питер, 2015. [ISBN 978-5-496-00337-7](#)
2. Д. Паттерсон, Дж. Хеннесси. Архитектура компьютера и проектирование компьютерных систем. 4-е издание. СПб.: Питер, 2015. [ISBN 978-5-459-00291-1](#)