**Санкт-Петербургский государственный университет**

**Р А Б О Ч А Я П Р О Г Р А М М А**

**УЧЕБНОЙ ДИСЦИПЛИНЫ**

Операционные системы

Operating Systems

**Язык(и) обучения**

русский

Трудоемкость в зачетных единицах: 3

Регистрационный номер рабочей программы: 003591

**Раздел 1. Характеристики учебных занятий**

**1.1. Цели и задачи учебных занятий**

Операционная система предоставляет программистам удобную абстракцию аппаратного обеспечения компьютера, а также управляет разделением ресурсов между пользователями. Темы данного курса затрагивают вопросы, влияющие на проектирование современных операционных систем. В течение многих лет операционные системы и их абстракции становились все более сложными по сравнению с обычными прикладными программами. Прежде чем перейти к изучению реализацию внутренних алгоритмов и структур данных, необходимо убедиться, что обучающиеся осознают, насколько широко в современном мире используются операционные системы. Поэтому данные темы связаны как с использованием операционных систем, так и с их проектированием и реализацией. Многие идеи, возникшие в процессе разработки операционных систем, нашли приложения в других областях информатики, например, в параллельном программировании. Изучение внутренней структуры операционных систем отражается на таких областях, как программирование с повышенными требованиями к надежности, проектирование и реализация алгоритмов, разработка современных устройств, создание виртуальных сред, кэширование документов в Internet, создание безопасных и защищенных систем, управление сетями и многих других.

Отдельные параметры курса могут варьироваться по степени сложности в зависимости от уровня подготовки обучающихся. Курс должен быть построен на принципах компетентностного, деятельностного подхода к операционным системам как к текстовым преобразователям, позволяющим получать эффективный способ обоснованного проведения различных расчетов, с применением высокопроизводительных компьютеров с общей памятью, что предполагает распределение содержания обучения по следующим видам деятельности: изучение основных концепций параллельного программирования, изучение средств распараллеливания, изучение элементов программирования, составление алгоритмов, проведение простейших вычислений, анализ ошибок, связанных с различными погрешностями, в том числе, с алгебраически допустимым изменением порядка вычислений.

Основным методологическим принципом построения программы курса, равно как и всей концепции изучения операционных систем, в целом, является принцип поэтапного системного накопления знаний и формирования необходимых компетенций по модели: от простого и/или знакомого - к сложному и/или незнакомому, а основной методологической стратегией прохождения отдельных разделов программы является ступенчатость и цикличность, предусматривающие постепенный возврат к ранее усвоенному материалу на более высоком языковом и концептуальном уровне.

Цель изучения дисциплины: обучение понятиям и методам, сложившимся в течение 50-летней теории и практики разработки операционных систем, развитие у обучающихся доказательного, логического мышления; знакомство с различными алгоритмами, применяемыми не только при разработке и реализации операционных систем, но и при решении широкого класса задач, связанных со взаимодействием как аппаратуры с ее программной поддержкой, так и программных приложений с конечным пользователем, подготовка к самостоятельному решению различных алгоритмических задач, принадлежащих к данной предметной области.

**1.2. Требования подготовленности обучающегося к освоению содержания учебных занятий (пререквизиты)**

Программа курса предназначена для обучающихся 2 курса бакалавриата и рассчитана на обучающихся, изучавших дисциплину «Основы программирования» в объеме первого года обучения и, таким образом, владеющих навыками работы в среде той или иной операционной системы.

Максимальная эффективность программы будет обеспечена при условии, что обучающийся владеет знаниями математики в рамках курсов, читаемых на математико-механическом факультете; владеет основами программирования, достаточными для составления простейших программ; владеет навыками использования операционных систем в качестве пользователя; владеет знаниями в рамках дисциплины «Архитектура вычислительных систем».

**1.3. Перечень результатов обучения (learning outcomes)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Наименование категории (группы) компетенций | Код и наименование компетенции | Планируемые результаты обучения, обеспечивающие формирование компетенции | Код индикатора и индикатор достижения универсальной компетенции |
|  | 1 | 2 | 3 | 4 |
| 1 | Общепрофессиональные компетенции | ОПК-1 – способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности | Знание назначения и типичных возможностей операционных систем | ОПК-1.2 Определение качественных характеристик каждого компонента |
| 2 | Общепрофессиональные компетенции | 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 | Знание классических алгоритмов управления ресурсами, используемых в реализации библиотек и ядра операционных систем | ОПК-2.1 Уметь писать программный код с использованием языков программирования, определения и манипулирования данными |
| 3 | Общепрофессиональные компетенции | ОПК-5 – Способен инсталировать программное и аппаратное обеспечение для информационных и автоматизированных систем | Навыки установки операционных систем | ОПК-5.1 Установка и настройка системного и прикладного ПО, необходимого для функционирования ИС в соответствии с трудовым заданием |
| 4 | Общепрофессиональные компетенции | ОПК-7 – способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой; | Знание принципов работы файловых систем и сетевого стека, моделей распределённых систем, алгоритмов обеспечения безопасности | ОПК-7.2 Оценка осуществимости функционирования и сопровождения программного средства |
| 5 | Профессиональные компетенции | ПКП-1 – способен проектировать программные системы; | Умение эффективно использовать современные операционные системы | ПКП-1.1 Уметь разрабатывать технические спецификации на программные компоненты и их взаимодействие |
| 6 | Профессиональные компетенции | ПКП-5 – способен использовать современные системные программные средства: операционные системы, операционные и сетевые оболочки, сервисные программы; | Умение реализовывать алгоритмы управления ресурсами | ПКП-5.1 Уметь выбирать технологии и средства разработки программного обеспечения |
| 7 | Профессиональные компетенции | ПКП-6 – способен формировать суждения о проблемах современной информатики, ее категорий и связей с другими научными дисциплинами; | Навыки декомпозировать сложные системы, анализировать системы во время работы с ней | ПКП-6.1 Уметь разрабатывать процедуры проверки работоспособности и измерения характеристик программного обеспечения |
| 8 | Профессиональные компетенции | ПКП-7 – способен систематизировать и применять знания о содержании основных этапов и тенденций развития программирования, математического обеспечения и информационных технологий; | Умение проводить низкоуровневые и высокоуровневые оптимизации программ | ПКП-7.2 Оценка и выбор архитектуры развертывания каждого компонента |
| 9 | Универсальные компетенции | УК-1 – способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач; | Навык выбирать оптимальные пути решения задач | УК 1.1. Анализирует задачу, выделяя ее базовые составляющие; |
| 10 | Универсальные компетенции | УКБ-3 – способен понимать сущность и значение информации в развитии общества, использовать основные методы получения и работы с информацией с учетом современных технологий цифровой экономики и информационной безопасности; | Навык выявлять и устранять неэффективность в работе систем | УКБ-3.1. Находит и использует различные источники информации. |

Знания:

* назначения и типичных возможностей операционных систем
* классических алгоритмов управления ресурсами, используемых в реализации библиотек и ядра операционных систем
* принципов работы файловых систем и сетевого стека
* моделей распределённых систем
* алгоритмов обеспечения безопасности

Умения:

* эффективно использовать современные операционные системы
* реализовывать алгоритмы управления ресурсами
* проводить низкоуровневые и высокоуровневые оптимизации программ

Навыки:

* декомпозировать сложные системы
* анализировать системы во время работы с ней
* выбирать оптимальные пути решения задач
* выявлять и устранять неэффективность в работе систем

**1.4. Перечень и объём активных и интерактивных форм учебных занятий**

Активная и интерактивная форма учебных занятий (10 часов практических занятий в течение семестра) заключается в обсуждении в аудитории самостоятельно изученной темы и научной дискуссии по ней.

**Раздел 2. Организация, структура и содержание учебных занятий**

**2.1. Организация учебных занятий**

**2.1.1 Основной курс**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Трудоёмкость, объёмы учебной работы и наполняемость групп обучающихся | | | | | | | | | | | | | | | | | | |
| Код модуля в составе дисциплины,  практики и т.п. | Контактная работа обучающихся с преподавателем | | | | | | | | | | | | Самостоятельная работа | | | | Объём активных и интерактивных  форм учебных занятий | Трудоёмкость |
| лекции | семинары | консультации | практические  занятия | лабораторные работы | контрольные работы | коллоквиумы | текущий контроль | промежуточная  аттестация | итоговая аттестация | под руководством преподавателя | в присутствии  преподавателя | сам. раб. с использованием  методических материалов | текущий контроль (сам.раб.) | промежуточная аттестация (сам.раб.) | итоговая аттестация  (сам.раб.) |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | | | | | | | | | | | | | |
| Форма обучения: очная | | | | | | | | | | | | | | | | | | |
| Семестр 3 | 30 |  |  | 30 |  |  |  |  | 2 |  |  |  | 39 |  | 7 |  | 10 | 3 |
|  | 2-42 |  |  | 2-25 |  |  |  |  | 2-25 |  |  |  | 1-1 |  | 1-1 |  |  |  |
| ИТОГО | 30 |  |  | 30 |  |  |  |  | 2 |  |  |  | 39 |  | 7 |  |  | 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Виды, формы и сроки текущего контроля успеваемости и промежуточной аттестации | | | | | | |
| Код модуля в составе дисциплины, практики и т.п. | Формы текущего контроля успеваемости | | Виды промежуточной аттестации | | Виды итоговой аттестации  (только для программ итоговой аттестации и дополнительных образовательных программ) | |
| Формы | Сроки | Виды | Сроки | Виды | Сроки |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | |
| Форма обучения: очная | | | | | | |
| Семестр 3 |  |  | зачёт, устно, традиционная форма | по графику промежуточной аттестации |  |  |

**2.2. Структура и содержание учебных занятий**

Период обучения (модуль): Семестр 3

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование темы (раздела, части) | Вид учебных занятий | Количество часов |
| I. | Понятие операционной системы | Лекции | 6 |
| Практические занятия | 6 |
| II. | Управление памятью | Лекции | 4 |
| Практические занятия | 4 |
| III. | Планирование | Лекции | 6 |
| Практические занятия | 6 |
| IV. | Введение в параллельное программирование | Лекции | 6 |
| Практические занятия | 6 |
| V. | Файловые системы | Лекции | 2 |
| Практические занятия | 2 |
| VI. | Распределенные системы | Лекции | 6 |
| Практические занятия | 6 |
| VII | Промежуточная аттестация | Самостоятельная работа | 7 |
| Зачет | 2 |

1. Понятие операционной системы

Определения ОС: расширенная машина и менеджер ресурсов. История развития операционных систем. Типы ОС по назначению.

Взаимодействие ОС и программы (API/ABI). Исключения. Режимы работы пользователя/супервизора. Реализация виртуальной машины (из определения ОС). Сигнал - аналог исключений в Unix. Типы ядер ОС: монолитные, микроядерные, экзоядерные. Определения процесса, адресного пространства, потока. Логическое адресное пространство. Системные вызовы для управления адресным пространством. Cегментная и страничная модель памяти. Отображение адресов, таблицы трансляции, Translation Lookaside Buffer (TLB). Создание процессов, fork. Оптимизация Copy-On-Writе. Exec. Файловые дескрипторы. Таблица дескрипторов, dup, close. Реализация Bourne shell: запуск программы, перенаправление в файл. Реализация Bourne shell: pipe.

1. Управление памятью

Типы и примеры. Иерархия выделения памяти. Декомпозиция выделения, метрики. Алгоритмы на списках. Граничные маркеры. Близнецы. Раздельный поиск, хранение. Индексированный поиск. Отложенное слияние. Специализированные алгоритмы: пул, obstack, slab.

Виртуальная память. Swapping, paging. Алгоритмы выбора страницы для замещения: First In First Out (FIFO), второй шанс, часы, Not Recently Used (NRU), Least Recently Used (LRU), Not Frequently Used (NFU), рабочий набор.

1. Планирование

Измерение времени в ОС. Аппаратный таймер. Программный таймер. Назначение и простая реализация. Оптимизированные реализации программного таймера. Вычисление реального времени (gettimeofday). Простой обработчик прерывания, проблема долгого обработчика. Обработка прерывания в два этапа.

Планирование, планировщик. Политики планирования. Невытесняющее планирование, стркутуры данных планировщика. Кооперативное планирование, политики.

Вытесняющее планирование, переключение контекстов. Принудительное перекелючение. Добровольное перекелючение. Проблема ожидания события. Вытесняемое и невытесняемое ядро.

1. Введение в параллельное программирование

Состязательная ситуация, взаимное исключение. Использование прерываний. Блокирующая переменная. Алгоритм Петерсона. Семафор, mutex. Инверсия приоритета. Наследование приоритета. Монитор. Барьер.

Задача обедающих философов, решения. Задача читателя/писателя, решениe с приоритетом читателю. ... писателю. ... с честным распределением. Взаимоблокировка. Необходимые условия, предотвращение. Уклонение, алгоритм банкира. Обнаружение и восстановления.

Многопроцессорные системы. Мультипроцессоры. UMA, NUMA. Типы ОС для мультипроцессоров: собственная для процессора, AMP, SMP. Подходы к реализации планирования. Реентерабельность. Spinlock, реализация. Мультикомпьютеры. Подходы к реализации: посылка сообщений, удалённый вызов, обмен страницами.

Кеш оперативной памяти, устройство, характеристики. Проблемы с использованием в мультипроцессорных системах. Протокол MESI. Барьеры памяти. Read-Copy-Update (RCU), реализация.

1. Файловые системы

Блочные устройства. Файлы, cистемные вызовы для работы с ними. Дерево файлов, системные вызовы. Ссылки: жесткие, мягкие. Файловая система, реализация с непрерывное размещением, связного списка, связного списка в таблице, inode, журнала. Сбои, непротиворечивость ФС. Журналирование. Виртуальная файловая система. FUSE: Filesystem in Userspacе.

1. Распределенные системы

История появления компьютерных сетей, Интернета. Типы коммутации. Протокол, стек протоколов. Канальный уровень, ethernet. Сетевой уровень, IP. Транспортный уровень, UDP, TCP. Berkeley сокеты, модель программирования для клиента. Модель программирования для сервера. Реалзиации HTTP сервера: многопроцессная, многопоточная, событийная.

Распределённые системы, фундаметальные проблемы. Типы: сильносвязанная, слабосвязанная. Модели распределённых систем, синхронность. Построение остовного дерева узлов с выделенным корнем. Взаимное исключение. Остовное дерево без выделеного корня. Выбор лидера в кольце. Синхронизация времени в распределённой системе, NTP. Логическое время, алгоритм Лампорта. Векторные часы. Отказы: аварии, византийские. Консенсус, алгоритм достижения при возможных авариях, византийских отказах. Отказоустойчивость, транзакции, восстановление. Реализации транзакций: закрытое пространство, журнал. Репликация, транзакции в распределённой системе: вложенная, распределённая. Реализация распределённого подтвеждения.

Безопасность. Модели политик. Модели авторизации. Криптография. Симметричное шифрование: шифры-подстановки, XOR, одноразовые блокнотыю Ассиметричное шифрование. Гибридное шифрование. Man-in-the-Middle, Interlock. Цифровая подпись, реализация.

**Раздел 3. Обеспечение учебных занятий**

**3.1. Методическое обеспечение**

**3.1.1 Методические указания по освоению дисциплины**

Для освоения дисциплины обучающиеся должны посещать лекции и практические занятия, выполнять задания преподавателей.

Практические занятия рекомендуется проводить по теме, разобранной на лекции. Для этого предлагается демонстрировать релевантные части существующих операционных систем с открытым исходным кодом. Если соответствующие части слишком велики или сложны для учебных целей, их можно заменить на заранее подготовленные примеры либо предложить реализовать такие примеры обучающимся. В любом случае, во время практических занятий должны освещаться различные подходы к решению и проводиться их сравнение.

Практическое занятие должно оканчиваться заданием домашней работы, которое должно быть направлено на закрепление темы. На роль такого задания хорошо подходит реализация улучшенного алгоритма или механизма, выделение библиотеки из текущей реализации и т. п.

**3.1.2 Методическое обеспечение самостоятельной работы**

При самостоятельном изучении теоретического материала, выполнении практических заданий и во время подготовки доклада целесообразно использовать рекомендованную основную и дополнительную литературу.

**3.1.3.1 Методика проведения текущего контроля успеваемости и промежуточной аттестации и критерии оценивания**

Для промежуточного контроля успеваемости используются домашние работы и зачёт, проводимый в конце семестра.

Задания для домашних работ выдаются раз в неделю. Учащиеся сдают работы электронно. Работа для оценки должа быть сдана за одну неделю с момента получения задания. Каждая задача оценивается либо как зачтённая, либо как незачтённая. В случае, если задача не зачтена, обучающемуся высказываются замечания и предоставляется возможность их исправить в течении ещё одной недели.

Зачет проводитсся в традиционной устной форме. В билет входит два вопроса из различных модулей. Время подготовки ответа на вопросы билета составляет не менее 1 академического часа. Использование конспектов и учебников, а также электронных устройств хранения, обработки или передачи информации при подготовке и ответе на вопросы запрещено. После ответа на основные вопросы билета, преподаватель вправе задать уточняющие вопросы по услышанному. В качестве уточняющих вопросов рекомендуется использовать вопросы, не требующие длительного ответа, призванные определить понимание взаимосвязей между различными частями дисциплины и способность применять полученые знания на практике. Для этого, используются модельные задачи, позволяющие различные пути решения.

**3.1.3.2 Критерии оценивания итогового процента освоения дисциплины**

В течение семестра обучающимся предлагаются домашние работы, примеры которых приведены в разделе 3.1.4. Каждая домашняя задача оценивается по шкале от 0 (решение отсутствует или обладает существенными недостатками) до 1 (решение существенными недостатками не обладает). На зачете каждый ответ оценивается по шкале от 0 (нет ответа) до 10 (очень хороший ответ). Итоговая оценка выставляется по формуле , где A1 и A2 – оценка ответов, Hs – сумма оценок за домашние работы, T – общее количество домашних работ.

|  |  |  |
| --- | --- | --- |
| Итоговый процент выполнения, % | Оценка СПбГУ при проведении зачёта | Оценка ECTS |
| от 90% до 100% | зачтено | A |
| от 80% до 90% | зачтено | B |
| от 70% до 80% | зачтено | C |
| от 60% до 70% | зачтено | D |
| от 50% до 60% | зачтено | E |
| менее 50% | не зачтено | F |

**3.1.4 Методические материалы для проведения текущего контроля успеваемости и промежуточной аттестации (контрольно-измерительные материалы, оценочные средства)**

Каждая задача домашней работы должно содержать точную формулировку критерия выполненности. Проверка домашнего работы должна оценивать выполнение критерия. Возможно комментирование отдельных моментов или подходов решения. Такие комментарии должны использоваться с целью поощрения или быть началом обсуждения альтернативных стилистических подходов, но не должны влиять на оценку задания. Рекомендуется использовать автоматические средства проверки домашнего работы.

*Примеры условий домашних работ:*

1. Дана программа, использующая функции выделения и освобождения памяти. Необходимо реализовать функции работы с памятью. Критерий зачёта домашней работы: запросы удовлетворяются при наличии ресурсов.
2. Дана программа для ввода задач, набор задач с приоритетами и полтитика планирования. Задачи могут совершать ввод/вывод. Реализовать алгоритм планирования задач так, чтобы в любой момент времени исполнение происходит согласно политке планирования.

*Пример списока вопросов:*

1. Определения ОС: расширенная машина и менеджер ресурсов.
2. История развития операционных систем.
3. Взаимодействие ОС и программы (API/ABI).
4. Исключения.
5. Режимы работы пользователя/супервизора.
6. Реализация виртуальной машины (из определения ОС).
7. Сигнал - аналог исключений в Unix.
8. Типы ядер ОС: монолитные, микроядерные, экзоядерные.
9. Управление памятью, типы и примеры.
10. Иерархия выделения памяти. Декомпозиция выделения, метрики.
11. Алгоритмы на списках.
12. Граничные маркеры.
13. Близнецы.
14. Раздельный поиск, хранение.
15. Индексированный поиск. Отложенное слияние.
16. Специализированные алгоритмы: пул, obstack.
17. Slab.
18. Определения процесса, адресного пространства, потока.
19. Логическое адресное пространство. Системные вызовы для управления адресным пространством.
20. Cегментная и страничная модель памяти.
21. Отображение адресов, таблицы трансляции, Translation Lookaside Buffer (TLB).
22. Создание процессов, fork. Оптимизация Copy-On-Writе. Exec.
23. Файловые дескрипторы. Таблица дескрипторов, dup, close.
24. Реализация Bourne shell: запуск программы, перенаправление в файл.
25. Реализация Bourne shell: pipe.
26. Планирование, планировщик.
27. Политики планирования.
28. Невытесняющее планирование, стркутуры данных планировщика.
29. Кооперативное планирование, политики.
30. Измерение времени в ОС. Аппаратный таймер.
31. Программный таймер. Назначение и простая реализация.
32. Оптимизированные реализации программного таймера.
33. Вычисление реального времени (gettimeofday).
34. Простой обработчик прерывания, проблема долгого обработчика.
35. Обработка прерывания в два этапа.
36. Вытесняющее планирование, переключение контекстов.
37. Принудительное перекелючение.
38. Добровольное перекелючение. Проблема ожидания события.
39. Вытесняемое и невытесняемое ядро.
40. Состязательная ситуация, взаимное исключение.
41. Использование прерываний.
42. Блокирующая переменная.
43. Алгоритм Петерсона.
44. Семафор, mutex.
45. Инверсия приоритета. Наследование приоритета.
46. Монитор. Барьер.
47. Задача обедающих философов, решения.
48. Задача читателя/писателя, решениe с приоритетом читателю.
49. Решение задача читателя/писателя с приоритетом писателю.
50. Решение задача читателя/писателя с честным распределением.
51. Взаимоблокировка. Необходимые условия, предотвращение.
52. Уклонение, алгоритм банкира.
53. Обнаружение и восстановления.
54. Блочные устройства.
55. Файлы, cистемные вызовы для работы с ними.
56. Дерево файлов, системные вызовы.
57. Ссылки: жесткие, мягкие.
58. Файловая система, реализация с непрерывное размещением.
59. Реализация ФС со связным списком.
60. Реализация ФС со связным списком в таблице.
61. Реализация ФС с inode.
62. Реализация ФС с журналом.
63. Сбои, непротиворечивость ФС. Журналирование.
64. Виртуальная файловая система. FUSE: Filesystem in Userspacе.
65. Виртуальная память. Swapping, paging.
66. Алгоритмы выбора страницы для замещения: First In First Out (FIFO).
67. Алгоритм второго шанса.
68. Алгоритм «часы».
69. Алгоритм Not Recently Used (NRU).
70. Алгоритм Least Recently Used (LRU).
71. Алгоритм Not Frequently Used (NFU).
72. Алгоритм рабочий набор.
73. Многопроцессорные системы. Мультипроцессоры. UMA, NUMA.
74. Типы ОС для мультипроцессоров: собственная для процессора, AMP, SMP.
75. Подходы к реализации планирования.
76. Реентерабельность. Spinlock, реализация.
77. Мультикомпьютеры. Подходы к реализации: посылка сообщений, удалённый вызов, обмен страницами.
78. Кеш оперативной памяти, устройство, характеристики.
79. Проблемы с использованием в мультипроцессорных системах.
80. Протокол MESI.
81. Барьеры памяти.
82. Read-Copy-Update (RCU). Реализация.
83. История появления компьютерных сетей, Интернета.
84. Типы коммутации.
85. Протокол, стек протоколов.
86. Канальный уровень, ethernet.
87. Сетевой уровень, IP.
88. Транспортный уровень, UDP, TCP.
89. Berkeley сокеты, модель программирования для клиента.
90. Модель программирования для сервера.
91. Реалзиации HTTP сервера: многопроцессная, многопоточная, событийная.
92. Распределённые системы, фундаметальные проблемы.
93. Типы: сильносвязанная, слабосвязанная.
94. Модели распределённых систем, синхронность.
95. Построение остовного дерева узлов с выделенным корнем.
96. Взаимное исключение.
97. Остовное дерево без выделеного корня.
98. Выбор лидера в кольце.
99. Синхронизация времени в распределённой системе, NTP.
100. Логическое время, алгоритм Лампорта.
101. Векторные часы.
102. Отказы: аварии, византийские.
103. Консенсус, алгоритм достижения при возможных авариях, византийских отказах.
104. Отказоустойчивость, транзакции, восстановление.
105. Реализации транзакций: закрытое пространство, журнал.
106. Репликация, транзакции в распределённой системе: вложенная, распределённая.
107. Реализация распределённого подтвеждения.
108. Безопасность.
109. Модели политик.
110. Модели авторизации.
111. Криптография.
112. Симметричное шифрование: шифры-подстановки, XOR, одноразовые блокнотыю
113. Ассиметричное шифрование.
114. Гибридное шифрование.
115. Man-in-the-Middle, Interlock.
116. Цифровая подпись, реализация.

|  |  |  |
| --- | --- | --- |
| № | Код индикатора и индикатор достижения универсальной компетенции | Контрольно-измерительные материалы (КИМ) (тестовые вопросы, контрольные задания, кейсы и пр.) |
|  | 1 | 2 |
| 1 | ОПК-1.2 Определение качественных характеристик каждого компонента | все домашние работы независимо оцениваются с точки зрения продемонстрированных знаний в области операционных систем по шкале от 0 (не сделано) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100. |
| 2 | ОПК-2.1 Уметь писать программный код с использованием языков программирования, определения и манипулирования данными | все домашние работы независимо оцениваются с точки зрения использованного инструментария по шкале от 0 (не сделано) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100. |
| 3 | ОПК-5.1 Установка и настройка системного и прикладного ПО, необходимого для функционирования ИС в соответствии с трудовым заданием | обучающегося просят продемонстрировать установку операционной системы в определённых преподавателем условиях и/или использованного в домашних работах инструментария, каждая установка независимо оценивается с точки зрения использованного инструментария по шкале от 0 (не сделано) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100. |
| 4 | ОПК-7.2 Оценка осуществимости функционирования и сопровождения программного средства | ответы на оба вопроса на зачёте и ответы на дополнительные вопросы независимо оцениваются по шкале от 0 (нет ответа) до 10 (очень хороший ответ), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 5 | ПКП-1.1 Уметь разрабатывать технические спецификации на программные компоненты и их взаимодействие | все домашние работы независимо оцениваются с точки зрения проектирования программного обеспечения по шкале от 0 (не сделано) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100. |
| 6 | ПКП-5.1 Уметь выбирать технологии и средства разработки программного обеспечения | все домашние работы независимо оцениваются с точки зрения использованного инструментария по шкале от 0 (не сделано) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100. |
| 7 | ПКП-6.1 Уметь разрабатывать процедуры проверки работоспособности и измерения характеристик программного обеспечения | ответы на оба вопроса на зачёте и ответы на дополнительные вопросы независимо оцениваются по шкале от 0 (нет ответа) до 10 (очень хороший ответ), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 8 | ПКП-7.2 Оценка и выбор архитектуры развертывания каждого компонента | ответы на оба вопроса на зачёте и ответы на дополнительные вопросы независимо оцениваются по шкале от 0 (нет ответа) до 10 (очень хороший ответ), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |
| 9 | УК 1.1. Анализирует задачу, выделяя ее базовые составляющие; | все домашние работы независимо оцениваются с точки зрения проектирования программного обеспечения по шкале от 0 (не сделано) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100. |
| 10 | УКБ-3.1. Находит и использует различные источники информации. | ответы на оба вопроса на зачёте и ответы на дополнительные вопросы независимо оцениваются по шкале от 0 (нет ответа) до 10 (очень хороший ответ), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |

**3.1.5 Методические материалы для оценки обучающимися содержания и качества учебного процесса**

Для оценки обучающимися содержания и качества учебного процесса применяется анкетирование в соответствии с методикой и графиком, утвержденными в установленном порядке.

**3.2. Кадровое обеспечение**

**3.2.1 Образование и (или) квалификация штатных преподавателей и иных лиц, допущенных к проведению учебных занятий**

К преподаванию дисциплины могут быть допущены преподаватели, имеющие диплом о высшем техническом образовании по направлениям «Программная инженерия», «Математическое обеспечение и администрирование информационных систем» и другим смежным направлениям, связанным с информационными технологиями.

**3.2.2 Обеспечение учебно-вспомогательным и (или) иным персоналом**

Специальных требований нет.

**3.3. Материально-техническое обеспечение**

**3.3.1 Характеристики аудиторий (помещений, мест) для проведения занятий**

Учебные аудитории для проведения учебных занятий, оснащенные стандартным оборудованием, используемым для обучения в СПбГУ в соответствии с требованиями материально-технического обеспечения.

**3.3.2 Характеристики аудиторного оборудования, в том числе неспециализированного компьютерного оборудования и программного обеспечения общего пользования**

Стандартное оборудование, используемое для обучения в СПбГУ. MS Windows, MS Office, Mozilla FireFox, Google Chrome, Acrobat Reader DC, WinZip, Антивирус Касперского.

**3.3.3 Характеристики специализированного оборудования**

Специальных требований нет.

**3.3.4 Характеристики специализированного программного обеспечения**

Специальных требований нет.

**3.3.5 Перечень и объёмы требуемых расходных материалов**

Для аудиторий с маркерными досками необходимы стирающиеся маркеры в объеме, достаточном для проведения курса. Для аудиторий с меловыми досками необходим мел в объеме, достаточном для проведения курса.

**3.4. Информационное обеспечение**

**3.4.1 Список литературы**

1. Таненбаум Э. С., Херберт Б. Современные операционные системы. 3-е изд. – СПб: Питер, 2013. Мм – 15 экз.
2. Таненбаум Э., Вудхалл А. Операционные системы. Разработка и реализация. - 3-е изд. – СПб: Питер, 2007. Мм – 13 экз.
3. Таненбаум Э. и др. Распределенные системы. Принципы и парадигмы. – СПб: Питер, 2003. Мм – 13 экз.
4. Дейтел Г. Введение в операционные системы. – М., 1987. В 2-х томах. Мм – 13 экз.

**3.4.3 Перечень иных информационных источников**

1. Attiya H., Welch J. Distributed computing: fundamentals, simulations, and advanced topics. – John Wiley & Sons, 2004. – ЭР по подписке СПбГУ: http://spbu.summon.serialssolutions.com/
2. Herlihy M., Shavit N. The art of multiprocessor programming. – Morgan Kaufmann, 2011. – ЭР по подписке СПбГУ: http://spbu.summon.serialssolutions.com/
3. Сайт Научной библиотеки им. М. Горького СПбГУ: <http://www.library.spbu.ru/>
4. Электронный каталог Научной библиотеки им. М. Горького СПбГУ: <http://www.library.spbu.ru/cgi-bin/irbis64r/cgiirbis_64.exe?C21COM=F&I21DBN=IBIS&P21DBN=IBIS>
5. Перечень электронных ресурсов, находящихся в доступе СПбГУ: <http://cufts.library.spbu.ru/CRDB/SPBGU/>
6. Перечень ЭБС, на платформах которых представлены российские учебники, находящиеся в доступе СПбГУ: <http://cufts.library.spbu.ru/CRDB/SPBGU/browse?name=rures&resource%20type=8>

**Раздел 4. Разработчики программы**

Козлов Антон Павлович ассистент кафедры системного программирования st035425@student.spbu.ru