

**Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Санкт-Петербургский политехнический университет Петра Великого»**

---

УТВЕРЖДАЮ  
Директор ИКНК  
\_\_\_\_\_ Д.П. Зегжда  
«17» июня 2025 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

**«Операционные системы»**

Разработчик	Высшая школа компьютерных технологий и информационных систем
Направление (специальность) подготовки	09.03.01 Информатика и вычислительная техника
Наименование ООП	09.03.01_01 Разработка компьютерных систем
Квалификация (степень) выпускника	<b>бакалавр</b>
Образовательный стандарт	<b>СУОС</b>
Форма обучения	<b>Очная</b>

СОГЛАСОВАНО  
Руководитель ОП  
\_\_\_\_\_ Р.В. Цветков  
«19» мая 2025 г.

Соответствует СУОС  
Утверждена протоколом заседания  
высшей школы "ВШКТиИС"  
от «19» мая 2025 г. № 4

РПД разработал:  
Старший преподаватель И.А. Малышев

# 1. Цели и планируемые результаты изучения дисциплины

## Цели освоения дисциплины

1. обучение студентов принципам построения и функционирования операционных систем
2. обучение студентов практическим навыкам системного администрирования и системного программирования

## Результаты обучения выпускника

Код	Результат обучения (компетенция) выпускника ООП
ОПК-5	Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем
ИД-2 ОПК-5	Устанавливает системное программное обеспечение
ОПК-9	Способен осваивать методики использования программных средств для решения практических задач
ИД-2 ОПК-9	Выполняет написание программного кода с использованием языков программирования, определения и манипулирования данными, используя выбранную систему контроля версий и инструментальные программные средства

## Планируемые результаты изучения дисциплины

### знания:

- Знает технологии установки (инсталляции) операционных систем на физические и виртуальные машины
- языки и инструментальные средства программирования сценариев администрирования операционных систем

### умения:

- Умеет выполнять установку (инсталляцию) клиентских и серверных версий операционных систем общего назначения
- выполнять программирование, отладку и сопровождение сценариев администрирования операционных систем

### навыки:

- Владеет навыками установки (инсталляции) операционных систем семейства Microsoft Windows 2k и семейства Linux

- программирование сценариев администрирования операционных систем на языках их командных оболочек

## **2. Место дисциплины в структуре ООП**

В учебном плане дисциплина «Операционные системы» относится к модулю «Модуль цифровых компетенций».

Изучение дисциплины базируется на результатах освоения следующих дисциплин:

- Алгоритмизация и программирование
- Структуры данных

### 3. Распределение трудоёмкости освоения дисциплины по видам учебной работы и формы текущего контроля и промежуточной аттестации

#### 3.1. Виды учебной работы

Виды учебной работы	Трудоёмкость по семестрам
	Очная форма
Лекционные занятия	30
Лабораторные занятия	14
Самостоятельная работа	80
Промежуточная аттестация (зачет)	4
Промежуточная аттестация (зачет с оценкой)	0
Курсовое проектирование	16
Общая трудоёмкость освоения дисциплины	144, ач
	4, зет

#### 3.2. Формы текущего контроля и промежуточной аттестации

Формы текущего контроля и промежуточной аттестации	Количество по семестрам
	Очная форма
Текущий контроль	
Контрольные, шт.	1
Курсовые проекты, шт.	1
Промежуточная аттестация	
Зачеты, шт.	1

### 4. Содержание и результаты обучения

#### 4.1 Разделы дисциплины и виды учебной работы

№ раздела	Разделы дисциплины, мероприятия текущего контроля	Очная форма		
		Лек, ач	Лаб, ач	СР, ач

1.	ВВЕДЕНИЕ В ОПЕРАЦИОННЫЕ СИСТЕМЫ			
1.1.	Феномен операционных систем в сфере компьютерных технологий	1	0	2
1.2.	Миры операционных систем	3	0	6
1.3.	Базовые принципы построения и функционирования операционных систем	3	2	6
1.4.	Архитектуры операционных систем	3	4	8
2.	УПРАВЛЕНИЕ ПРОЦЕССАМИ			
2.1.	Процессы в операционных системах	2	3	12
2.2.	Взаимодействие процессов в операционных системах	2	3	12
3.	УПРАВЛЕНИЕ РАСПРЕДЕЛЕНИЕМ ПАМЯТИ			
3.1.	Методы распределения оперативной памяти	2	0	4
3.2.	Методы распределения виртуальной памяти	2	0	4
4.	УПРАВЛЕНИЕ ФАЙЛАМИ			
4.1.	Файловые системы локальных операционных систем	2	2	4
4.2.	Управление данными в распределённых системах	2	0	4
5.	УПРАВЛЕНИЕ УСТРОЙСТВАМИ			
5.1.	Системное программное обеспечение ввода-вывода	2	0	4
5.2.	Драйверы в операционных системах	2	0	6
6.	УПРАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯМИ			
6.1.	Пользовательские интерфейсы	2	0	4
6.2.	Категории пользователей операционных систем	2	0	4
<b>Итого по видам учебной работы:</b>		30	14	80
Зачеты, ач				0
<b>Часы на контроль, ач</b>				0
<b>Курсовое проектирование</b>			16	
<b>Промежуточная аттестация (зачет)</b>			4	
<b>Общая трудоёмкость освоения: ач / зет</b>			144 / 4	

## 4.2. Содержание разделов и результаты изучения дисциплины

Раздел дисциплины	Содержание
<b>1. ВВЕДЕНИЕ В ОПЕРАЦИОННЫЕ СИСТЕМЫ</b>	
<b>1.1. Феномен операционных систем в сфере компьютерных технологий</b>	Операционная система – ключевая категория системного программного обеспечения. Характеристические функции операционной системы. Поколения компьютерных платформ, операционных систем и других компонентов системного программного обеспечения. Базовая классификация и примеры современных операционных систем. Многообразие потребительских характеристик операционных систем. Тенденции и перспективы развития операционных систем.

<p><b>1.2. Миры операционных систем</b></p>	<p>Мир вычислений: обеспечение координации всех форм взаимодействий компонентов вычислительной системы.</p> <p>Мир терминов: формирование словаря ключевых терминов сферы программно-аппаратной автоматизации решения вычислительных задач.</p> <p>Мир возможностей: реализация всего спектра возможностей применения компьютерных технологий.</p> <p>Мир ограничений: установка допустимых современными технологиями граничных условий распределения функций управления вычислениями между человеком и компьютером.</p> <p>Мир моделей: построение широкого спектра моделей (классифицируемых по моделируемым свойствам (структуры, функции, динамика и т.д.), уровням модельных абстракции (натурные, полунатурные, имитационные)) и т.п.) для изучения, анализа, проектирования, совершенствования операционных систем и их компонентов.</p> <p>Мир алгоритмов: инкапсуляция эффективных (эталонных в своих классах) алгоритмов решения типовых задач управления ресурсами.</p> <p>Мир стандартов: унификация и совместимость компьютерных технологий обработки информации на всех уровнях абстракции.</p> <p>Мир продуктов: адаптация сервисов автоматизации обработки информации под различные потребительские запросы и компьютерные платформы.</p> <p>Мир потребностей: координация взаимодействия всех компонентов инфраструктуры средств (субъекты и объекты, поставщики и потребители, заказчики и исполнители и т.п.) решения информационных задач с помощью вычислительных систем.</p>
<p><b>1.3. Базовые принципы построения и функционирования операционных систем</b></p>	<p>Принцип управляемости: процессы и ресурсы.</p> <p>Принцип реактивности: события и обработчики событий.</p> <p>Принцип виртуальности: типизация данных.</p> <p>Принцип целостности: системный тип данных.</p> <p>Принцип совместимости: семейства программных продуктов.</p>

<b>1.4. Архитектуры операционных систем</b>	<p>Типовая архитектура операционной системы: ядро, оболочки, пользовательские интерфейсы.</p> <p>Архитектурные типы операционных систем: системы пакетной обработки, системы разделения времени, системы реального времени.</p> <p>Внутренние структуры операционных систем. Файл-ориентированные, объектно-ориентированные, агенто-ориентированные операционные системы. Проектирование операционных систем.</p>
<b>2. УПРАВЛЕНИЕ ПРОЦЕССАМИ</b>	
<b>2.1. Процессы в операционных системах</b>	<p>Ядро – ключевой компонент ОС. Принцип специализации современных ОС. Аппаратные и программные модули ядра ОС. Основные этапы загрузки ОС. Ядро ОС Unix.</p> <p>Категории процессных объектов ОС: задачи, процессы, потоки, волокна. Типовой жизненный цикл процесса. Контекст и дескриптор процесса. Процессы в Unix-подобных ОС. Процессы в Windows NT-подобных ОС.</p> <p>Категории задач управления процессами ОС: планирование, диспетчеризация, синхронизация. Динамический баланс планирования и диспетчеризации процессов. Сущность и алгоритмы решения задачи планирования процессов. Проблема синхронизации процессов, методы и средства её решения.</p>
<b>2.2. Взаимодействие процессов в операционных системах</b>	<p>Механизмы локальных межпроцессных взаимодействий: сигналы, неименованные каналы (pipes), именованные каналы (FIFO), очереди сообщений, семафоры, разделяемая память, локальные программные гнёзда (local sockets).</p> <p>Механизмы удалённых межпроцессных взаимодействий: потоки (streams), стек протоколов (TCP/IP), сетевые программные гнёзда (network sockets), вызовы удалённых процедур (RPC).</p>
<b>3. УПРАВЛЕНИЕ РАСПРЕДЕЛЕНИЕМ ПАМЯТИ</b>	
<b>3.1. Методы распределения оперативной памяти</b>	<p>Основные типы памяти и задачи распределения памяти.</p> <p>Распределение памяти разделами постоянного объёма.</p> <p>Распределение памяти разделами переменного объёма. Решение задач перераспределения памяти.</p>
<b>3.2. Методы распределения виртуальной памяти</b>	<p>Понятие виртуальной памяти. Метод страничного распределения памяти. Метод сегментного распределения памяти. Метод сегментно-страничного распределения памяти. Метод свопинга.</p>
<b>4. УПРАВЛЕНИЕ ФАЙЛАМИ</b>	



<b>4.1. Файловые системы локальных операционных систем</b>	<p>Логическая организация файловых систем. Понятие файловой системы (ФС). Типы и атрибуты файлов. Логическая организация файла. Права доступа к файлу. ФС Unix-подобных ОС (логический уровень). ФС Windows NT-подобных ОС (логический уровень). Физическая организация файловых систем. Физическая организация и адрес файла. Общая модель файловой системы. Современные архитектуры файловых систем. ФС Unix-подобных ОС (физический уровень). ФС Windows NT-подобных ОС (физический уровень).</p>
<b>4.2. Управление данными в распределённых системах</b>	<p>Примитивы передачи сообщений. Способы адресации. Блокирующие и неблокирующие примитивы. Буферизуемые и небуферизуемые примитивы. Надёжные и ненадёжные примитивы.</p> <p>Синхронизация в распределённых системах. Алгоритм синхронизации логических часов. Алгоритмы взаимного исключения. Неделимые транзакции.</p> <p>Клиент-серверная модель взаимодействий. Файловый сервер и файловый сервис. Семантика разделяемого доступа к данным. Выбор параметров архитектуры распределенной системы. Кэширование и репликация в распределённых системах.</p>
<b>5. УПРАВЛЕНИЕ УСТРОЙСТВАМИ</b>	
<b>5.1. Системное программное обеспечение ввода-вывода</b>	<p>Подсистема ввода-вывода. Устройства ввода-вывода. Архитектура системного ПО ввода-вывода. Подсистема ввода-вывода Unix-подобных ОС. Подсистема ввода-вывода Windows-подобных ОС.</p>
<b>5.2. Драйверы в операционных системах</b>	<p>Унифицированная структура драйвера. Драйверы для Unix-подобных ОС. Драйверы для Windows-подобных ОС. Примеры.</p>
<b>6. УПРАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯМИ</b>	
<b>6.1. Пользовательские интерфейсы</b>	<p>Интерфейсы операционных систем. Локальные интерфейсы пользователей ОС. Сетевые интерфейсы пользователей ОС.</p>
<b>6.2. Категории пользователей операционных систем</b>	<p>Классификация пользователей ОС. Суперпользователи ОС. Права и разрешения пользователей. Модели управления правами пользователей. Примеры.</p> <p>Учётные записи пользователей. Группы пользователей. Профили пользователей. Системные сервисы пользователей. Примеры.</p>

## 5. Образовательные технологии

Используются традиционные образовательные технологии: лекции, лабораторные занятия, контрольные работы и курсовое проектирование.

## 6. Лабораторный практикум

№ раздела	Наименование лабораторных работ	Трудоемкость, ач
		Очная форма
1.	Платформа виртуализации Oracle VM VirtualBox. Установка клиентских версий операционных систем	2
2.	Установка серверных версий операционных систем. Настройка серверных ролей операционных систем	2
3.	Командные оболочки Linux-совместимых / Windows-совместимых операционных систем	2
4.	Файловая система операционной системы Linux	2
5.	Управление процессами и потоками в операционной системе Linux / Windows	2
6.	Межпроцессные взаимодействия в операционной системе Linux / Windows	2
7.	Сетевые средства взаимодействия процессов операционных систем	2
Итого часов		14

## 7. Практические занятия

Не предусмотрено

## 8. Организация и учебно-методическое обеспечение самостоятельной работы

Текущая самостоятельная работа преимущественно ориентирована на изучение дополнительных учебных материалов по тематике лекций, подготовку к выполнению лабораторных работ и к написанию контрольной работы. Некоторые разделы дисциплины, в которых преобладают знания практического характера – приёмы программирования, инструментальные средства системного администрирования, методики решения типовых системных задач – могут (по согласованию с преподавателем) изучаться самостоятельно.

Творческая проблемно-ориентированная самостоятельная работа в наибольшей мере сконцентрирована на выполнение курсового проекта. Кроме того, учитывая исключительную значимость дисциплины в профессиональной области ИТ-специалистов, для расширения базовых знаний, умений и навыков, а также приобретения дополнительных профессиональных

компетенций – проведение поисковых работ, публичные выступления на тематических форумах и т.п. – предусмотрен дополнительный объём учебных часов для поддержки индивидуального творческого роста.

## Примерное распределение времени самостоятельной работы студентов

Вид самостоятельной работы	Примерная трудоемкость, ач
	Очная форма
<b>Текущая СР</b>	
работа с лекционным материалом, с учебной литературой	16
опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	0
самостоятельное изучение разделов дисциплины	8
выполнение домашних заданий, домашних контрольных работ	4
подготовка к лабораторным работам, к практическим и семинарским занятиям	14
подготовка к контрольным работам, коллоквиумам	4
<b>Итого текущей СР:</b>	<b>46</b>
<b>Творческая проблемно-ориентированная СР</b>	
выполнение расчётно-графических работ	0
выполнение курсового проекта или курсовой работы	16
поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме	0
работа над междисциплинарным проектом	0
исследовательская работа, участие в конференциях, семинарах, олимпиадах	12
анализ данных по заданной теме, выполнение расчётов, составление схем и моделей на основе собранных данных	6
<b>Итого творческой СР:</b>	<b>34</b>
<b>Общая трудоемкость СР:</b>	<b>80</b>

## 9. Учебно-методическое обеспечение дисциплины

### 9.1. Адрес сайта курса

Дисциплина не имеет отдельного сайта.

## 9.2. Рекомендуемая литература

### Основная литература

№	Автор, название, место издания, издательство, год (годы) издания	Год изд.	Источник
1	Таненбаум Э. Современные операционные системы: Санкт-Петербург: Питер, 2002. URL: <a href="http://www.piter.com">http://www.piter.com</a>	2002	ИБК СПбПУ
2	Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX: Санкт-Петербург: БХВ-Петербург, 2015. URL: <a href="https://ibooks.ru/reading.php?short=1&amp;productid=18515">https://ibooks.ru/reading.php?short=1&amp;productid=18515</a>	2015	Подписное издание

### Дополнительная литература

№	Автор, название, место издания, издательство, год (годы) издания	Год изд.	Источник
1	Душутин Е.В. Системное программное обеспечение. Практические вопросы разработки системных приложений: Санкт-Петербург: Изд-во Политехн. ун-та, 2016. URL: <a href="http://elib.spbstu.ru/dl/2/s17-71.pdf">http://elib.spbstu.ru/dl/2/s17-71.pdf</a>	2016	ЭБ СПбПУ
2	Душутин Е.В. Системное программное обеспечение. Межпроцессные взаимодействия в операционных системах: Санкт-Петербург: Изд-во Политехн. ун-та, 2016. URL: <a href="http://elib.spbstu.ru/dl/2/s17-72.pdf">http://elib.spbstu.ru/dl/2/s17-72.pdf</a>	2016	ЭБ СПбПУ
3	Лавров А.А., Лупин А.В., Малышев И.А. Основы операционных систем, 2022. URL: <a href="https://elib.spbstu.ru/dl/5/tr/2022/tr22-70.pdf">https://elib.spbstu.ru/dl/5/tr/2022/tr22-70.pdf</a>	2022	ЭБ СПбПУ

### Ресурсы Интернета

1. <https://intuit.ru>: <https://intuit.ru>
2. <https://linux.org.ru>: <https://linux.org.ru>
3. <https://learn.microsoft.com/ru-ru/>: <https://learn.microsoft.com/ru-ru/>

## 9.3. Технические средства обеспечения дисциплины

Для выполнения лабораторных работ используется специализированный компьютерный класс, состоящий из персональных компьютеров, объединенных в локальную компьютерную сеть и имеющих постоянный доступ к ресурсам сети Интернет. На каждом рабочем компьютере должны быть установлены две операционные системы (Linux-совместимая и Windows-совместимая), в конфигурации каждой из которых должны быть интегрированы инструментальные среды для решения задач системного администрирования (на встроенных

языках командных оболочек операционных систем) и системного программирования (на низкоуровневых алгоритмических языках общего назначения). Типовая конфигурация аппаратной платформы рабочего компьютера – IBM PC-совместимый ПК, имеющий многоядерный процессор Intel Core i7 или функционально совместимый, объем оперативной памяти – не менее 16 Гигабайт, объем дисковой памяти – не менее 0.5 Терабайт, материнская плата с поддержкой UEFI, видеоадаптер для TFT (IPS) монитора с разрешением не менее 1800 x 1200, монитор, сетевой адаптер для сети Ethernet 100M/1000M, клавиатура, мышь.

## **10. Материально-техническое обеспечение дисциплины**

Для проведения лекций в очной форме необходима лекционная аудитория, в которой установлено проекционное оборудование (видеопроектор с настенным экраном или интерактивная доска), для проведения лекций в дистанционной форме необходимо терминальное оборудование (веб-камера, микрофон) и программный клиент для подключения к корпоративной или арендуемой системе видеоконференций.

Для проведения лабораторных занятий и выполнения курсового проектирования необходима специализированная лаборатория, оснащенная персональными компьютерами с установленными на них лицензионными копиями практически изучаемых в дисциплине операционных систем (Linux- Windows- совместимых) и имеющая постоянный доступ к ресурсам сети Интернет. Количество рабочих мест в лаборатории должно определяться типовой численностью студенческой учебной группы. Оно должно быть не менее 15 (со стационарными компьютерами). Также необходимо иметь технические средства для организации дополнительных рабочих мест с помощью личных мобильных компьютеров студентов. Все практические работы должны выполняться каждым студентом строго индивидуально на отдельном рабочем месте.

## **11. Критерии оценивания и оценочные средства**

### **11.1. Критерии оценивания**

Для дисциплины «Операционные системы» формой аттестации является зачёт. Дисциплина реализуется с применением системы индивидуальных достижений.

#### **Текущий контроль успеваемости**

Максимальное значение персонального суммарного результата обучения (ПСРО) по приведенной шкале - 100 баллов

Максимальное количество баллов приведенной шкалы по результатам прохождения двух точек контроля - 80 баллов.

Подробное описание правил проведения текущего контроля с указанием баллов по каждому контрольному мероприятию и критериев выставления оценки размещается в СДО в навигационном курсе дисциплины.

#### **Промежуточная аттестация по дисциплине**

Максимальное количество баллов по результатам проведения аттестационного испытания в период промежуточной аттестации – 20 баллов приведенной шкалы.

Промежуточная аттестация по дисциплине проводится в соответствии с расписанием.

Промежуточная аттестация по дисциплине проводится по расписанию зачетов, защит курсовых проектов и экзаменов.

Форма проведения промежуточной аттестации: **портфолио + устное собеседование.**

Портфолио, представляемое на промежуточную аттестацию, включает работы, выполненные студентом в течение учебного семестра.

Портфолио состоит из трёх видов отчётных работ:

- 1) лабораторные работы;
- 2) контрольная работа;

## 2) курсовой проект.

Каждое из заданий по каждому из разделов портфолио выполняется каждым студентом индивидуально.

В состав портфолио включены семь из десяти лабораторных работ, предусмотренных учебным планом. Выбор конкретного состава этих работ определяется условиями проведения лабораторного практикума (в форме очных, дистанционных или смешанных занятий), а также наличием инструментального ПО в учебной лаборатории. Оценка выполнения заданий, предусмотренных программой каждой лабораторной работы, производится на основе индивидуально оформленного (в соответствии с Правилами оформления студенческих отчетных работ СПбПУ) и представленного (в виде электронного документа) студентом к защите отчета. Отчет должен содержать результаты, полученные студентом в ходе выполнения работы, в том числе: скриншоты, исходные тексты программ, протоколы исполнения программ, а также текстовые комментарии, раскрывающие сущность выполненных заданий и оценки их качества. Защита отчета состоит в демонстрации преподавателю работоспособности программных кодов, разработанных и реализованных студентом, а также ответах студента на вопросы, поясняющие особенности выбранных студентом методов и средств решения текущей и тематически смежных с ней задач.

В состав портфолио включена контрольная работа, которая выполняется студентом самостоятельно вне учебной лаборатории. Состав предлагаемых к решению контрольных задач является единым для всех студентов, а номер варианта исходных данных к каждой из задач, существенно влияющего на трудоёмкость решения, назначается преподавателем индивидуально каждому студенту. Все контрольные задачи посвящены разработке сценариев системного администрирования. Для Linux-совместимых операционных систем сценарии должны быть написаны на языке командной оболочки `bash`. Для Windows-совместимых операционных систем сценарии должны быть написаны либо на языке командной оболочки `Windows PowerShell`, либо на языке `Visual Basic Script`. Контрольная работа считается успешно написанной, если работоспособны и правильно выполняют все предусмотренные заданием функции: для Linux-совместимой ОС по крайней мере один сценарий (из трёх возможных), а для Windows-совместимой ОС по крайней мере три сценария (из пяти возможных).

В состав портфолио включен курсовой проект, обеспечивающий систематизацию и практическую апробацию профессиональных компетенций, полученных студентом при изучении дисциплины. Выполнение курсового проекта производится в четыре этапа: 1)



формулировка постановки задач на проектирование и календарного плана их выполнения; 2) структурное и алгоритмическое проектирование целевого программного продукта; 3) программирование, отладка и тестирование; 4) оценка потребительских характеристик результатов проектирования, оформление документации на разработку, подготовка презентации и доклада для публичной защиты и защита курсового проекта. Результат выполнения каждого этапа рекомендуется оформлять в виде промежуточного отчета в виде электронного документа. Правила оформления отчётов по этапам курсового проекта аналогичны правилам оформления отчётов по лабораторным работам. Защита итогового отчёта по курсовому проекту производится на семинаре в форме публичного выступления и свободного обсуждения результатов разработки.

В ходе проведения промежуточной аттестации каждый студент представляет своё индивидуальное портфолио и проходит краткое устное собеседование в форме опроса без подготовки, направленное на подтверждение результатов обучения. Темами собеседования являются как теоретические аспекты построения и функционирования операционных систем, так и практические приёмы решения системных задач администрирования и программирования. Количество и содержание вопросов, заданных на устном собеседовании, выбирается лицом, проводящим промежуточную аттестацию, и зависит от полноты выполнения студентом отчётных работ, входящих в состав портфолио.

Студент заслуживает оценки **"зачтено"** при выполнении всех нижеуказанных условий:

- 1) успешно защитивший результаты выполнения всех лабораторных работ, входящих в состав портфолио, по системному администрированию одной любой (по выбору студента) изученной (Linux- или Windows-совместимой) операционной системе;
- 2) успешно защитивший результаты выполнения всех лабораторных работ, входящих в состав портфолио, по системному программированию в среде одной любой (по выбору студента) изученной (Linux- или Windows-совместимой) операционной системы;
- 3) успешно написавший контрольную работу по дисциплине;
- 4) успешно защитивший курсовой проект по дисциплине;
- 5) правильно ответивший на не менее, чем 1/2 вопросов устного собеседования.

Студент заслуживает оценки **"не зачтено"** при выполнении любого из нижеуказанных условий:

- 1) студентом не было представлено портфолио;
- 2) студентом успешно защищены результаты выполнения отдельных лабораторных работ по системному администрированию и/или отдельных лабораторных работ по системному программированию, не составляющих полные циклы работ в среде операционной системы одного типа (Linux- или Windows-совместимой);
- 3) студентом не была успешно написана контрольная работа по дисциплине;
- 4) студентом не был успешно защищён курсовой проект по дисциплине;
- 5) студент не смог правильно ответить на более, чем 1/2 вопросов устного собеседования.

Результаты промежуточной аттестации, определяются на основе баллов, набранных в рамках применения, СИД

Баллы по приведенной шкале в рамках применения СИД (ПСРО+ ПА)	Оценка по результатам промежуточной аттестации
	Экзамен/диф.зачет/зачет
0 - 60 баллов	Неудовлетворительно/не зачтено
61 - 75 баллов	Удовлетворительно/зачтено
76 - 89 баллов	Хорошо/зачтено
90 и более	Отлично/зачтено

## 11.2. Оценочные средства

Оценочные средства по дисциплине представлены в фонде оценочных средств, который является неотъемлемой частью основной образовательной программы и размещается в электронной информационно-образовательной среде СПбПУ на портале etk.spbstu.ru.

## 12. Методические рекомендации по организации изучения дисциплины

При изучении дисциплины рекомендуется согласовывать проведение лекционных и лабораторных занятий, относящихся как к одному циклу (системное администрирование или системное программирование), так и к одинаковым темам внутри цикла. При этом разделы дисциплины, относящиеся к общим вопросам построения и функционирования операционных систем и применяемым на практике инструментам для их системного администрирования, полезно изучать до тематики внутреннего устройства операционных систем и алгоритмов автономного и согласованного поведения их компонентов, образующей область системного программирования.

Для успешного выполнения лабораторных работ в компьютерном классе необходима дополнительная самостоятельная подготовка. Она может производиться (в зависимости от тематики работы) либо до, либо после проведения лабораторного занятия. До начала выполнения лабораторной работы следует изучить методические материалы и подготовить прототипы (файлы с исходными текстами программ решения всех задач). После выполнения лабораторной работы на компьютере в учебной лаборатории рекомендуется повторно выполнить все задания лабораторной работы на своем личном компьютере, изменив конфигурацию системных программ (например, применив обновлённую версию операционной системы) или расширив список учебных задач (например, состав изучаемых системных сервисов).

Завершающим этапом выполнения каждой лабораторной работы является оформление и защита отчёта о результатах работы. При этом для ознакомительных лабораторных работ, посвящённых технологиям виртуализации и инсталляции системных программ и выполняемых фронтально, причём с повышенной интенсивностью (интерактивно, под непосредственным руководством преподавателя), оформление и защита отчетов не целесообразны.

Задания на контрольную работу относятся к циклу системного администрирования, изучение разделов которого завершается в первой половине семестра, поэтому написание контрольной работы полезно организовать не ранее середины учебного семестра. Форму проведения контрольной работы (очно – в компьютерном классе или заочно – в виде домашней контрольной работы) выбирает преподаватель, корректируя необходимым образом контрольные задания и критерии оценивания результатов их выполнения. Выбор метода проверки решений контрольных задач производит преподаватель. Для массовой и оперативной проверки, получая пороговые оценки («зачтено/незачтено»), можно применять средства автоматизации (например, плагины для LMS-систем). Для более тщательной проверки надлежит использовать метод «сухой прокрутки» или аналогичный ему, который позволяет настроить шкалу оценок с учётом всех критериев правильности результатов.

Для выполнения курсового проекта необходимо наличие опыта постановки и решения типовых системных задач, приобретаемого во время лабораторного практикума, поэтому рекомендуется вопросам проектирования посвятить вторую половину или даже последнюю четверть учебного семестра.

Курсовое проектирование рекомендуется разнообразить по категориям заданий и по их тематикам. Категория стандартного задания должна предусматривать разработку и реализацию на языке системного программирования многомодульной (не менее 4-х модулей) программной системы, в которой связность модулей обеспечивается различными средствами межпроцессных взаимодействий, выступающей в роли определённого системного сервиса и функционирующей с заданным показателем эффективности. Категория расширенного задания должна быть адаптируемой под уровень профессиональной подготовки автора проекта и либо включать только изучение и частичный реинжиниринг системной утилиты операционной системы с открытым исходным кодом, либо быть посвящена разработке собственной утилиты или сервиса, интегрируемого в конфигурацию операционной системы (в том числе, на уровне ядра). Вне зависимости от категории задания наряду с проектированием программного обеспечения необходимо разработать пакет программных документов, а также презентацию и доклад для публичной защиты результатов курсового проектирования.

### **13. Адаптация рабочей программы для лиц с ОВЗ**

Адаптированная программа разрабатывается при наличии заявления со стороны обучающегося (родителей, законных представителей) и медицинских показаний (рекомендациями психолого-медико-педагогической комиссии). Для инвалидов адаптированная образовательная программа разрабатывается в соответствии с индивидуальной программой реабилитации.