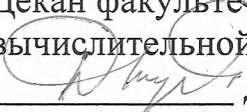


МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ИМЕНИ В.Ф. УТКИНА»

Кафедра «Космические технологии»

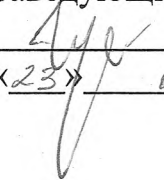
«СОГЛАСОВАНО»

Декан факультета  
вычислительной техники  
 Д.А. Перепелкин  
«25» 06 2020 г.



«УТВЕРЖДАЮ»

Проректор РОПиМД  
А.В. Корячко  
06 2020 г

Заведующий кафедрой КТ  
 С.И. Гусев  
«23» 06 2020 г.

## РАБОЧАЯ ПРОГРАММА

дисциплины

**Б1.В.12 «Операционные системы и системное программное  
обеспечение»**

Направление подготовки – 02.03.01 «Математика и компьютерные науки»

ОПОП академического бакалавриата  
«Математика и компьютерные науки»

Квалификация (степень) выпускника — бакалавр  
Форма обучения — очная

Рязань 2020

## ЛИСТ СОГЛАСОВАНИЙ

Рабочая программа является составной частью основной профессиональной образовательной программы по направлению 02.03.01 Математика и компьютерные науки, разработанной в соответствии с Федеральным государственным образовательным стандартом высшего образования по направлению 02.03.01 Математика и компьютерные науки, утвержденным приказом Минобрнауки России № 807 от 23.08.2017.

Разработчик:

доцент кафедры

«Космические технологии»

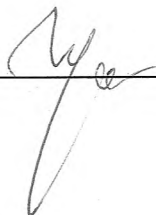


Д.А. Наумов

Программа рассмотрена и одобрена на заседании кафедры КТ  
«23» июня 2020 г., протокол № 8.

Заведующий кафедрой

«Космические технологии»



С.И. Гусев

## 1 ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

**Цель освоения дисциплины** — получение теоретических знаний о принципах построения и архитектуре современных операционных систем и сред (в том числе распределенных), обеспечивающих организацию вычислительных процессов в корпоративных информационных системах экономического, управленческого, производственного, научного и другого назначения, а также практических навыков по созданию (настройке) вычислительной среды для реализации бизнес-процессов в корпоративных сетях (интрасетях) предприятий.

### **Задачи дисциплины:**

– формирование и развитие компетенций, знаний, практических навыков и умений, обеспечивающих разработку и эксплуатацию программного обеспечения компьютерных сетей, автоматизированных систем, вычислительных комплексов, сервисов, операционных систем;

– изучение языков программирования, алгоритмов, библиотек и пакетов программ, продуктов системного программного обеспечения.

## 2 МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина «Операционные системы и системное программное обеспечение» является обязательной, относится к вариативной части блока №1 дисциплин основной профессиональной образовательной программы по направлению подготовки 02.03.01 Математика и компьютерные науки ФГБОУ ВО «РГРТУ им. В.Ф. Уткина».

Дисциплина изучается по очной форме обучения в 7 семестре на 4 курсе и базируется на знаниях, полученных в ходе изучения следующих дисциплин:

- Б1.О.01.12 Сети и телекоммуникации;
- Б1.О.01.13 Информатика;
- Б1.О.04.04 Основы компьютерных наук;
- Б1.О.04.06 Алгоритмы и структуры данных;
- Б1.В.01.01 Основы алгоритмизации и объектно-ориентированное программирование.

*Пререквизиты дисциплины.* До начала изучения учебной дисциплины обучающиеся должны:

а) знать:

- основных принципов и технологий программирования;

б) уметь:

– использовать принципы структурного проектирования при разработке программных систем;

в) владеть:

- навыками работы с различными средами разработки;
- навыками использования различных структур и методов обработки данных применительно к реальным задачам из различных предметных областей.

*Постреквизиты дисциплины.* Компетенции, полученные в результате освоения дисциплины, необходимы обучающемуся при изучении специальных дисциплин и при выполнении выпускной квалификационной работы.

### 3 КОМПЕТЕНЦИИ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Процесс изучения дисциплины направлен на формирование следующих компетенций в соответствии с ФГОС ВО, ПООП (при наличии) по данному направлению подготовки, а также компетенций (при наличии), установленных университетом. В таблице (Таблица 1) приведены коды компетенций, содержание компетенций и перечень планируемых результатов обучения по дисциплине.

**Таблица 1 — Самостоятельно устанавливаемые профессиональные компетенции выпускников и индикаторы их достижения**

Задача ПД	Объект или область знания	Код и наименование профессиональной компетенции	Код и наименование индикатора достижения профессиональной компетенции	Обоснование (ПС, анализ опыта)
<i>Направленность (профиль), специализация:</i> математика и компьютерные науки				
<i>Тип задач профессиональной деятельности:</i> организационно-управленческий				
Управление работами по созданию программных систем и комплексов.	Математические и алгоритмические модели, программы, программные системы и комплексы, методы их проектирования и реализации, способы производства, сопровождения, эксплуатации и администрирования в различных областях, в том числе в междисциплинарных. Объектами профессиональной деятельности могут быть имитационные модели сложных процессов управления, программные средства, администрирование вычислительных, информационных процессов, а также других процессов цифровой экономики	ПК-8 Способен принимать участие в управлении проектами создания информационных систем и программных комплексов	ПК-8.1 Знать: концепцию мультипрограммирования, процессов и потоков; систем управление памятью, вводом-выводом и устройствами в современных операционных системах  ПК-8.2 Уметь: разрабатывать и использовать системное программное обеспечение в задачах управления разработкой программных систем и комплексов  ПК-8.3 Владеть: навыками инсталляции и сопровождения операционных систем и сред, разработки программных моделей вычислительного процесса многопрограммных операционных систем с детализацией уровней задач, процессов, потоков и взаимоблокировок	ПС 06.022 «Системный аналитик»
		ПК-9	ПК-9.1	

Задача ПД	Объект или область знания	Код и наименование профессиональной компетенции	Код и наименование индикатора достижения профессиональной компетенции	Обоснование (ПС, анализ опыта)
		Способен учитывать знания проблем и тенденций развития рынка ПО в профессиональной деятельности	<p>Знать: знания проблем и тенденций развития рынка современных операционных систем и системного программного обеспечения</p> <p>ПК-9.2 Уметь: применять знания проблем и тенденций развития рынка ПО при использовании и разработке системного программного обеспечения</p> <p>ПК-9.3 Владеть: современными средствами разработки системного программного обеспечения</p>	«Программист»

#### 4 СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

**4.1 Объем дисциплины по семестрам (курсам) и видам занятий в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся**

Общая трудоемкость дисциплины составляет в зачетных единицах (ЗЕ) для очной формы обучения: 5,00 ЗЕ.

**Таблица 2 — Трудоемкость дисциплины**

Вид учебной работы	Часов	Семестры
		7
<b>Аудиторные занятия, всего</b>	<b>66,35</b>	<b>66,35</b>
в том числе:		
Лекции (Лек)	32	32
Лабораторные работы (Лаб)	16	16
Практические занятия (Пр)	16	16
Консультации (Конс)	2	2
Иная контактная работа (ИКР)	0,35	0,35
<b>Контактная внеаудиторная работа (КВР)</b>		
<b>Самостоятельная работа, всего</b>	<b>69</b>	<b>69</b>
в том числе:		
Контрольные работы (КоР)		
Реферат (Р)		
Иные виды самостоятельной работы (СР)	69	69
Иные формы работы (ИФР)		
<b>Курсовое проектирование/курсовая работа (КРП)</b>		
<b>Контроль</b>	<b>44,65</b>	<b>44,65</b>
Вид промежуточной аттестации (зачет, дифференцированный зачет, экзамен)	экзамен	экзамен
<b>Общая трудоемкость</b>	<b>180</b>	<b>180</b>
Зачетные единицы трудоемкости	<b>5</b>	<b>5</b>
Контактная работа (по учебным занятиям)	<b>66,35</b>	<b>66,35</b>

#### **4.2 Содержание дисциплины, структурированное по темам**

В структурном отношении программа дисциплины представлена следующими разделами:

##### **Раздел I. Основы системного программирования на языке С**

###### *Тема 1. Основы программирования на языке С*

Краткая характеристика языка С. Основные типы данных. Переменные. Константы. Перечислимый тип. Операции над базовыми типами. Приведение типов. Операторы управления.

###### *Тема 2. Структурное программирование*

Функции. Прототип. Области видимости и области существования переменных.

Препроцессор. Директивы препроцессора. Макроподстановка и условная компиляция.

Указатели. Массивы. Связь массивов и указателей.

###### *Тема 3. Структуры и работа с динамической памятью*

Интерпретация сложных деклараций. Указатели на функцию. Передача функции как параметра. Оператор *typedef*.

Структуры и объединения. Декларация структуры. Размещение структуры в памяти.

Работа с динамической памятью. Выделение памяти Освобождение памяти. Изменение размера блока памяти.

## **Раздел II. Архитектура, назначение и функции операционных систем**

### *Тема 4. Введение в операционные системы*

Структура вычислительной системы. История развития операционных систем.

Основные понятия и концепции операционных систем. Системные вызовы. Интерфейс прикладного программирования. Прерывания. Исключительные ситуации. Файлы. Процессы и нити.

Архитектурные особенности современных операционных систем. Системы с монолитным ядром. Микроядерная архитектура. Многоуровневые системы.

## **Раздел III. Управление ресурсами и задачами в Unix**

### *Тема 5. Буферизированный ввод-вывод*

Буферизированный ввод-вывод. Ввод-вывод с пользовательским буфером. Стандартный ввод-вывод. Открытие файлов. Закрытие потоков данных. Считывание из потоков. Запись в поток. Позиционирование в потоке. Сброс потока данных. Обработка ошибок.

### *Тема 6. Низкоуровневый ввод-вывод*

Файловый ввод-вывод. Открытие файлов. Чтение файлов. Запись данных. Синхронный ввод-вывод. Закрытие файлов. Позиционирование. Усечение файла.

### *Тема 7. Управление процессами*

Программы, процессы и потоки. Идентификатор процесса. Процесс инициализации. Иерархия процессов. Пользователи и группы. Запуск нового процесса. Системный вызов `exec`. Окружение. Аргументы программы. Завершение программы.

### *Тема 8. Основы многопоточного программирования*

Потоки. Создание потока. Передача аргументов потоку. Завершение потока. Ожидание потока. Получение данных, возвращаемых из потока. Получение информации из потока. Отмена потока.

### *Тема 9. Файловая система*

Основные понятия, связанные с файловой системой. Типы файлов. Файлы. Каталоги. Символические ссылки. Устройства. Каналы. Сокеты. Права доступа. Служебные файловые системы. Монтирование файловых систем. Смена, открытие и закрытие текущего каталога. Чтение каталога. Биты режима файла. Чтение ссылок.

Операции над файлами. Удаление файла. Индексы. Перемещение файла. Создание ссылок. Создание каталога. Маска прав доступа. Удаление каталога.

### *Тема 10. Методы межпроцессного взаимодействия*

Общие сведения о межпроцессном взаимодействии в Linux. Локальные методы межпроцессного взаимодействия. Удаленное межпроцессное взаимодействие.

Сигналы. Понятие сигнала в Linux. Отправка сигнала: `kill()`. Обработка сигнала: `sigaction()`. Сигналы и многозадачность. Получение дополнительной информации.

Использование общей памяти. Выделение памяти: `shmget()`. Активизация совместного доступа: `shmat()`. Отключение совместного доступа: `shmdt()`. Контроль использования памяти: `shmctl()`. Использование семафоров. Контроль за семафорами: `semctl()`.

Использование общих файлов. Размещение файла в памяти: `mmap()`. Освобождение памяти: `munmap()`. Синхронизация: `msync()`.

Каналы. Создание канала: pipe(). Перенаправление ввода-вывода: dup(). Получение дополнительной информации.

Именованные каналы FIFO. Создание именованного канала. Чтение, запись и закрытие FIFO.

Сокеты. Типы сокетов. Создание и удаление сокетов. Назначение адреса: bind(). Соединение сокетов: connect(). Прослушивание сокета: listen(). Принятие запроса на подключение: accept(). Прием и передача данных через сокет.

### 4.3 Тематический план дисциплины

Тематический план дисциплины включает информацию о следующих формах учебного процесса:

- лекции (ЛК);
- практические занятия (ПЗ);
- лабораторные работы (ЛАБ);
- самостоятельная работа (СР).
- контроль (КТР) (подготовка к зачету или экзамену).

**Таблица 3 — Тематический план**

№ п/п	Раздел дисциплины	Общая трудоемкость, всего часов	Лек	Лаб	Пр	КРП	Конс	КВР	ИФР	СР	ИКР	Контроль
<b>Семестр 7</b>												
1	Основы программирования на языке С	4								4		
2	Структурное программирование	4								4		
3	Структуры и работа с динамической памятью	4								4		
4	Введение в операционные системы	22	2		16					4		
5	Буферизированный ввод-вывод	14	4	2						8		
6	Низкоуровневый ввод-вывод	14	4	2						8		
7	Управление процессами	16	6	2						8		
8	Основы многопоточного программирования	20	6	2						8		
9	Файловая система	16	4	4						8		
10	Методы межпроцессного взаимодействия	23	6	4						13		
	Подготовка к промежуточной аттестации	47					2				0,35	44,65
	<b>Всего:</b>	<b>180</b>	<b>32</b>	<b>16</b>	<b>16</b>					<b>69</b>	<b>0,35</b>	<b>44,65</b>



В таблице (Таблица 4) приведены темы лекционных занятий по дисциплине. В таблице (Таблица 5) приведены виды лабораторных работ по дисциплине. В таблице (Таблица 6) приведены виды самостоятельных работ по дисциплине.

**Таблица 4 — Виды и содержание лекционных занятий**

<b>№ п/п</b>	<b>Тема</b>	<b>Наименование и содержание лекции</b>	<b>Часов</b>
1	4	Введение в операционные системы. Структура вычислительной системы. История развития операционных систем. Основные понятия и концепции операционных систем. Системные вызовы. Интер-фейс прикладного программирования. Прерывания. Исключительные ситуации. Файлы. Процессы и нити. Архитектурные особенности современных операционных систем. Системы с монолитным ядром. Микро-ядерная архитектура. Многоуровневые системы.	2
2	5	Буферизированный ввод-вывод. Буферизированный ввод-вывод. Ввод-вывод с пользовательским буфером. Стандартный ввод-вывод. Открытие файлов. Закрытие потоков данных. Считывание из потоков. Запись в поток. Positionирование в потоке. Сброс потока данных. Обработка ошибок.	4
3	6	Низкоуровневый ввод-вывод. Файловый ввод-вывод. Открытие файлов. Чтение файлов. Запись данных. Синхронный ввод-вывод. Закрытие файлов. Positionирование. Усечение файла.	4
4	7	Управление процессами. Программы, процессы и потоки. Идентификатор процесса. Процесс инициализации. Иерархия процессов. Пользователи и группы. Запуск нового процесса. Системный вызов <code>exec</code> . Окружение. Аргументы программы. Завершение программы	6
5	8	Основы многопоточного программирования. Потоки. Создание потока. Передача аргументов потоку. Завершение потока. Ожидание потока. Получение данных, возвращаемых из потока. Получение информации из потока. Отмена потока.	6
6	9	Файловая система. Основные понятия, связанные с файловой системой. Типы файлов. Файлы. Каталоги. Символические ссылки. Устройства. Каналы. Сокеты. Права доступа. Служебные файловые системы. Монтирование файловых систем. Смена, открытие и закрытие текущего каталога. Чтение каталога. Биты режима файла. Чтение ссылок. Операции над файлами. Удаление файла. Индексы. Перемещение файла. Создание ссылок. Создание каталога. Маска прав доступа. Удаление каталога.	4
7	10	Методы межпроцессного взаимодействия. Общие сведения о межпроцессном взаимодействии в Linux. Локальные методы меж-процессного взаимодействия. Удаленное межпроцессное взаимодействие. Сигналы. Понятие сигнала в Linux. Отправка сигнала: <code>kill()</code> . Обработка сигнала: <code>sigaction()</code> . Сигналы и многозадачность. Полу-	6

№ п/п	Тема	Наименование и содержание лекции	Часов
		<p>чение дополнительной информации.</p> <p>Использование общей памяти. Выделение памяти: shmget(). Активизация совместного доступа: shmat(). Отключение совместного доступа: shmdt(). Контроль использования памяти: shmctl().</p> <p>Использование семафоров. Контроль за семафорами: semctl().</p> <p>Использование общих файлов. Размещение файла в памяти: mmap(). Освобождение памяти: munmap(). Синхронизация: msync().</p> <p>Каналы. Создание канала: pipe(). Перенаправление ввода-вывода: dup(). Получение дополнительной информации.</p> <p>Именованные каналы FIFO. Создание именованного канала. Чтение, запись и закрытие FIFO.</p> <p>Сокеты. Типы сокетов. Создание и удаление сокетов. Назначение адреса: bind(). Со-единение сокетов: connect(). Прослушивание сокета: listen(). Принятие запроса на подключение: accept().</p> <p>Прием и передача данных через сокеты</p>	
		<b>Итого</b>	<b>32</b>

**Таблица 5 — Виды и содержание практических работ**

№ п/п	Тема	Наименование и содержание работы	Часов
1	4	Основы программирования в Linux	6
2	4	Работа с окружением	2
3	4	Библиотеки	4
4	4	Аргументы и опции программы	2
5	4	Использование отладчика gdb	2
		<b>Итого</b>	<b>16</b>

**Таблица 6 — Виды и содержание лабораторных работ**

№ п/п	Тема	Наименование и содержание работы	Часов
1	5	Стандартный ввод-вывод	2
2	6	Низкоуровневый ввод-вывод	2
3	7	Управление процессами	2
4	8	Основы многопоточного программирования	2
5	9	Файловая система	2
6	10	Сигналы	1
7	10	Использование общей памяти	1
8	10	Каналы	1
9	10	Именованные каналы FIFO	1
10	10	Сокеты	2
		<b>Итого</b>	<b>16</b>

**Таблица 7 — Виды и содержание самостоятельных работ**

<b>№ п/п</b>	<b>Тема</b>	<b>Наименование и содержание работы</b>	<b>Часов</b>
1	1	Основы программирования на языке С Изучение лекционного материала и методических указаний. Изучение дополнительных материалов. Доработка программы лабораторной работы. Оформление отчета.	4
2	2	Структурное программирование Изучение лекционного материала и методических указаний. Изучение дополнительных материалов. Доработка программы лабораторной работы. Оформление отчета.	4
3	3	Структуры и работа с динамической памятью Изучение лекционного материала и методических указаний. Изучение дополнительных материалов. Доработка программы лабораторной работы. Оформление отчета.	4
4	4	Введение в операционные системы Изучение лекционного материала и методических указаний. Изучение дополнительных материалов. Доработка программы лабораторной работы. Оформление отчета.	4
5	5	Буферизированный ввод-вывод Изучение лекционного материала и методических указаний. Изучение дополнительных материалов. Доработка программы лабораторной работы. Оформление отчета.	8
6	6	Низкоуровневый ввод-вывод Изучение лекционного материала и методических указаний. Изучение дополнительных материалов. Доработка программы лабораторной работы. Оформление отчета.	8
7	7	Управление процессами Изучение лекционного материала и методических указаний. Изучение дополнительных материалов. Доработка программы лабораторной работы. Оформление отчета.	8
8	8	Основы многопоточного программирования Изучение лекционного материала и методических указаний. Изучение дополнительных материалов. Доработка программы лабораторной работы. Оформление отчета.	8
9	9	Файловая система Изучение лекционного материала и методических указаний. Изучение дополнительных материалов. Доработка программы лабораторной работы. Оформление отчета.	8
10	10	Методы межпроцессного взаимодействия Изучение лекционного материала и методических указаний. Изучение дополнительных материалов. Доработка программы лабораторной работы. Оформление отчета.	13
<b>Итого:</b>			<b>69</b>

## **5 ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ**

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине представлен в виде оценочных материалов и приведен в Приложении «Оценочные материалы по дисциплине».

## **6 ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

### **6.1 Основная учебная литература**

1. Крищенко В.А. Основы программирования в ядре операционной системы GNU/Linux [Электронный ресурс]: учебное пособие/ Крищенко В.А., Рязанова Н.Ю.— Электрон. текстовые данные.— М.: Московский государственный технический университет имени Н.Э. Баумана, 2010.— 36 с.— Режим доступа: <http://www.iprbookshop.ru/31141.html>.— ЭБС «IPRbooks»
2. Курячий Г.В. Операционная система Linux. Курс лекций [Электронный ресурс]: учебное пособие/ Курячий Г.В., Маслинский К.А.— Электрон. текстовые данные.— Саратов: Профобразование, 2019.— 348 с.— Режим доступа: <http://www.iprbookshop.ru/88000.html>.— ЭБС «IPRbooks»
3. Гриценко Ю.Б. Операционные среды, системы и оболочки [Электронный ресурс]: учебное пособие/ Гриценко Ю.Б.— Электрон. текстовые данные.— Томск: Томский государственный университет систем управления и радиоэлектроники, 2005.— 281 с.— Режим доступа: <http://www.iprbookshop.ru/13954.html>.— ЭБС «IPRbooks»

### **6.2 Дополнительная учебная литература**

4. Волосатова Т.М. Основные концепции операционной системы UNIX [Электронный ресурс]: учебное пособие/ Волосатова Т.М., Грошев С.В., Родионов С.В.— Электрон. текстовые данные.— М.: Московский государственный технический университет имени Н.Э. Баумана, 2010.— 96 с.— Режим доступа: <http://www.iprbookshop.ru/31491.html>.— ЭБС «IPRbooks»
5. Назаров С.В. Современные операционные системы [Электронный ресурс]/ Назаров С.В., Широков А.И.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 351 с.— Режим доступа: <http://www.iprbookshop.ru/52176.html>.— ЭБС «IPRbooks»

### **6.3 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине**

В таблице (Таблица 7) приведен перечень учебно-методического обеспечения для самостоятельной работы с указанием трудоемкости самостоятельной работы.

**Таблица 8 — Виды и содержание самостоятельных работ**

<b>№ п/п</b>	<b>Наименование и содержание работы</b>	<b>Список литературы (с указанием разделов, глав, страниц)</b>	<b>Часов</b>
1	Подготовка к выполнению и сдачи лабораторных работ	ЭУМД. доп.лит.3, стр. 25-363	71
2	Подготовка к зачету, экзамену	ПУМД, осно.лит. 1-2; ЭУМД. осн.лит. 1;	45
<b>Итого:</b>			

## **7 ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО– ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ ИНТЕРНЕТ, НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

1. Информационно-правовой портал ГАРАНТ.РУ [Электронный ресурс]. – URL: <http://www.garant.ru>. – Режим доступа: свободный доступ (дата обращения 02.02.2019).
2. Справочная правовая система КонсультантПлюс [Электронный ресурс]. – URL: <http://www.consultant.ru/online/>. – Режим доступа: свободный доступ (будние дни – 20.00 - 24.00, выходные и праздничные дни – круглосуточно) (дата обращения 02.02.2019).

## **8 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Успешное усвоение курса предполагает активное, творческое участие студента на всех этапах ее освоения путем планомерной, повседневной работы. Студентам необходимо ознакомиться:

- с содержанием рабочей программы дисциплины;
- с целями и задачами дисциплины, ее связями с другими дисциплинами образовательной программы;
- методическими разработками по данной дисциплине, имеющимися на сайтах библиотеки РГРТУ;
- с графиком консультаций преподавателей кафедры.

К изучению дисциплины предъявляются следующие организационные требования:

- обязательное посещение студентом всех видов контактных занятий;
- качественная самостоятельная подготовка к практическим занятиям, активная работа на них;
- активная, ритмичная самостоятельная аудиторная и внеаудиторная работа студента в соответствии с планом-графиком;
- своевременная сдача преподавателю отчетных документов по контактным видам работ;
- в случае наличия пропущенных студентом занятий, необходимо получить консультацию по подготовке и оформлению отдельных видов заданий.

При подготовке к практическим занятиям студентам следует:

- приносить с собой рекомендованную преподавателем материалы к конкретному занятию;
- до очередного практического занятия по рекомендованным литературным источникам проработать теоретический материал, соответствующей темы занятия;
- задать преподавателю вопросы по материалу, вызвавшему затруднения в его понимании и освоении при решении задач, заданных для самостоятельного решения;
- на занятии доводить каждую задачу до окончательного решения, демонстрировать понимание проведенных расчетов (анализов, ситуаций), в случае затруднений обращаться к преподавателю.

Студентам, пропустившим занятия (независимо от причин), рекомендуется обратиться к преподавателю в день консультаций и получить индивидуальное задание.

К выполнению заданий для самостоятельной работы предъявляются следующие требования: задания должны исполняться самостоятельно и представляться в установленный срок,

а также соответствовать установленным требованиям по оформлению. Любая форма самостоятельной работы студента (подготовка к семинарскому занятию, написание эссе, курсовой работы, доклада и т.п.) начинается с изучения соответствующей литературы как в библиотеке, так и дома.

По завершению изучения дисциплины сдается зачет с оценкой. При подготовке к зачету необходимо ориентироваться на рабочую программу дисциплины, учебную и рекомендуемую литературу. Основное в подготовке к сдаче зачета – это проработка контрольных вопросов и систематизация теоретических знаний, подтверждение практическими примерами.

Подготовка студента к промежуточной аттестации по дисциплине включает в себя следующие этапы: систематическая работа в течение семестра; непосредственная подготовка в дни, предшествующие экзамену по темам курса.

Зачет проводится по билетам, содержащем практическое задание. Для успешной сдачи зачета студенты должны выполнить не менее 60% пунктов задания.

Во время испытаний промежуточной аттестации студенты могут пользоваться рабочими программами учебных дисциплин, а также справочниками и прочими источниками информации, разрешенными преподавателем.

На промежуточной аттестации нельзя пользоваться электронными средствами связи и материалами, неразрешенными преподавателем. Также не разрешается общение с другими студентами и несанкционированные перемещения по аудитории. Указанные нарушения являются основанием для удаления студента из аудитории с последующим проставлением в ведомости оценки «не удовлетворительно».

## **9 ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ**

К числу информационных технологий, программ и программного обеспечения, наличие которых необходимо для успешного изучения студентами учебной дисциплины «Обработка звука», следует отнести:

- операционная система Ubuntu (доступ: <http://code.launchpad.net/ubuntu>, лицензия GNU GPL);
- среда разработки Visual Studio Code (доступ: <http://code.visualstudio.com>, лицензия открытого программного обеспечения MIT);
- пакет создания документов Apache OpenOffice 4.1.5 (доступ: <http://openoffice.org>, лицензия: Apache License 2.0).

## **10 ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ**

Для освоения дисциплины необходимы:

- учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, практических занятий, в том числе выполнения учебных, курсовых и дипломных работ, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации;
- аудитория для проведения лабораторных работ с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду РГРТУ с установленным программным обеспечением;
- аудитория для самостоятельной работы с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду РГРТУ.



**ПРИЛОЖЕНИЕ**

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ИМЕНИ В.Ф. УТКИНА»**

**Кафедра «Космические технологии»**

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДИСЦИПЛИНЫ  
Б1.В.02.06 «Операционные системы и системное программное обеспечение»**

Направление подготовки  
02.03.01 Математика и компьютерные науки

Направленность (профиль) подготовки  
Математика и компьютерные науки

Уровень подготовки  
бакалавриат

Квалификация выпускника – бакалавр

Формы обучения – очная

**Рязань 2020 г**



## 1 ОБЩИЕ ПОЛОЖЕНИЯ

*Оценочные средства* (ОС) – это совокупность учебно-методических материалов (контрольных заданий, описаний форм и процедур проверки), предназначенных для оценки качества освоения обучающимися данной дисциплины как части ОПОП. Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины.

*Цель фонда оценочных средств* (ФОС) – предоставить объективный механизм оценивания соответствия знаний, умений и владений, приобретенных обучающимся в процессе изучения дисциплины, целям и требованиям ОПОП в ходе проведения текущего контроля и промежуточной аттестации.

*Основная задача ФОС* – обеспечить оценку уровня сформированности универсальных, общепрофессиональных и профессиональных компетенций.

## 2 ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

В таблице (Таблица 8) представлен перечень компетенций, формируемых дисциплиной.

**Таблица 9 — Компетенции дисциплины**

Коды компетенции	Содержание компетенций
<i>ПК</i>	<i>Самостоятельно устанавливаемые профессиональные компетенции выпускников и индикаторы их достижения</i>
ПК-8	Способен принимать участие в управлении проектами создания информационных систем и программных комплексов
ПК-9	Способен учитывать знания проблем и тенденций развития рынка ПО в профессиональной деятельности

В таблице (Таблица 9) представлены этапы формирования компетенций в процессе освоения основной профессиональной образовательной программы.

**Таблица 10 — Этапы формирования компетенций в процессе освоения ОПОП**

Дисциплина		Семестр							
Код	Наименование	1	2	3	4	5	6	7	8
<i>ПК-8</i>									
Б1.В.01.04	Технологии разработки информационных систем					+			
Б1.В.02.04	Основы CASE- и CALS-технологий						+		
Б1.В.02.06	Операционные системы и системное программное обеспечение							+	
Б1.В.ДВ.02.01	Методы и средства защиты информации								+
Б1.В.ДВ.02.02	Космические системы и технологии								+
Б2.В.01.01(П)	Технологическая (проектно-технологическая) практика						+		

Б3.01	Выполнение и защита выпускной квалификационной работы									+
<i>ПК-9</i>										
Б1.В.02.06	Операционные системы и системное программное обеспечение									+
Б1.В.02.07	Геоинформатика									+
Б1.В.ДВ.02.01	Методы и средства защиты информации									+
Б1.В.ДВ.02.02	Космические системы и технологии									+
Б2.В.01.01(П)	Технологическая (проектно-технологическая) практика							+		
Б3.01	Выполнение и защита выпускной квалификационной работы									+
ФТД.01	Современные технологии программирования			+						
ФТД.03	Машинное обучение и искусственный интеллект							+		

В таблице (Таблица 10) приведен перечень этапов обучения дисциплины.

В таблице (Таблица 11) представлены этапы формирования компетенций и их частей в процессе освоения дисциплины.

**Таблица 11 — Этапы обучения дисциплины**

№ п/п	Этап обучения (разделы дисциплины)
1	Основы системного программирования на языке С
2	Архитектура, назначение и функции операционных систем
3	Управление ресурсами и задачами в Unix

**Таблица 12 — Этапы формирования компетенций в процессе освоения дисциплины**

№	Код компетенции	Планируемые результаты обучения, характеризующие этапы формирования компетенций		Этапы обучения		
		Код	Результат обучения	1	2	3
1	ПК-8	ПК-8.1	Знать: концепцию мультипрограммирования, процессов и потоков; систем управление памятью, вводом-выводом и устройствами в современных операционных системах	+		+
2	ПК-8	ПК-8.2	Уметь: разрабатывать и использовать системное программное обеспечение в задачах управления разработкой программных систем и комплексов		+	+
3	ПК-8	ПК-8.3	Владеть: навыками инсталляции и сопровождения операционных систем и сред, разработки программных моделей вычислительного процесса многопрограммных операционных систем с детализацией уровней задач, процессов, потоков и взаимоблокировок	+	+	+

№	Код компетенции	Планируемые результаты обучения, характеризующие этапы формирования компетенций		Этапы обучения		
		Код	Результат обучения	1	2	3
4	ПК-9	ПК-9.1	Знать: знания проблем и тенденций развития рынка современных операционных систем и системного программного обеспечения	+		
5	ПК-9	ПК-9.2	Уметь: применять знания проблем и тенденций развития рынка ПО при использовании и разработке системного программного обеспечения		+	+
6	ПК-9	ПК-9.3	Владеть: современными средствами разработки системного программного обеспечения	+	+	+

Перечень видов оценочных средств, используемых в ФОС дисциплины, представлен в таблице (Таблица 12).

**Таблица 13 — Перечень видов оценочных средств, используемых в процессе освоения дисциплины**

№	Наименование вида оценочного средства	Характеристика оценочного средства	Представление оценочного средства в ФОС
1	Устный опрос	Средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п	Контрольные вопросы по темам/разделам дисциплины Теоретический вопросы к зачету
2	Практическое задание/задача	Средство оценки умения применять полученные теоретические знания в практической ситуации. Задача должна быть направлена на оценивание тех компетенций, которые подлежат освоению в данной дисциплине, должна содержать четкую инструкцию по выполнению или алгоритм действий	Задание к лабораторным работам Вопросы и задания к экзамену

В паспорте фонда оценочных материалов (Таблица 13) приведено соответствие между контролируемыми компетенциями и оценочными средствами контроля компетенции.

**Таблица 14 — Паспорт фонда оценочных средств дисциплины**

№	Код компетенции	Планируемые результаты обучения, характеризующие этапы формирования компетенций		Наименование оценочного средства
		Код	Результат обучения	
1	ПК-8	ПК-8.1	Знать: концепцию мультипрограммирования, процессов и потоков; систем управление памятью, вводом-выводом и устройствами в современных операционных системах	Вопросы текущего контроля  ЛР6-ЛР15  Экзамен
2	ПК-8	ПК-8.2	Уметь: разрабатывать и использовать системное программное обеспечение в задачах управления разработкой программных систем и комплексов	ЛР6-ЛР15
3	ПК-8	ПК-8.3	Владеть: навыками инсталляции и сопровождения операционных систем и сред, разработки программных моделей вычислительного процесса многопрограммных операционных систем с детализацией уровней задач, процессов, потоков и взаимоблокировок	ЛР6-ЛР15
4	ПК-9	ПК-9.1	Знать: знания проблем и тенденций развития рынка современных операционных систем и системного программного обеспечения	Вопросы текущего контроля  Экзамен
5	ПК-9	ПК-9.2	Уметь: применять знания проблем и тенденций развития рынка ПО при использовании и разработке системного программного обеспечения	ЛР1-ЛР5
6	ПК-9	ПК-9.3	Владеть: современными средствами разработки системного программного обеспечения	Вопросы текущего контроля  ЛР1-ЛР15  Экзамен

**3 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ**

Для оценки знаний, умений, навыков, характеризующих этапы формирования компетенций в процессе освоения ОПОП, применяются:

- типовые задания к лабораторным работам;
- теоретические вопросы текущего контроля;
- практические задания текущего контроля;
- вопросы промежуточной аттестации.

**3.1 Типовые задания для контроля компетенция**

*3.1.1 Контроль компетенции ПК-8*

1) Какое значение примет переменная z ?

```
int x=1, y=2, z;
```

a)  $z=x \& y | x \&\& x==y;$

b)  $z=x \& y | (x \&\& y);$

2) Изменяются ли значения переменных  $x, y, z$  и какое значение будет присвоено переменной  $k$ ?

```
int x, y, z, k;
```

```
x=y=z=1;
```

```
k=++x &\& ++y || ++z;
```

3) Как будет вычисляться выражение  $z=x+++y$  ?

4) Какое значение примет переменная  $k$  ?

```
int k;
```

```
k=~(~0<<2);
```

5) Определить значение выражения и тип результата.

a)  $-3*4L-6/5;$

б)  $(7+6)\%5/2;$

6) Какие значения примут целочисленные переменные  $x, y, z$  после выполнения следующих операторов:

```
x=3; y=3; z=1;
```

```
z+=x<y?x++;y++;
```

7) Реализовать эффективное умножение и деление переменной  $x$  на 16, используя операции поразрядного сдвига (указать возможный тип переменной  $x$  и диапазон ее начальных значений для корректного выполнения операции).

8) Используя тернарную операцию, присвоить  $z$  наименьшее из значений  $a, b$  и  $c$ .

9) `int x;`

Инвертировать 3 бита переменной  $x$ , начиная 5-го, а остальные биты оставить без изменения.

10) `int z;`

Обнулить в  $z$  все разряды, кроме 5 младших, а в 3 младших разрядах установить 1.

11) Правильно ли записан оператор присваивания  $k=i+++j;$  и в каком порядке будут выполняться операции.

12)

```
#define MAXLEN 256
```

```
char str[MAXLEN];
```

```
int i;
```

Пусть в символьный массив  $str$  записана строка, длина которой меньше  $MAXLEN$ .

a) Реверсировать строку, не используя дополнительных массивов.

b) `char str1[MAXLEN];`

Копировать строку  $str$  в  $str1$ .

c) Преобразовать строку в целое.

d) Целое число, записанное в переменной *i*, преобразовать в строку.

13. Входной поток состоит из слов. Длина слова не превышает 80 символов. Разделители между словами: пробел, запятая, конец строки.

a) Распечатать входной поток по одному слову в строке.

b) Распечатать размер самой длинной строки

c) Распечатать самую длинную строку.

14. Ввести строку из стандартного входного потока в массив типа `char`, удалить из строки комментарии. (`/* . . . */`). Дополнительные массивы не использовать.

15. Написать машинно-независимую программу, обнуляющую каждую четную двоичную единицу в числе типа `int`.

16. Написать машинно-независимую программу, транспонирующую двоичный код целого.

17. Написать функцию `getletter()`, читающую одну букву из стандартного ввода с помощью `getchar()` и возвращающую введенную букву, если это большая или малая латинская буква, и `-1` в противном случае.

18. Написать функцию, которая считывает со стандартного ввода *n* чисел (*n* – параметр функции) и возвращает их среднее арифметическое.

19. Написать рекурсивную функцию вычисления факториала неотрицательного числа.

20. Написать функцию, которая для заданных натурального *n* и вещественного *x* вычисляет значение выражения:

21. Написать функцию конкатенации строк (аналог функции `strcat`).

22. Написать функцию сравнения строк (аналог функции `strcmp`).

23. Присвоить переменной единицу, если одна строка содержится в конце другой, и ноль в противном случае.

24. Поменять местами первый отрицательный элемент вещественного массива *m1* с последним положительным элементом вещественного массива *m2*.

25. Изменить знак у всех отрицательных элементов вещественного массива *X*.

26. Определить, у скольких элементов целочисленного массива *X* равны соседние(предыдущий и последующий элементы). Записать ответ в переменную *n*.

27. Написать функцию проверки на равенство строк *s1* и *s2* при условии, что пробелы не учитываются.

28. Написать аналог функции `strstr`, которая возвращает указатель на первое вхождение одной строки в другую и нулевой указатель в противном случае. Используя функцию `strstr`, найти последнее вхождение строки "end" во введенной строке. Распечатать символы, которые следуют за этим вхождением.

29. Функции передаются 2 Си-строки: `char s[], t[]`.

Все цифры строки *s* записать в начало строки *t*, а остальные символы – в конец (в любом порядке).

30. Дано множество точек на плоскости, заданных координатами (*x*,*y*). Написать функцию, возвращающую минимальное расстояние между двумя точками.

31. Написать функцию обработки вещественной матрицы (50x20). Функция должна находить строку, содержащую наибольшее количество отрицательных элементов и распечатывать все элементы этой строки. Если таких строк несколько, то печатать все такие строки.

32. Написать функцию, осуществляющую умножение двух матриц порядка MxN и NxK.

33. Что будет выведено на печать?

```
int a[3][3]={ {1,2,3}, {4,5,6}, {7,8,9} };
int *pa[3]={a[0],a[1],a[2]};
int *p=a[0];
main()
{
    int i;
    for(i=0;i<3;i++)
        printf("%d %d %d %d %d\n",a[i][2-i], *a[i], *((a+i)+i), *pa[i],p[i]);
}
```

33. Что будет выведено на печать?

```
char *c[]={"ENTER", "NEW", "POINT", "FIRST"};
char **cp[]={c+3,c+2,c+1,c};
char ***cpp=cp;
main()
{
    printf("%s",***+cpp);
    printf("%s",*--**+cpp+3);
    printf(" %s",*cpp[-2]+3);
    printf("%s\n",cpp[-1][-1]+1);
}
```

34. Упорядочить вещественный массив x по неубыванию, используя метод сортировки выбором (найти максимальный элемент массива и переставить его с последним элементом; затем применить этот же метод ко всем элементам, кроме последнего).

35. Упорядочить вещественный массив x по неубыванию, используя метод сортировки обменом ("метод пузырька") (последовательно сравнивать пары соседних элементов: x1 с x2, x2 с x3 и т.д., и если первый элемент пары больше второго, то переставлять их – наибольший элемент окажется в конце массива; затем применить этот же метод ко всем элементам, кроме последнего).

36. Упорядочить вещественный массив x по неубыванию, используя метод сортировки вставками (пусть первые k элементов уже упорядочены по неубыванию; взять (k+1)-ый элемент и разместить его между первыми k, не нарушая порядок).

37. Дан массив, содержащий заданное количество вещественных чисел. Написать функцию, возвращающую максимальное значение этого массива.

38. Описать функцию, определяющую, сколько элементов вещественного массива X из n элементов равны числу y.

39. Написать функцию, которой передаются 2 100-элементных целых массива, определяющую, составлены ли эти 2 массива из одних и тех же чисел без учета порядка их следования, но с учетом повторяющихся чисел (функция возвращает 1 в случае положительного ответа).

40. Написать функцию, которая за 1 просмотр циклически сдвигает полученный вещественный массив на 20 позиций влево (функция для работы должна использовать вспомогательный массив).

41. Написать функции, которым в качестве параметра передается массив указателей на строки. Признак конца – нулевой указатель.

a) Написать функцию, которая распечатывает первые три символа из каждой строки.

b) Написать функцию, которая печатает самую длинную строку.

c) Написать функцию, которая печатает строки, включающие в себя строку “begin”.

42. Распечатать все аргументы командной строки ( без имени вызываемой программы ).

43. Если среди параметров командной строки, есть строки, содержащие “end” более одного раза, то распечатать символы этих строк, находящиеся после второго вхождения “end”.

44. Написать программу, печатающую имена и значения всех переменных окружения.

45. Дана строка символов. Написать функцию создания бинарного дерева (рекурсивный и нерекурсивный варианты).

46. Написать рекурсивную функцию подсчета количества узлов дерева.

47. Написать рекурсивную функцию определения высоты дерева.

48. Написать рекурсивную функцию печати узлов дерева в убывающем порядке.

49. Написать функцию печати узлов дерева по слоям (рекурсивный и нерекурсивный варианты).

50. Ввести набор слов. Разделители между словами: пробел, запятая, точка с запятой, конец строки. Признак конца текста - точка. Длина каждого слова не должна превышать 20 символов. Сформировать двоичное дерево, каждый узел которого содержал бы указатель на слово и число вхождений этого слова во входной поток. Ключом при построении дерева должно являться само слово. По окончании формирования дерева распечатать слова в убывающем порядке, в возрастающем порядке и по уровням.

51. Ввести набор слов. Слов не более 20-ти. Разделители между словами: пробел, запятая, конец строки. Длина слова не превышает 20 символов. Разместить слова в динамической памяти (лишнего места не занимать) и сформировать массив указателей на слова. Использовать в качестве признака конца массива нулевой указатель NULL. Упорядочить слова в убывающем порядке. Распечатать слова, каждое на отдельной строке.

52. Ввести набор строк. Количество строк не более 15-ти. Длина каждой строки не должна превышать 80 символов. Под каждую введенную строку зарезервировать в динамической памяти место под реальную ее длину, скопировать туда строку и записать адрес в массив указателей. Признак конца в массиве указателей – NULL( нулевой указатель). Написать функцию, которой передается в качестве параметра массив указателей, а выходное значение – адрес самой большой (лексикографически) строки. Используя этот адрес, распечатать самую большую строку.

#### Описание шкалы оценивания:

Шкала оценива-	Критерий
----------------	----------



<b>ния</b>	
«зачтено»	Задание выполнено полностью, решение корректное, возможно, содержит незначительные синтаксические ошибки (при решении задачи без использования ЭВМ)
«не зачтено»	Задание не выполнено, выполнено не самостоятельно, содержит логические ошибки

### 3.1.2 Контроль компетенции ПК-9

**Задание 1.** Выполнить отладку учебного примера, содержащего обращения к функциям.

- добавить в откомпилированный файл отладочную информацию;
- запустить программу под отладчиком;
- выполнить команды просмотра исходного кода программы;
- задать список аргументов программы в отладчике;
- выполнить пошаговое выполнение кода;
- вывести информацию в процессе отладки;
- продемонстрировать команды работы с точками останова.

**Описание шкалы оценивания:**

<b>Шкала оценивания</b>	<b>Критерий</b>
«зачтено»	Выполнены все пункты задания
«не зачтено»	Выполнены не все пункты задания

**Задание 2.** Написать и протестировать работу функции для решения следующей задачи. Дана непустая последовательность слов, разделенных одним или несколькими пробелами. Признак конца текста – точка. Распечатать этот текст, удалив из него лишние пробелы (каждую группу из нескольких пробелов заменить одним пробелом).

**Задание 3.** Написать и протестировать работу функции для решения следующей задачи

Дана непустая последовательность слов из прописных (больших) латинских букв. Слова разделены пробелом; признак конца текста – точка.

- подсчитать количество слов в этом тексте;
- подсчитать количество слов, у которых совпадают первая и последняя буквы;
- подсчитать количество слов, являющихся некоторым фрагментом латинского алфавита;
- подсчитать количество слов, содержащих все буквы, которые входят в состав слова UNIX.

**Задание 4.** Дана непустая последовательность слов из прописных (больших) латинских букв. Слова разделены пробелом; признак конца текста – точка.

- распечатать все слова, у которых не совпадают первая и последняя буквы;

– распечатать все слова, являющиеся «перевертышами», т.е. словами, одинаково читающимися слева направо и справа налево;

– распечатать текст, оставив из рядом стоящих одинаковых слов только одно;

– распечатать текст, удалив все слова, где есть символы, отличные от латинских букв.

**Задание 5.** Дана непустая последовательность слов из строчных (малых) латинских букв. Слова разделены пробелом; признак конца текста – точка. Напечатать все буквы, которые

– чаще других встречаются в данном тексте;

– входят в каждое слово данного текста;

– входят в наибольшее количество слов данного текста.

**Задание 6.** Написать и протестировать работу функции для решения следующей задачи. Входные данные (имена файлов, данные и параметры обработки данных) должны поступать в программу в виде аргументов командной строки. Предусмотреть запуск программы с ключом `--help` для вывода справочной информации для работы с системой

– определить, сколько раз в файле встречается заданный символ.

– определить, сколько раз в файле встречается заданная строка.

– распечатать все строки данного файла, содержащие заданную строку в качестве подстроки.

– написать программу, определяющую какой символ чаще других встречается в данном файле.

– определить, какая строка является самой длинной в заданном файле. Если таких строк несколько, то в качестве результата выдать первую из них.

**Задание 7.** Даны два непустых файла. Определить номер строки и номер символа в этой строке, где встречается первый символ, отличающий содержимое одного файла от другого.

**Задание 8.** В файле записана непустая последовательность целых чисел (целое число – это непустая последовательность десятичных цифр, возможно начинающаяся знаком `+` или `-`). Найти наибольшее из этих чисел.

**Задание 9.** Написать программу, создающую файл – копию заданного файла.

**Задание 10.** Создать файл, являющийся конкатенацией других файлов. Имена файлов задаются в командной строке.

**Задание 11.** Написать программу, определяющую количество строк, слов (последовательности символов, разделенных пробелами) и символов в заданном файле.

**Задание 12.** Написать и протестировать работу функции для решения следующей задачи. Входные данные (имена файлов, данные и параметры обработки данных) должны поступать в программу в виде аргументов командной строки. Предусмотреть запуск программы с ключом `--help` для вывода справочной информации для работы с системой

**Задание 13:** Дан файл `f`. Создать файл `g`, полученный из файла `f` заменой всех его прописных латинских букв соответствующими строчными.

**Задание 14:** В конец файла `f` приписать содержимое файла `g`.

**Задание 15:** Даны два файла, строки в которых упорядочены по алфавиту. Написать программу, осуществляющую слияние этих двух файлов в третий, строки которого тоже упорядочены по алфавиту.

**Задание 16:** Дан файл и две строки. Все вхождения первой строки в файл заменить второй строкой (вхождения первой строки в качестве подстроки не рассматривать).

**Задание 17:** В данном файле символы каждой строки упорядочить по алфавиту.

**Задание 18:** В данном файле упорядочить все строки по возрастанию их длин. Имя файла и максимальная длина строки задаются в командной строке.

**Задание 19-22:** В файле записана непустая последовательность целых чисел (целое число – это непустая последовательность десятичных цифр, возможно начинающаяся знаком + или - ). Создать новый файл, где

- все отрицательные числа заменены нулем;
- минимальный элемент последовательности поставлен в ее начало, а максимальный – в конец;
- переставлены максимальный и минимальный элементы этой последовательности;
- удалены все числа, являющиеся полными квадратами.

**Задание 23.** Используя низкоуровневый ввод-вывод, реализовать следующие функции:

- int putchar ( int c);
- char \*gets ( char \*s);
- int puts ( const char \*s).

**Задание 24.** Написать программу, копирующую свой стандартный ввод в стандартный вывод.

**Задание 25.** Написать программу, создающую файл – копию заданного файла. Имена файлов задаются в командной строке.

- копирование по одной букве;
- копирование блоками.

**Задание 25.** Создать файл, являющийся конкатенацией других файлов. Имена файлов задаются в командной строке.

**Задание 26.** В каждом из данных файлов удалить те N-ки байт, в которых первый байт равен коду символа s. Имена файлов, символ s и величина N задаются в командной строке.

**Задание 27.** Описать функцию, определяющую количество символов s в тексте, состоящем из нечетных N-ок байт заданного файла. Имя файла, символ s и величина N – параметры функции.

**Задание 28.** Написать программу слияния двух файлов в третий. Файл - результат формируется чередованием N-ок символов первого и второго файлов (если один из файлов длиннее другого, то его оставшаяся часть приписывается в конец файла-результата). Имена файлов и величина N задаются в командной строке.

#### Описание шкалы оценивания:

Шкала оценивания	Критерий
«зачтено»	Задание выполнено полностью, решение корректное, возможно, содержит незначительные синтаксические ошибки (при решении задачи без использования ЭВМ)

«не зачтено»	Задание не выполнено, выполнено не самостоятельно, содержит логические ошибки
--------------	---

### 3.2 Перечень вопросов промежуточной аттестации

Перечень вопросов промежуточной аттестации (зачет с оценкой) включает теоретические вопросы (

Таблица 14).

**Таблица 15 — Перечень теоретических вопросов промежуточной аттестации**

№	Вопрос	Код компетенции или ее части
1.	Понятие операционной системы (ОС), цели ее работы.	ПК-8, ПК-9
2.	Классификация операционных систем: по типу централизации; по особенностям алгоритмов управления ресурсами; по особенностям областей использования; по особенностям аппаратных платформ.	ПК-8, ПК-9
3.	Ядро операционной системы.	ПК-8, ПК-9
4.	Классификация операционных систем по типу архитектуры ядра системы.	ПК-8, ПК-9
5.	Режим ядра (защищенный) и пользовательский режим.	ПК-8, ПК-9
6.	Микроядерная архитектура ОС.	ПК-8, ПК-9
7.	Принципы функционирования ОС. Обзор существующих ОС.	ПК-8, ПК-9
8.	Мультипрограммный режим.	ПК-8, ПК-9
9.	Режим реального времени.	ПК-8, ПК-9
10.	Режим пакетной обработки.	ПК-8, ПК-9
11.	Предпосылки появления ОС.	ПК-8, ПК-9
12.	Диспетчер как прообраз ОС.	ПК-8, ПК-9
13.	Эволюция ОС.	ПК-8, ПК-9
14.	Обзор функций ОС: управление памятью, файлами, процессами, сетями, командными интерпретаторами, сервисы ОС, системные вызовы.	ПК-8, ПК-9
15.	Уровни абстракции ОС.	ПК-8, ПК-9
16.	Архитектура компьютерной системы.	ПК-8, ПК-9
17.	Управление прерываниями, памятью, вводом-выводом.	ПК-8, ПК-9
18.	Иерархия памяти.	ПК-8, ПК-9
19.	Кэширование.	ПК-8, ПК-9
20.	Защита памяти.	ПК-8, ПК-9
21.	Понятие процесса, задачи.	ПК-8, ПК-9
22.	Управление процессами.	ПК-8, ПК-9
23.	Процессы и потоки управления.	ПК-8, ПК-9
24.	Процессы с поддержкой многопоточности.	ПК-8, ПК-9
25.	Коммуникация и синхронизация процессов.	ПК-8, ПК-9
26.	Высокоуровневые подходы.	ПК-8, ПК-9
27.	Семафоры. Тупики.	ПК-8, ПК-9
28.	Модели для анализа свойств асинхронных процессов.	ПК-8, ПК-9
29.	Понятие ресурса. ОС как средство распределения вычислительных ресурсов.	ПК-8, ПК-9

№	Вопрос	Код компетенции или ее части
30.	Принципы управления ресурсами.	ПК-8, ПК-9
31.	Стратегии диспетчеризации с перераспределением процессора.	ПК-8, ПК-9
32.	Многоуровневые очереди с обратной связью.	ПК-8, ПК-9
33.	Основные концепции планирования загрузки процессора.	ПК-8, ПК-9
34.	Уровни планирования.	ПК-8, ПК-9
35.	Состояния процессов.	ПК-8, ПК-9
36.	Основная память.	ПК-8, ПК-9
37.	Привязка адресов.	ПК-8, ПК-9
38.	Связные и несвязные распределения памяти.	ПК-8, ПК-9
39.	Виртуальная память.	ПК-8, ПК-9
40.	Управление виртуальной памятью.	ПК-8, ПК-9
41.	Виртуальная память.	ПК-8, ПК-9
42.	Страничная организация виртуальной памяти.	ПК-8, ПК-9
43.	Обработка отсутствия страницы в памяти (page fault).	ПК-8, ПК-9
44.	Файлы и файловые системы.	ПК-8, ПК-9
45.	Распределенные файловые системы.	ПК-8, ПК-9

Для промежуточной аттестации используется шкала оценивания устных ответов приведенная в таблице (Таблица 16).

#### **4 МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ХАРАКТЕРИЗУЮЩИЕ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ**

Контроль сформированности компетенций по дисциплине проводится:

- в форме текущего контроля успеваемости (лабораторные работы, самостоятельная работа);

- в форме промежуточной аттестации (зачет с оценкой).

Текущий контроль успеваемости проводится с целью:

- определения степени усвоения учебного материала;
- своевременного выявления и устранения недостатков в подготовке обучающихся и принятия необходимых мер по совершенствованию методики преподавания учебной дисциплины;

- организации работы обучающихся в ходе учебных занятий и самостоятельной работы;

- оказания обучающимся индивидуальной помощи (консультаций).

К контролю текущей успеваемости относится проверка обучающихся:

- по результатам выполнения заданий на лабораторных работах;

- по результатам выполнения заданий для самостоятельной работы.

Текущая успеваемость студента оценивается **положительно**, если студент полностью выполнил все работы согласно графику текущего контроля, в противном случае текущая успеваемость студента оценивается **отрицательно**.

Результаты текущего контроля успеваемости учитываются преподавателем при проведении промежуточной аттестации. Отставание студента от графика текущего контроля успеваемости по изучаемой дисциплине приводит к образованию **текущей задолженности**.

Промежуточная аттестация проводится в форме **зачета с оценкой**.

Форма проведения зачета – устный ответ (на первый вопрос билета) и выполнение практического задания (второй вопрос билета), сформулированным с учетом содержания учебной дисциплины. В процессе подготовки к устному ответу обучающийся может составить в письменном виде план ответа, включающий в себя определения, формулы, рисунки и т.п. Практическое задание выполняется на компьютере и предоставляется в электронном виде

## **5 ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ НА РАЗЛИЧНЫХ ЭТАПАХ ИХ ФОРМИРОВАНИЯ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ**

Основными этапами формирования компетенций при изучении студентами дисциплины являются последовательное изучение содержательно связанных между собой разделов (тем) учебных занятий. Изучение каждого раздела (темы) предполагает овладение студентами необходимыми компетенциями.

Сформированность каждой компетенции в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

– пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;

– продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;

– эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

Критерии и шкалы для интегрированной оценки уровня сформированности компетенций приведены в таблице (Таблица 15).

**Таблица 16 — Критерии оценивания компетенций**

<b>Индикаторы компетенции</b>	<b>Уровень сформированности компетенции</b>		
	<b>пороговый</b>	<b>продвинутый</b>	<b>эталонный</b>
Полнота знаний	Минимально допустимый уровень знаний. Допущено много негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки, без ошибок
Наличие умений	Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме	Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продемонстрированы все основные умения, решены все основные задачи с отдельными несущественными недочетами, выполнены все задания в полном объеме.

Индикаторы компетенции	Уровень сформированности компетенции		
	пороговый	продвинутый	эталонный
Наличие навыков (владение опытом)	Имеется минимальный набор навыков для решения стандартных задач с некоторыми недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторыми недочетами	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов.
Мотивация (личностное отношение)	Учебная активность и мотивация низкие, слабо выражены, стремление решать задачи качественно	Учебная активность и мотивация проявляются на среднем уровне, демонстрируется готовность выполнять поставленные задачи на среднем уровне качества	Учебная активность и мотивация проявляются на высоком уровне, демонстрируется готовность выполнять все поставленные задачи на высоком уровне качества
Характеристика сформированности компетенции	Сформированность компетенции соответствует минимальным требованиям. Имеющихся знаний, умений, навыков в целом достаточно для решения практических (профессиональных) задач, но требуется дополнительная практика по большинству практических задач	Сформированность компетенции в целом соответствует требованиям, но есть недочеты. Имеющихся знаний, умений, навыков и мотивации в целом достаточно для решения практических (профессиональных) задач, но требуется дополнительная практика по некоторым профессиональным задачам.	Сформированность компетенции полностью соответствует требованиям. Имеющихся знаний, умений, навыков и мотивации в полной мере достаточно для решения сложных практических (профессиональных) задач.

Критерии и шкалы для оценивания ответов на устные вопросы приведены в таблице (Таблица 16).

**Таблица 17 — Критерии и шкала оценивания устных ответов**

№ п/п	Критерии оценивания	Оценка/Зачет
1	1) полно и аргументированно отвечает по содержанию задания; 2) обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только по учебнику, но и самостоятельно составленные; 3) излагает материал последовательно и правильно.	Отлично
2	студент дает ответ, удовлетворяющий тем же требованиям, что и для оценки «5», но допускает 1-2 ошибки, которые сам же исправляет	Хорошо
3	ставится, если студент обнаруживает знание и понимание основ-	Удовлетворительно

	ных положений данного задания, но: 1) излагает материал неполно и допускает неточности в определении понятий или формулировке правил; 2) не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; 3) излагает материал непоследовательно и допускает ошибки	
4	студент обнаруживает незнание ответа на соответствующее задание, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал; отмечаются такие недостатки в подготовке студента, которые являются серьезным препятствием к успешному овладению последующим материалом	Не удовлетворительно

Критерии и шкалы для оценивания результатов выполнения практических задач приведены в таблице (Таблица 17).

**Таблица 18 — Критерии и шкала оценивания практических задач**

<b>№ п/п</b>	<b>Критерии оценивания</b>	<b>Оценка/Зачет</b>
1	Студентом выполнены все этапы практического задания, обосновано применены требуемые методы, техники, технологии, инструменты. Результат выполнения задания корректен. Результаты полно и грамотно оформлены в виде отчета.	Отлично
2	Студентом выполнены все этапы практического задания с несущественными ошибками, обосновано применены требуемые методы, техники, технологии, инструменты. Результат выполнения задания корректен. Результаты полно и грамотно оформлены в виде отчета.	Хорошо
3	Студентом выполнены все этапы практического задания с несущественными ошибками, часть методов, техник, технологий, инструментов применена необоснованно или некорректно. Результат выполнения задания в целом корректен. Результаты оформлены в виде отчета с несущественными ошибками.	Удовлетворительно
4	Студентом не выполнена часть этапов практического задания, либо выполнена с существенными ошибками, либо требуемые методы, техники, технологии, инструменты не применены, либо результат выполнения задания не корректен, либо результаты не оформлены в виде отчета или оформлены с существенными ошибками.	Не удовлетворительно