

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Узунов Федор Владимирович

Должность: Ректор

Дата подписания: 19.06.2026 18:16:49

Уникальный программный ключ: fd935d10451b860e912264c0378f8448452bfd603f94388008a29877a6bcbf5

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
«ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ»**

«УНИВЕРСИТЕТ ЭКОНОМИКИ И УПРАВЛЕНИЯ»

Факультет экономики, управления и юриспруденции

Кафедра управления и бизнес-информатики

УТВЕРЖДАЮ

Проректор по учебно-методической работе

/ Г.П. Узунова

«02» февраля 2026 г



ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Направление подготовки

09.03.01 Информатика и вычислительная техника

Профиль: специалист по компьютерным системам

Квалификация выпускника: бакалавр

Для всех
форм обучения

Симферополь, 2026 г.

1. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ

Средства оценивания в ходе текущего контроля:

- устные опросы в ходе семинарских занятий;
- рефераты;
- тестирование;
- практические задания, выполняемые в ходе семинарского (практического) занятия или рекомендуемые для самостоятельной работы.

Коды компетенции	Результаты освоения ОПОП	Перечень планируемых результатов обучения по дисциплине
ОПК-3	Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	<p>ОПК-3.1. Знать: принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.</p> <p>ОПК-3.2. Уметь: решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.</p> <p>ОПК-3.3. Владеть: навыками подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с учетом требований информационной безопасности.</p>
ОПК-8	Способен разрабатывать алгоритмы и программы, пригодные для практического применения	<p>ОПК-8.1. Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения.</p> <p>ОПК-8.2. Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули.</p> <p>ОПК-8.3. Владеть: языком программирования; навыками отладки и тестирования работоспособности программы.</p>

1.1 Вопросы к текущему контролю

- 1 Что такое класс и объект в объектно-ориентированном программировании (ООП)?
- 2 Какие основные принципы ООП вы знаете? Что такое инкапсуляция?
- 3 Что такое поля и методы класса? В чем их отличия?
- 4 Для чего используются модификаторы доступа public и private?
- 5 Что такое конструктор класса и каково его назначение?
- 6 В чем разница между классом и структурой (struct) в C#?

- 7 Что такое перечисление (enum) и для чего оно применяется?
- 8 Как объявить массив объектов? Как обратиться к элементу массива?
- 9 Что такое список (List<T>) и в чем его преимущество перед обычным массивом?
- 10 Почему строки в C# (string) являются неизменяемыми (immutable)?
- 11 Для чего используется класс StringBuilder и в каких случаях он эффективнее string?
- 12 Что такое наследование классов? Как создать производный класс в C#?
- 13 В чём разница между модификаторами доступа protected и private при наследовании?
- 14 Что такое перекрытие метода (override) и для чего нужно ключевое слово virtual?
- 15 Объясните разницу между перегрузкой (overload) и перекрытием (override) методов.
- 16 Что такое абстрактный класс? Могут ли у него быть реализованные методы?
- 17 Для чего применяется ключевое слово sealed при описании класса или метода?
- 18 Что такое интерфейс в C#? Может ли интерфейс содержать реализацию методов (до C# 8.0 и после)?
- 19 В чём принципиальное отличие интерфейса от абстрактного класса?
- 20 Как реализовать множественное наследование в C# с помощью интерфейсов?
- 21 Что такое свойства (property) в C# и как они обеспечивают инкапсуляцию полей?
- 22 Для чего нужны индексы и как они позволяют обращаться к объектам как к массивам?
- 23 Что такое делегат в C#? Как он связан с методами обратного вызова?
- 24 Объясните разницу между анонимными методами и лямбда-выражениями.
- 25 Что такое событие (event) в C# и как оно связано с делегатами?
- 26 Для чего используются параметры sender и EventArgs в обработчиках событий?
- 27 Как создать собственный класс, производный от EventArgs, для передачи данных события?
- 28 Что такое позднее связывание, ковариантность и контравариантность в контексте делегатов и методов?
- 29 Как правильно генерировать исключения (throw) и создавать собственные классы исключений для надежного кода?
- 30 Опишите жизненный цикл события: от его объявления и генерации до динамического связывания с обработчиком и отладки в Visual Studio.

1.2 Темы рефератов:

1. Архитектура ядра ОС Windows (NT, XP, Vista, 7, 8).
2. Сравнительный анализ управления памятью в ОС FreeBSD и Linux.
3. Управление процессами в ОС Windows.
4. Управление процессами в ОС FreeBSD и Linux.
5. Обзор ОСРВ и их особенностей.
6. Планировщик ОСРВ (на примере «малых» ОС: SOS, Contiki, µC/OS, FreeRTOS, RETOS, MANTIS, t-kernel, других).
7. Телекоммуникации в ОСРВ (на примере «малых» ОС: SOS, Contiki, µC/OS, FreeRTOS, RETOS, MANTIS, t-kernel, других).
8. ОС, используемые во встроенном программировании.
9. Файловые системы ОС Windows (FAT, NTFS, HPFS, CDFS, UDF).
10. Параллельное развитие и взаимное влияние Windows и MacOS.
11. Микроядро Mach и ОС на его основе.
12. Реестр Windows.
13. Сравнительный анализ файловых систем UNIX (SFS, FFS, Ext2, Ext3, RaiserFS, NFS).
14. Реализация многозадачности в современных ОС.
15. Управление процессами в ОС мобильных устройств (на примере Android, Symbian, Windows Phone 7, Bada, webOS, других).
16. Управление ресурсами мобильных устройств (на примере ОС Android, Symbian, Windows Phone 7, Bada, webOS, других).

17. Управление устройствами ввода-вывода в ОС Unix.
18. Управление устройствами ввода-вывода в ОС Windows.
19. Особенности серверных ОС.
20. Стандарты в области разработок ОС (POSIX, ISO 7498, сетевые протоколы и др.).
21. Защищённость и надёжность современных ОС.
22. ОС мэйнфреймов корпорации IBM (DOS, OS/360, VM, OS/390, Z/VM).
23. ОС, реализующие концепцию виртуальной машины.
24. Развитие интерфейсов ОС (графический, командный, прикладных программ).
25. Многозадачность в Windows от NT до 8.
26. Сравнительный анализ ОС Windows XP, Vista и 8.
27. Взаимодействие процессов в ОС.
28. Жизненный цикл ОС.
29. Оболочки Unix (sh, ksh, csh, bash и др.).
30. Особенности и применение ОС QNX.
31. Алгоритмы управления памятью в ОС Windows и Unix.
32. ОС производства IBM (AIX, OS/2).

1.3 Тестовые задания

1. Что такое ядро операционной системы?

- а) Программа для управления файлами.
- б) Совокупность системных библиотек.
- в) Основная часть ОС, управляющая ресурсами и процессами. (*Правильный ответ: в*)
- г) Графическая оболочка пользователя.

2. Какой из режимов работы процессора обеспечивает максимальный доступ к оборудованию?

- а) Пользовательский режим.
- б) Режим ядра. (*Правильный ответ: б*)
- в) Безопасный режим.
- г) Режим эмуляции.

3. Что такое процесс в контексте ОС?

- а) Запущенная программа. (*Правильный ответ: а*)
- б) Файл на диске.
- в) Сетевое соединение.
- г) Устройство ввода-вывода.

4. Что такое «диспетчеризация» в ОС?

- а) Управление памятью.
- б) Переключение процессора между процессами. (*Правильный ответ: б*)
- в) Управление файловой системой.
- г) Обработка ошибок.

5. Какой механизм используется для синхронизации потоков?

- а) Файловый дескриптор.
- б) Семафор. (*Правильный ответ: б*)
- в) Драйвер устройства.
- г) Системный вызов.

6. Что такое «фрагментация» памяти?

- а) Увеличение объема памяти.
- б) Разделение памяти на мелкие, не смежные участки. (*Правильный ответ: б*)
- в) Объединение памяти в один блок.
- г) Ошибка при чтении памяти.

7. Какой уровень абстракции предоставляет ОС для работы с устройствами?

- а) Файловая система.
- б) Драйвер устройства. (*Правильный ответ: б*)
- в) Оболочка пользователя.

г) Сетевой протокол.

8. Какой компонент ОС отвечает за управление файлами и каталогами?

а) Диспетчер памяти.

б) Файловая система. (Правильный ответ: б)

в) Планировщик задач.

г) Монитор ресурсов.

9. Что такое «свопинг»?

а) Обмен данными между процессами.

б) Перенос страниц памяти на диск и обратно. (Правильный ответ: б)

в) Смена приоритетов процессов.

г) Управление очередью ввода-вывода.

10. Какой формат пакетов используется в Debian для установки программного обеспечения?

а) .rpm.

б) .deb. (Правильный ответ: б)

в) .pkg.

г) .msi.

11. Какие функции выполняет операционная система?

а) Управление памятью.

б) Управление процессами.

в) Компиляция программ.

г) Управление устройствами.

Ответ: а, б, г.

12. Что относится к системному программному обеспечению?

а) Драйверы устройств.

б) Операционная система.

в) Текстовый редактор.

г) Утилиты.

Ответ: а, б, г.

13. Какие состояния может иметь процесс?

а) Выполнение.

б) Ожидание.

в) Завершение.

г) Компиляция.

Ответ: а, б, в.

14. Что относится к видам операционных систем?

а) Однопользовательские.

б) Многозадачные.

в) Компиляторные.

г) Распределенные.

Ответ: а, б, г.

15. Какие методы планирования процессов существуют?

а) FCFS.

б) SJF.

в) FIFO для памяти.

г) Round Robin.

Ответ: а, б, г.

16. Что относится к ресурсам компьютера?

а) Процессор.

б) Оперативная память.

в) Файлы.

г) Пользователь.

Ответ: а, б, в.

17. Какие виды памяти существуют?

- а) Оперативная.
- б) Виртуальная.
- в) Постоянная.
- г) Логическая.

Ответ: а, б, в.

18. Что относится к файловым системам?

- а) NTFS.
- б) FAT32.
- в) ext4.
- г) HTTP.

Ответ: а, б, в.

19. Какие операции выполняются над файлами?

- а) Создание.
- б) Удаление.
- в) Компиляция.
- г) Чтение.

Ответ: а, б, г.

20. Какие компоненты включают ОС?

- а) Ядро.
- б) Оболочка.
- в) BIOS.
- г) Системные утилиты.

Ответ: а, б, г.

21. Установить соответствие между термином и определением:

1. Процесс	А) Интерфейс взаимодействия пользователя в ОС
2. Поток	Б) Минимальная единицы выполнения внутри процесса
3. Ядро	В) Центральная часть ОС, управляющая ресурсами
4. Оболочка	Г) Выполняемая программа в памяти

22. Установить соответствие между термином и определением:

1. Планировщик	А) Структура, содержащая процессы, ожидающие выполнения
2. Контекст процесса	Б) Набор данных, описывающих состояние процесса
3. Квант времени	В) Компонент, выбирающий следующий процесс
4. Очередь процессов	Г) Время, выделяемое процессу

23. Установить соответствие между термином и определением:

1. Виртуальная память	А) Непрерывная область логической
-----------------------	-----------------------------------

	памяти
2. Фрагментация	Б) Блок фиксированного размера
3. Страницы	В) Потери памяти из-за разбиения
4. Сегмент	Г) Расширение памяти за счет диска

24. Установить соответствие между термином и определением:

1. Файл	А) Структура для организации файлов
2. Каталог	Б) Логическая единица хранения данных
3. Файловая система	В) Числовой идентификатор открытого файла
4. Дескриптор файла	Г) Совокупность методов хранения и доступа

25. Установить соответствие между термином и процессом:

1. FCFS	А) По времени выполнения
2. SJF	Б) Первый пришел — первый обслужен
3. Round Robin	В) По приоритету процесса
4. Приоритетный	Г) Не изменяется во времени

26. Установить соответствие между термином и определением:

1. Семафор	А) Участок кода с ограниченным доступом
2. Мьютекс	Б) Примитив синхронизации с счетчиком
3. Критическая секция	В) Ситуация взаимного ожидания процессов
4. Взаимная блокировка	Г) Механизм взаимоисключения

27. Установить соответствие между термином и определением:

1. Драйвер устройства	А) Временное хранение данных
2. Буферизация	Б) Программа управления устройством
3. Спулинг	В) Механизм уведомления процессора
4. Прерывание	Г) Организация очереди ввода-вывода

28. Установить соответствие между термином и определением:

1. Аутентификация	А) Проверка прав доступа
2. Авторизация	Б) Проверка личности

3. Шифрование	В) Преобразование данных в защищенный вид
4. Аудит	Г) Регистрация действий

29. Установить соответствие между термином и определением:

1. Однопользовательская	А) Работает с несколькими пользователями
2. Многопользовательская	Б) Обрабатывает задачи с жесткими временными ограничениями
3. Реального времени	В) Использует несколько компьютеров
4. Распределенная	Г) Рассчитана на одного пользователя

30. Установить соответствие между термином и определением:

1. TCP	А) Преобразование доменных имен
2. IP	Б) Надежная передача данных
3. DNS	В) Назначение IP-адресов
4. DHCP	Г) Адресация пакетов

1.4 Практические задания

Практическое задание № 1.

Основы Bash.

Составьте скрипт, который выполняет следующие операции. Прикрепить полученный в итоге архив и скрипт в качестве ответа.

1. Создать каталог с вашей фамилией.
2. Перейти в этот каталог.
3. Сохранить список файлов в домашнем каталоге в файл home_list.txt.
4. Сохранить содержимое каталога /usr/bin/ в файл system_list.txt.
5. Посчитать количество строк в файлах home_list.txt и system_list.txt, сохранить числа в две строки в файл line_count.txt.
6. Выйти из каталога на уровень вверх и создать zip архив созданного каталога.
7. Удалить созданный каталог.

Практическое задание № 2.

Обработка условий в Bash.

Написать скрип, который делает следующее: выводит приветственное сообщение и читает строку с одним числом 1, 2 или 3 и в зависимости от этого запустить соответствующее приложение. Приветственное сообщение должно содержать название запускаемых приложений. Использовать оператор case.

1. Запустить консольный текстовый редактор.
2. Запустить оконный текстовый редактор.
3. Запустить среду разработки.
4. Запустить приложение из офисного пакета.
5. Запустить приложение для просмотра или редактирования графики.

Вариант получить по формуле (Номер в списке-1)%5+1.

Практическое задание № 3.

Групповое переименование файлов.

Написать скрипт, который производит групповое переименование файлов в каталог new_files. Использовать цикл for.

- 1.Переименовать файлы, изменив расширение .html на .txt.
- 2.Переименовать файлы, приведя их имена к верхнему регистру.
- 3.Переименовать файлы, приведя их имена к нижнему регистру.
- 4.Переименовать файлы, добавив номер по порядку к началу каждого файла. Для этого использовать вспомогательную переменную.
- 5.Переименовать файлы, удалив расширение.

Практическое задание № 4.

Написать программу, которая выводит все аргументы, которые были переданы программе при запуске

Написать программу, которая выводит все числа из диапазона [a,b], удовлетворяющие некоторому условию (в соответствии с вариантом). Границы диапазона приложение должно получать из аргументов командной строки:

1. Делится на 33
2. Заканчивается на 5
3. Делится на 55
4. Заканчивается на 44
5. Делится на 15

Дополнить программу так, чтобы она выводила только числа, удовлетворяющие ещё и следующему условию:

1. Содержит цифру 7
2. Является полным квадратом
3. Сумма цифр равна 12
4. Является полным кубом
5. Не содержит цифру 2

В случае, если в качестве единственного аргумента приложению передан символ '-', границы отрезка брать из стандартного ввода

Программа должна выводить сообщение об ошибке при неправильном использовании (с неправильным количеством аргументов).

Вариант для первого задания получить по формуле (Номер в списке-1)%5+1, для второго floor((Номер в списке-1)/5+1).

Практическое задание № 5.

Примените библиотеку PThreads для создания и синхронизации потоков.

Измените программу из задания Обработка аргументов командной строки так, чтобы она использовала для перебора чисел два потока.

Вместо вывода всех найденных чисел, посчитайте их количество и сумму. Для синхронизации используйте pthread_mutex_t.

Реализуйте алгоритм Петерсона и примените его для синхронизации в предыдущей задаче.

В качестве результата выполнения этого задания прикладывайте исходный код всех трёх программ.

Примечание. Для того, чтобы в программе была поддержка потоков, нужно добавить флаг -pthread. Например, чтобы из main.c скопировать программу main, нужно выполнить gcc main.c -o main -pthread.

Практическое задание № 6.

Решить задания на тему "Файловая система". Считывать файл целиком в память запрещено.

Написать программу, которая при запуске создает текстовый файл, доступный для чтения и записи только пользователю и группе. В файл записать свою фамилию и

группу через пробел. Для создания файла использовать функцию open, для записи - write

Написать программу, которая выводит количество байт, в которых отличаются два файла, пути к которым переданы через аргументы командной строки. Использовать функции open и read.

Написать программу, работающую с файлами прямого доступа. В файле хранятся в двоичном формате записи одинаковой структуры. Программа должна иметь возможность по номеру записи читать и изменять запись, а также добавлять новую. Используйте функции open, read, write и lseek. Для текстовых данных используйте массивы символов. Варианты:

Студент: Фамилия, Имя, Отчество, День, Месяц и Год рождения, Факультет, Курс.

Абитуриент: Фамилия, Имя, Отчество, Названия 3 предметов и оценки по ним предметам.

Авиарейс: Номер рейса, Пункт вылета, Пункт прилёта, Тип самолета, Время вылета (часы, минуты).

Поезд: Пункт отправления, Пункт назначения, Номер поезда, Время отправления, Число общих, купейных и плацкартных мест.

Написать программу, работающую с текстовым файлом в формате "ключ=значение". Допустимая длина ключа и значения - по 255 байт. Предусмотреть возможность находить значение по ключу, добавлять новые строки или изменять старые, если они уже существуют. Использовать функции open, read, write. При замене значения использовать временный файл и функцию rename. Сравнить время работы с реализацией, использующей функции из stdin fopen, fscanf, fprintf, fputs и объяснить разницу.

Практическое задание № 7.

Создание и настройка виртуальной машины в Oracle VM VirtualBox.

С рассматриваемым инструментом вы сможете устанавливать и использовать различные операционные системы без их «чистой» установки на ваш компьютер. Основная платформа вашего ПК остается прежней, остальные вы сможете включать или отключать по необходимости.

Практическое задание № 8.

Установка Debian в VirtualBox.

Установка операционной системы Debian на виртуальную машину дает возможность ее использования в качестве дополнительной (к основной) ОС компьютера.

Практическое задание № 9.

Знакомство с операционной системой Linux. Научиться выполнять запуск и вход в систему ОС Линукс. Получить базовые навыки работы в текстовом режиме. Пользоваться встроенной справочной системой.

Практическое задание № 10.

Терминалы и текстовый режим. Научиться основным навыкам работы в многопользовательском режиме, переключению между терминалами и выполнению заданий от имени другого пользователя.

Практическое задание № 11.

Файловая система. Ознакомиться со структурой файловой системы Линукс, типами файлов и командами управления файловой системой.

Практическое задание № 12.

Процессы. Доступ процессов к файловой системе. Получить представление о процессах, как о способе управления ресурсами в Линукс. Научиться получать и

анализировать и анализировать информацию о процессах и управлять состоянием выполняющихся процессов.

Задания:

1. Получить справку о команде ps
 2. Командой ps вывести краткую информацию о выполняющихся процессах в текущем терминале и определить PID текущей оболочки
 3. Получить подробную информацию о загруженных процессах, содержащую имя пользователя, владеющего процессом, процент используемой памяти и использования процессора.
 4. Сохранить список процессов в файл, который затем преобразовать в электронную таблицу.
 5. Выяснить, какой из них использует максимальный объем памяти, а какой - максимально загружает процессор.
 6. Из таблицы, полученной в п.4 выяснить, какой PID имеет процесс init или systemd и от чьего имени он запущен.
 7. Открыть новый терминал и запустить в нем файловый менеджер MS.
 8. Вернуться в исходный терминал и снова просмотреть список процессов. Определить PID процесса mc.
 9. Командой kill снять процессы MS.
 10. Выполнить команду top. Сравнить ее возможности с возможностями ps. Повторить пункт 3 с помощью команды top. Сохранить скриншоты и набранные клавиши для получения в отчет.
 11. Используя top или ps определить, какие процессы порождены (поле PPID) процессом init/systemd (PID=1).
 12. Запустить поиск всех файлов .html от каталога /. Приостановить этот процесс (Ctrl+Z). Запустить команду man bash и приостановить ее выполнение.
 13. Командой jobs определить номера задач, запущенных в п. 15.
 14. Командой fg продолжить выполнение man bash.
 15. Принудительно (kill) завершить команду find.
- В отчете должны быть представлены выполненные команды и скриншоты с использованными клавишами из пункта 10.

Практическое задание № 13.

Системные вызовы fork и exec.

1. Написать программу, которая выводит номер своего процесса, а также номер родительского процесса. Добавить задержку в виде ввода данных или бесконечного цикла, чтобы процесс не завершился мгновенно. Найти его в списке процессов. Завершить его с помощью команды kill.
2. Написать программу, которая порождает дочерний процесс с помощью системного вызова fork(). Каждый процесс выводит номер своего процесса и номер процесса родителя. Добавить в основной процесс задержку, чтобы его номера всегда выводились после номеров дочернего процесса.
3. Написать программу, которая позволяет запускать другие приложения на выбор пользователя (по вариантам: текстовые редакторы, среды разработки, утилиты, приложения офисного пакета, приложения для просмотра и редактирования графики). Для их запуска использовать функцию system.
4. Модифицировать программу так, чтобы можно было вводить и запускать неограниченное количество приложений на выбор пользователя. Заменить system на fork и exec, чтобы программа не ожидала завершения запущенного приложения, а получала следующий ввод сразу же. Объяснить, почему после завершения дочерние процессы находятся в состоянии зомби. Исправить это, добавив игнорирование сигнала SIGCHLD.

Варианты приложений:

1. Запустить оконный текстовый редактор.
2. Запустить среду разработки.
3. Запустить приложение из офисного пакета.
4. Запустить приложение для просмотра или редактирования графики.

Практическое задание № 14.

Управление правами доступа.

Получить навыки по управлению правами доступа к файловым ресурсам системы.

Практическое задание № 15.

Утилиты обработки текста и текстовые редакторы.

Редактирование текстовых файлов одна из наиболее часто выполняемых работ на компьютере. Соответственно и программ для обработки текста разработано множество. Они отличаются друг от друга в той или иной степени, однако есть существенное различие, связанное с двумя типами текстовых файлов, которые могут быть созданы такими программами.

Практическое задание № 16.

Управление файловой системой.

Основными задачами администрирования файловых систем являются создание, подключение (монтирование) и отключение (демонтирование) физических файловых систем, а также поддержание их целостности путем проверки.

Практическое задание № 17.

Управление пакетами.

Ознакомиться с основными принципами комплектации системы в дистрибутивах Linux и научиться выполнять установку, удаление и обновление программного обеспечения.

2. КРИТЕРИИ ОЦЕНИВАНИЯ ПРИ ПРОВЕДЕНИИ ТЕКУЩЕГО КОНТРОЛЯ

Вид контроля	Наименование работы	Наименование оценочных средств	Шкала оценивания
Текущий контроль	<ul style="list-style-type: none"> - Вопросы для обсуждения на занятиях; - Устные опросы по ранее изученному материалу; - Письменные работы: рефераты, тестовые задания; - Практические задания; - Рефераты и доклады по темам (вопросам), вынесенным на самостоятельную работу. 	Оценка выступлений на практическом (семинарском) занятии, проверка заданий и аудиторных работ, устный опрос, оценивание докладов, рефератов	отлично хорошо удовлетворительно неудовлетворительно

Критерии оценивания устных ответов обучающихся

Шкала оценивания	Характеристика оценивания
отлично	оценивается ответ, который показывает прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры;

	свободное владение монологической речью, логичность и последовательность ответа.
хорошо	оценивается ответ, обнаруживающий прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа. Однако допускается одна - две неточности в ответе.
удовлетворительно	оценивается ответ, свидетельствующий в основном о знании процессов изучаемой предметной области, отличающийся недостаточной глубиной и полнотой раскрытия темы; знанием основных вопросов теории; слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры; недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа.
неудовлетворительно	оценивается ответ, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы; незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов; неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Допускаются серьезные ошибки в содержании ответа.

Критерии оценивания работы обучающихся на практических и семинарских занятиях

Шкала оценивания	Показатели	Критерии
Отлично	1. Полнота выполнения практического и тестового задания (полнота ответа); 2. Своевременность выполнения задания; 3. Последовательность и рациональность выполнения практического задания	Задание решено самостоятельно. При этом составлен правильный алгоритм решения задания, в логических рассуждениях, в выборе формул и решении нет ошибок, получен верный ответ, задание решено рациональным способом. Дан правильный и исчерпывающий ответ на поставленные теоретические и тестовые вопросы, в которых обучающийся показал всестороннее системное знание программного материала, усвоение основной и дополнительной литературы, четкое владение понятийным аппаратом.
Хорошо	4. Правильность ответов на вопросы; 5. Самостоятельность решения (владение дополнительным материалом); 6. Знание	Задание решено с помощью преподавателя. При этом составлен правильный алгоритм решения задания, в логическом рассуждении и решении нет существенных ошибок; правильно сделан выбор формул для решения; есть объяснение решения, но задание решено нерациональным способом или допущено не более двух несущественных ошибок, получен верный ответ. На поставленные теоретические и тестовые вопросы, при которых обучающийся показал достаточный уровень знаний основного программного материала: освоение информации лекционного курса и учебных

Шкала оценивания	Показатели	Критерии
	нормативно-законодательной базы и терминологии курса	пособий, овладение понятийным аппаратом, методикой исследований при попытке анализа различных ситуаций.
Удовлетворительно		Задание решено с подсказками преподавателя. Задание решено в общем виде. Обучающийся показал средний уровень знаний основного программного материала, но не мог убедительно аргументировать свой ответ, ошибся в использовании понятийного аппарата, показал недостаточные знания литературных источников.
Неудовлетворительно		Задание не решено. Обучающийся продемонстрировал значительные пробелы в знаниях основного программного материала, не аргументировал свой ответ, показал неудовлетворительные знания понятийного аппарата и специальной литературы.

Критерии оценивания рефератов

Средство контроля	Критерии оценивания	Шкала оценивания
Реферат	Выполнены все требования к написанию и защите реферата: обозначена проблема и обоснована её актуальность, сделан краткий анализ различных точек зрения на рассматриваемую проблему и логично изложена собственная позиция, сформулированы выводы, тема раскрыта полностью, выдержан объём, соблюдены требования к внешнему оформлению, даны правильные ответы на дополнительные вопросы. Реферат раскрывает поднятую проблематику в полном объеме.	отлично
	Основные требования к реферату и его защите выполнены, но при этом допущены недочёты. В частности, имеются неточности в изложении материала; отсутствует логическая последовательность в суждениях; не выдержан объём реферата; имеются упущения в оформлении; на дополнительные вопросы при защите даны неполные ответы. В реферате имеются неточности и предметная область выступления раскрыта не в полной мере.	хорошо
	Имеются существенные отступления от требований к реферированию. В частности: тема освещена лишь частично; допущены фактические ошибки в содержании реферата или при ответе на дополнительные вопросы; во время защиты отсутствует вывод. В реферате не в полной степени раскрыт понятийный аппарат, имеются существенные неточности в процессе формирования выводов.	удовлетворительно
	Тема реферата не раскрыта, обнаруживается существенное непонимание проблемы. Тема реферата не раскрыта или выполнена не по существу ранее поставленного вопроса. Реферат не сдан / доклад не сделан.	неудовлетворительно

Критерии оценивания тестов

Средство контроля	Критерии оценивания – процент положительных ответов	Шкала оценивания
Тестирование	90-100	отлично
	70-89	хорошо
	40-69	удовлетворительно
	< 39	неудовлетворительно

3. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Средства оценивания в ходе промежуточной аттестации:

- вопросы для зачета;
- тестовые задания к зачету.

Коды компетенции	Результаты освоения ОПОП	Перечень планируемых результатов обучения по дисциплине
ОПК-3	Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	<p>ОПК-3.1. Знать: принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.</p> <p>ОПК-3.2. Уметь: решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.</p> <p>ОПК-3.3. Владеть: навыками подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с учетом требований информационной безопасности.</p>
ОПК-8	Способен разрабатывать алгоритмы и программы, пригодные для практического применения	<p>ОПК-8.1. Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения.</p> <p>ОПК-8.2. Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули.</p> <p>ОПК-8.3. Владеть: языком программирования; навыками отладки и</p>

		тестирования работоспособности программы.
--	--	---

3.1. Вопросы к зачету с оценкой

1. Что такое класс и объект в объектно-ориентированном программировании (ООП)?
2. Какие основные принципы ООП вы знаете? Что такое инкапсуляция?
3. Что такое поля и методы класса? В чем их отличия?
4. Для чего используются модификаторы доступа public и private?
5. Что такое конструктор класса и каково его назначение?
6. В чем разница между классом и структурой (struct) в C#?
7. Что такое перечисление (enum) и для чего оно применяется?
8. Как объявить массив объектов? Как обратиться к элементу массива?
9. Что такое список (List<T>) и в чем его преимущество перед обычным массивом?
10. Почему строки в C# (string) являются неизменяемыми (immutable)?
11. Для чего используется класс StringBuilder и в каких случаях он эффективнее string?
12. Что такое наследование классов? Как создать производный класс в C#?
13. В чём разница между модификаторами доступа protected и private при наследовании?
14. Что такое перекрытие метода (override) и для чего нужно ключевое слово virtual?
15. Объясните разницу между перегрузкой (overload) и перекрытием (override) методов.
16. Что такое абстрактный класс? Могут ли у него быть реализованные методы?
17. Для чего применяется ключевое слово sealed при описании класса или метода?
18. Что такое интерфейс в C#? Может ли интерфейс содержать реализацию методов (до C# 8.0 и после)?
19. В чём принципиальное отличие интерфейса от абстрактного класса?
20. Как реализовать множественное наследование в C# с помощью интерфейсов?
21. Что такое свойства (property) в C# и как они обеспечивают инкапсуляцию полей?
22. Для чего нужны индексаторы и как они позволяют обращаться к объектам как к массивам?
23. Что такое делегат в C#? Как он связан с методами обратного вызова?
24. Объясните разницу между анонимными методами и лямбда-выражениями.
25. Что такое событие (event) в C# и как оно связано с делегатами?
26. Для чего используются параметры sender и EventArgs в обработчиках событий?
27. Как создать собственный класс, производный от EventArgs, для передачи данных события?
28. Что такое позднее связывание, ковариантность и контравариантность в контексте делегатов и методов?
29. Как правильно генерировать исключения (throw) и создавать собственные классы исключений для надежного кода?
30. Опишите жизненный цикл события: от его объявления и генерации до динамического связывания с обработчиком и отладки в Visual Studio.

3.2. Задания для зачета:

Задание 1

Определите (в %) размер основного раздела диска, если дополнительный (расширенный) раздел состоит из двух логических разделов размером 30% и 20%. Запустите программу DiskDirectSuite. Разбейте диск на указанные разделы.

Задание 2.

Обработка условий в Bash.

Написать скрипт, который делает следующее: выводит приветственное сообщение и читает строку с одним числом 1, 2 или 3 и в зависимости от этого запускает соответствующее приложение. Приветственное сообщение должно содержать название запускаемых приложений. Использовать оператор case.

6. Запустить консольный текстовый редактор.
 7. Запустить оконный текстовый редактор.
 8. Запустить среду разработки.
 9. Запустить приложение из офисного пакета.
 10. Запустить приложение для просмотра или редактирования графики.
- Вариант получить по формуле $(\text{Номер в списке} - 1) \% 5 + 1$.

Задание 3.

Групповое переименование файлов.

Написать скрипт, который производит групповое переименование файлов в каталог new_files. Использовать цикл for.

6. Переименовать файлы, изменив расширение .html на .txt.
7. Переименовать файлы, приведя их имена к верхнему регистру.
8. Переименовать файлы, приведя их имена к нижнему регистру.
9. Переименовать файлы, добавив номер по порядку к началу каждого файла. Для этого использовать вспомогательную переменную.
10. Переименовать файлы, удалив расширение.

Задание 4.

Написать программу, которая выводит все аргументы, которые были переданы программе при запуске

Написать программу, которая выводит все числа из диапазона [a,b], удовлетворяющие некоторому условию (в соответствии с вариантом). Границы диапазона приложения должны получать из аргументов командной строки:

6. Делится на 33
 7. Заканчивается на 5
 8. Делится на 55
 9. Заканчивается на 44
 10. Делится на 15
- Дополнить программу так, чтобы она выводила только числа, удовлетворяющие ещё и следующему условию:
6. Содержит цифру 7
 7. Является полным квадратом
 8. Сумма цифр равна 12
 9. Является полным кубом
 10. Не содержит цифру 2

В случае, если в качестве единственного аргумента приложению передан символ '-', границы отрезка брать из стандартного ввода

Программа должна выводить сообщение об ошибке при неправильном использовании (с неправильным количеством аргументов).

Вариант для первого задания получить по формуле $(\text{Номер в списке} - 1) \% 5 + 1$, для второго $\text{floor}((\text{Номер в списке} - 1) / 5 + 1)$.

Задание 5.

Примените библиотеку PThreads для создания и синхронизации потоков.

Измените программу из задания Обработка аргументов командной строки так, чтобы она использовала для перебора чисел два потока.

Вместо вывода всех найденных чисел, посчитайте их количество и сумму. Для синхронизации используйте pthread_mutex_t.

Реализуйте алгоритм Петерсона и примените его для синхронизации в предыдущей задаче.

В качестве результата выполнения этого задания прикладываете исходный код всех трёх программ.

Примечание. Для того, чтобы в программе была поддержка потоков, нужно добавить флаг `-pthread`. Например, чтобы из `main.c` скомпилировать программу `main`, нужно выполнить `gcc main.c -o main -pthread`.

Задание 6.

Решить задания на тему "Файловая система". Считывать файл целиком в память запрещено.

Написать программу, которая при запуске создает текстовый файл, доступный для чтения и записи только пользователю и группе. В файл записать свою фамилию и группу через пробел. Для создания файла использовать функцию `open`, для записи - `write`

Написать программу, которая выводит количество байт, в которых отличаются два файла, пути к которым переданы через аргументы командной строки. Использовать функции `open` и `read`.

Написать программу, работающую с файлами прямого доступа. В файле хранятся в двоичном формате записи одинаковой структуры. Программа должна иметь возможность по номеру записи читать и изменять запись, а также добавлять новую. Используйте функции `open`, `read`, `write` и `lseek`. Для текстовых данных используйте массивы символов.

Варианты:

Студент: Фамилия, Имя, Отчество, День, Месяц и Год рождения, Факультет, Курс.

Абитуриент: Фамилия, Имя, Отчество, Названия 3 предметов и оценки по ним предметам.

Авиарейс: Номер рейса, Пункт вылета, Пункт прилёта, Тип самолета, Время вылета (часы, минуты).

Поезд: Пункт отправления, Пункт назначения, Номер поезда, Время отправления, Число общих, купейных и плацкартных мест.

Написать программу, работающую с текстовым файлом в формате "ключ=значение". Допустимая длина ключа и значения - по 255 байт. Предусмотреть возможность находить значение по ключу, добавлять новые строки или изменять старые, если они уже существуют. Использовать функции `open`, `read`, `write`. При замене значения использовать временный файл и функцию `rename`. Сравнить время работы с реализацией, использующей функции из `stdin` `fopen`, `fscanf`, `fprintf`, `fputs` и объяснить разницу.

Задание 7.

Создание и настройка виртуальной машины в Oracle VM VirtualBox.

С рассматриваемым инструментом вы сможете устанавливать и использовать различные операционные системы без их «чистой» установки на ваш компьютер. Основная платформа вашего ПК остается прежней, остальные вы сможете включать или отключать по необходимости.

Задание 8.

Установка Debian в VirtualBox.

Установка операционной системы Debian на виртуальную машину дает возможность ее использования в качестве дополнительной (к основной) ОС компьютера.

Практическое задание № 9.

Знакомство с операционной системой Linux. Научиться выполнять запуск и вход в систему ОС Линукс. Получить базовые навыки работы в текстовом режиме. Пользоваться встроенной справочной системой.

Задание 10.

Терминалы и текстовый режим. Научиться основным навыкам работы в многопользовательском режиме, переключению между терминалами и выполнению заданий от имени другого пользователя.

4. ОСНОВНЫЕ КРИТЕРИИ ОЦЕНИВАНИЯ ПРИ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Шкала оценивания уровня сформированности универсальной компетенций (зачет с оценкой)

Формируемые уровни освоения компетенций	Критерии оценивания	Шкала оценивания
Высокий уровень	Изложено правильное понимание вопроса, четко и самостоятельно дан исчерпывающий ответ, содержание раскрыто полно, профессионально, грамотно. Обучающимся усвоена взаимосвязь основных понятий дисциплины в их значении для приобретаемой профессии. Отражает успешное и систематическое применение навыков и умений по данной дисциплине в соответствии с ФГОС.	отлично
Базовый уровень	Изложено правильное понимание вопроса, дано достаточно подробное описание предмета ответа, приведены и раскрыты в тезисной форме основные понятия, относящиеся к предмету ответа. Ответ отражает полное знание учебно-программного материала, систематический характер знаний по дисциплине, а также наличие базового уровня овладения практическими умениями и навыками по данной дисциплине в соответствии с ФГОС	хорошо
Пороговый уровень	Ответ отражает теоретические знания основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии. Данная оценка может быть выставлена обучающемуся, допустившему неточности в ответе, но обладающими необходимыми знаниями для их устранения под руководством преподавателя, отмечен начальный уровень овладения практическими умениями и навыками по данной дисциплине в соответствии с ФГОС	удовлетворительно
Неудовлетворительный уровень	При ответе обучающегося обнаружено отсутствие знаний, умений и навыков и/или фрагментарные знания основного учебно-программного материала.	неудовлетворительно

Текущий контроль и промежуточная аттестация осуществляются в соответствии с «Положением о текущей и промежуточной аттестации обучающихся в Автономной некоммерческой организации «Образовательная организация высшего образования» «Университет экономики и управления».

Вид промежуточной аттестации – зачет с оценкой.

Форма проведения промежуточной аттестации – письменный зачет.