

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Радиофизический факультет

УТВЕРЖДЕНО:
Декан радиофизического факультета
А.Г. Коротаев

Оценочные материалы по дисциплине

Программирование

по направлению подготовки

03.04.03 Радиофизика, 12.04.03 Фотоника и оптоинформатика

Направленность (профиль) подготовки:
Цифровые технологии фотоники и радиофизики

Форма обучения
Очная

Квалификация
Магистр

Год приема
2025

СОГЛАСОВАНО:
Руководитель ОП
А.П. Коханенко

Председатель УМК
А.П. Коханенко

Томск – 2025

1. Компетенции и индикаторы их достижения, проверяемые данными оценочными материалами

Целью освоения дисциплины является формирование следующих компетенций:

ПК-1 Способен производить анализ состояния научно-технической проблемы, технического задания, формулировать цель и задачи научного исследования в области профессиональной деятельности.

ПК-2 Способен осуществлять построение математических моделей объектов исследования и выбор готового или разработку нового алгоритма решения задачи.

ПК-3 Способен использовать современное оборудование для решения профессиональных задач.

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

РОПК 2.2 Определяет алгоритм и набор параметров, с учётом которых должно быть проведено моделирование устройства или системы.

РОПК 2.3 Проводит компьютерное моделирование устройства или системы.

2. Оценочные материалы текущего контроля и критерии оценивания

Элементы текущего контроля:

- устные опросы.
- практические задания.

Устный опрос (РОПК-2.2)

Устный опрос состоит из 3 теоретических вопросов.

Перечень теоретических вопросов:

1. В чем преимущества использования интегрированных сред разработки?
2. Опишите этапы компиляции программы с помощью gcc.
3. Что такое make-файл, и какова его структура?
4. Какие основные принципы декомпозиции программного обеспечения в UNIX?
5. Какой подход используется для взаимодействия программ через текстовые потоки?
6. Что такое кроссплатформенное программирование и зачем оно нужно?
7. Какие возможности предоставляет Qt для создания кроссплатформенных приложений?
8. Какие основные этапы разработки консольных приложений?
9. Как обрабатываются аргументы командной строки в консольных утилитах?
10. Какие основные элементы включает в себя графический интерфейс приложения на Qt?
11. Что такое сетевые сокеты и какие их основные типы?
12. В чем разница между симплексным, полудуплексным и дуплексным режимами передачи данных?
13. Какие методы взаимодействия процессов существуют в UNIX?
14. Как работает разделяемая память в UNIX?
15. Что такое именованные каналы, и чем они отличаются от неименованных?
16. Какие задачи решаются с использованием объектов синхронизации?
17. Что такое динамически подключаемый модуль? Приведите пример его использования.
18. В чем различие между статическими и динамическими библиотеками?
19. Какие преимущества дает использование автоматизации сборки многофайловых проектов?
20. Что такое IP-адреса и порты? Как они используются в сетевых приложениях?

21. Какой подход рекомендуется для тестирования кроссплатформенного кода?
22. Какие принципы обеспечивают удобство и универсальность пользовательского интерфейса?

Перечень практических заданий:

1. **Создание программы для обработки файлов в формате CSV:** Разработать консольное приложение, которое читает CSV-файл, выполняет базовую фильтрацию строк и сохраняет результаты в новый файл.
2. **Реализация утилиты для управления процессами:** Написать программу, которая отображает список процессов пользователя и позволяет завершать их по PID.
3. **Программа для работы с сигналами:** Реализовать приложение, которое обрабатывает сигналы SIGINT и SIGTERM, корректно завершая свою работу.
4. **Создание make-файла для многофайлового проекта:** Разработать проект из трех файлов (заголовочный файл, библиотека и главный файл) и автоматизировать его сборку с использованием make-файла.
5. **Разработка программы для вычисления контрольной суммы файла:** Написать приложение, которое рассчитывает и отображает MD5-хэш указанного файла.
6. **Утилита для поиска файлов по маске:** Реализовать консольное приложение, которое ищет файлы в указанной директории и поддиректориях по заданной маске.
7. **Создание программы для подсчета строк кода:** Написать утилиту, которая подсчитывает количество строк кода в указанной директории, исключая комментарии и пустые строки.
8. **Реализация консольного текстового редактора:** Разработать минималистичный текстовый редактор для работы в терминале с базовыми функциями редактирования.
9. **Создание многопоточного приложения для подсчета чисел в файлах:** Написать программу, которая параллельно обрабатывает несколько файлов и подсчитывает сумму всех чисел в них.
10. **Реализация простого чата через сокеты:** Написать клиент и сервер для текстового чата на основе TCP-соединений.
11. **Создание утилиты для мониторинга сетевого трафика:** Реализовать программу, которая отображает объем входящего и исходящего трафика для указанного интерфейса.
12. **Реализация программы для взаимодействия процессов через каналы:** Написать два процесса, которые обмениваются данными через именованный канал.
13. **Создание динамически подключаемого модуля:** Написать библиотеку, которая предоставляет функцию вычисления факториала, и использовать её в отдельном приложении.
14. **Создание утилиты для обработки журнала эксперимента:** Написать программу, которая фильтрует строки журнала по заданным ключевым словам.
15. **Реализация программы для вычисления корней квадратного уравнения:** Написать консольное приложение, которое принимает коэффициенты уравнения и возвращает его корни.
16. **Разработка простого файлового менеджера:** Создать консольное приложение с функциями просмотра файлов, копирования и удаления.
17. **Создание GUI-приложения для управления задачами:** Разработать графическое приложение для создания и отслеживания задач с использованием Qt.
18. **Реализация программы для работы с базой данных SQLite:** Написать приложение, которое создает таблицу, добавляет записи и выводит их.
19. **Создание утилиты для синхронизации директорий:** Написать программу, которая синхронизирует содержимое двух директорий.

20. **Разработка программы для визуализации данных:** Реализовать приложение с графическим интерфейсом, отображающее данные в виде графиков.
21. **Создание сетевого файлового сервера:** Написать клиент и сервер для передачи файлов через сеть.
22. **Программа для планирования задач:** Реализовать консольное приложение для создания расписания задач с уведомлениями.
23. **Разработка GUI-калькулятора:** Создать приложение с графическим интерфейсом для выполнения арифметических операций.

Критерии оценивания:

Результаты устного опроса определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Оценка «отлично» выставляется, если даны правильные ответы на все теоретические вопросы.

Оценка «хорошо» выставляется, если не дан ответ на один теоретический вопрос.

Оценка «удовлетворительно» выставляется, если не даны ответы на два теоретических вопроса.

Оценка «неудовлетворительно» выставляется, если не дан ответ ни на один теоретический вопрос.

Результаты выполнения практического задания определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Оценка «отлично» выставляется, если задание выполнено в срок, программа корректно выполняет все указанные в задании функции, продемонстрирована самостоятельность выполнения данного задания, даны пояснения по реализации программы.

Оценка «хорошо» выставляется, если задание выполнено не в срок, либо программа корректно выполняет большинство указанных в задании функций, однако присутствуют не принципиальные недочёты; продемонстрирована самостоятельность выполнения данного задания, даны пояснения по реализации программы.

Оценка «удовлетворительно» выставляется, если задание выполнено не в срок, либо программа корректно выполняет не все указанные в задании функции либо продемонстрирована не полная самостоятельность выполнения данного задания; даны пояснения по реализации программы.

Оценка «неудовлетворительно» выставляется, если задание не выполнено либо выполнено несамостоятельно, либо не даны пояснения по реализации программы.

3. Оценочные материалы итогового контроля (промежуточной аттестации) и критерии оценивания

Промежуточная аттестация по дисциплине проводится в форме письменного экзамена, состоящего из трех вопросов.

Первая часть содержит один вопрос, проверяющий РОПК 2.2. Ответ на вопрос дается в развернутой форме.

Вторая часть содержит один вопрос, проверяющий РОПК 2.3. Ответ на вопрос дается в развернутой форме.

Третья часть содержит один вопрос, проверяющий РОПК 2.2 либо РОПК 2.3. Ответ на вопрос дается в развернутой форме.

В случае прохождения текущего контроля с оценкой «отлично» или «хорошо» студент освобождается от вопроса из первой части. Для допуска к экзамену необходимо выполнение практических заданий на оценку не ниже «удовлетворительно».

Перечень теоретических вопросов:

1. Функции компилятора gcc.
2. Make-файлы: структура и функции.
3. Основные принципы философии написания программ в UNIX.
4. Кроссплатформенное программирование.
5. Средства разработки кроссплатформенного программного обеспечения.
6. Особенности консольных приложений.
7. Сетевые сокеты.
8. Протоколы TCP и UDP.
9. Системный вызов fork().
10. Системный вызов exec().
11. Функция *.c файлов в проекте. Структура и содержание.
12. Функция *.h файлов в проекте. Структура и содержание.
13. Компиляция кода.
14. Интерпретация кода.
15. Средства межпроцессного взаимодействия.
16. Многозадачность и многозадачные приложения.
17. Многопоточность и многопоточные приложения.
18. Объекты синхронизации при межпроцессном взаимодействии.
19. Статические подключаемые модули.
20. Динамические подключаемые модули.
21. Практика применения подключаемых модулей.
22. Интегрированные среды разработки.
23. Типовые графические примитивы и стандартные объекты визуального интерфейса программ.
24. Библиотеки, используемые для работы с графикой.
25. Дуплексный режим передачи данных между приложениями.
26. Передача данных между процессами при помощи именованных каналов?
27. Полудуплексный режим передачи данных между приложениями.
28. Симплексный режим передачи данных между приложениями.

Критерии оценивания:

Результаты зачета определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Оценка «отлично» выставляется, если даны правильные ответы на все теоретические вопросы.

Оценка «хорошо» выставляется, если не дан ответ на один теоретический вопрос.

Оценка «удовлетворительно» выставляется, если не даны ответы на два теоретических вопроса.

Оценка «неудовлетворительно» выставляется, если не дан ответ ни на один теоретический вопрос.

4. Оценочные материалы для проверки остаточных знаний (сформированности компетенций)

Теоретические вопросы:

1. Чем отличается компилятор от интерпретатора?
2. Какую команду нужно использовать для компиляции программы с помощью gcc?
3. Что такое make-файл и для чего он нужен?
4. Какой символ используется для указания комментариев в make-файле?
5. Что такое среда разработки и какие её основные функции?
6. Что означает термин "кроссплатформенность" в программировании?
7. Для чего используется библиотека Qt?
8. Какое преимущество даёт использование библиотек при разработке программ?

9. Как в консольных приложениях обрабатываются аргументы командной строки?
10. Какой файл отвечает за графический интерфейс в Qt-проектах?
11. Что такое сокет и где он используется?
12. Какой протокол используется для потоковой передачи данных через сокеты?
13. Какой протокол используется для передачи датаграмм по сети?
14. Что такое именованный канал в UNIX?
15. В чем основное различие между статическими и динамическими библиотеками?
16. Что такое многопоточное приложение?
17. Чем отличается текстовый интерфейс от графического?

Информация о разработчиках

Булахов Николай Георгиевич, Томский государственный университет, радиофизический факультет, кафедра квантовой электроники и фотоники, старший преподаватель