



Частное профессиональное образовательное учреждение
«Северо-Кавказский межотраслевой колледж»

ИНН/КПП 0600001944/060001001 ОГРН 1220600000455
386101, Республика Ингушетия, г. Назрань, пр-т. Базоркина, д. 3

ПРИНЯТО
на заседании учебно-методического
совета Протокол
от «13» апреля 2026 г. № 3

КОМПЛЕКТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ МДК 01.04 Системное программирование

Специальность: 09.02.07 Информационные системы и программирование

*Тип образовательной программы: Программа подготовки специалистов
среднего звена*

Квалификация: Программист

Форма обучения: очная

Назрань, 2026

Содержание

| | |
|--|----|
| 1. ОБЩИЕ ПОЛОЖЕНИЯ..... | 3 |
| 2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ МДК | 4 |
| 3. ФОРМЫ И МЕТОДЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ | 5 |
| 4. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ ПО МДК.01.04.... | 6 |
| 5. ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ ПО МДК..... | 7 |
| 6. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО МДК | 10 |
| 7. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ | 12 |

1. ОБЩИЕ ПОЛОЖЕНИЯ

Комплект контрольно-оценочных средств (КОС) междисциплинарного курса МДК 01.04 Системное программирование является частью ПМ.01 программы подготовки специалистов среднего звена по специальности 09.02.07 Информационные системы и программирование (квалификация «Программист»).

КОС предназначен для оценки достижения запланированных результатов обучения по междисциплинарному курсу и включает оценочные средства для проведения текущего контроля успеваемости и промежуточной аттестации.

Текущий контроль успеваемости осуществляется в пределах учебного времени, отведённого на изучение МДК. Результаты текущего контроля фиксируются в журнале учебных занятий по пятибалльной системе («5», «4», «3», «2»).

Промежуточная аттестация по МДК проводится в форме дифференцированного зачета. Дифференцированный зачет проводится после завершения освоения МДК в сроки, установленные календарным учебным графиком. Вопросы и задания доводятся до сведения обучающихся в течение первых двух месяцев от начала обучения.

2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ МДК

Общие компетенции (ОК)

| Код | Формулировка компетенции |
|-------|---|
| ОК 01 | Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам |
| ОК 02 | Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности |
| ОК 03 | Планировать и реализовывать собственное профессиональное и личностное развитие |
| ОК 04 | Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами |
| ОК 05 | Осуществлять устную и письменную коммуникацию на государственном языке |
| ОК 09 | Использовать информационные технологии в профессиональной деятельности |
| ОК 10 | Пользоваться профессиональной документацией на государственном и иностранном языках |

Профессиональные компетенции

| Код | Формулировка компетенции |
|--------|---|
| ПК 1.2 | Разрабатывать программные модули в соответствии с техническим заданием |
| ПК 1.3 | Выполнять отладку программных модулей с использованием специализированных программных средств |

В результате освоения МДК обучающийся должен:

Иметь практический опыт:

- разработки системных модулей с использованием API операционной системы;
- использования средств многопоточности, синхронизации и межпроцессного взаимодействия;
- отладки системного кода с использованием профилировщиков и отладчиков.

Уметь:

- создавать и управлять процессами и потоками в ОС Windows;
- организовывать обмен данными между процессами (каналы, сокеты, сообщения);
- программировать сетевые взаимодействия с использованием сокетов (TCP/UDP);
- работать с динамическими библиотеками (DLL) и сервисами;
- выделять и освобождать виртуальную память;
- выполнять низкоуровневый ввод-вывод (работа с буфером экрана).

Знать:

- подсистемы управления ресурсами ОС;
- модели управления процессами и потоками;
- методы синхронизации (мьютексы, семафоры, критические секции, события);
- способы межпроцессного взаимодействия (каналы, очереди сообщений, разделяемая память, сокеты);
- архитектуру сетевых приложений (клиент-сервер);
- принципы работы динамических библиотек и служб Windows;
- управление виртуальной памятью (VirtualAlloc, VirtualFree);
- основы работы с консольным буфером экрана (WriteConsoleOutput).

3. ФОРМЫ И МЕТОДЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ

Формы текущего контроля по МДК:

- устный опрос (фронтальный, индивидуальный, комбинированный);
- тестирование (письменное или компьютерное);
- письменная проверка (ответы на вопросы, решение задач, составление схем, выполнение заданий для самостоятельной работы);
- практическая проверка (при проведении практических и лабораторных занятий);
- самоконтроль и взаимопроверка.

Критерии оценки профессиональных компетенций (фрагмент)

| Компетенция | Критерии оценки | Методы оценки |
|--|--|---|
| ПК 1.2 Разрабатывать программные модули в соответствии с ТЗ | «Отлично» – модуль разработан по алгоритму, полностью соответствует ТЗ, документация оформлена. | Экзамен (практическое задание), защита отчётов по лабораторным работам, наблюдение. |
| ПК 1.3 Выполнять отладку | «Отлично» – выполнена отладка модуля с использованием инструментария среды, пояснены особенности отладочных классов, сохранены результаты отладки. | Дифференцированный зачёт, защита отчётов, наблюдение. |

Оценки по ОК выставляются на основе экспертного наблюдения за выполнением работ.

4. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ ПО МДК.01.04

| № | Наименование темы | Результаты обучения (умения, знания) | ПК, ОК | Текущий контроль успеваемости |
|--|---|--------------------------------------|----------------|-------------------------------|
| Тема 1.4.1. Программирование на языке низкого уровня (управление ресурсами, процессами, потоками, памятью, сокетами) | | | | |
| 1 | Подсистемы управления ресурсами | З1 | ПК 1.2, ОК 01 | Устный опрос, тестирование |
| 2 | Управление процессами. Создание процессов (CreateProcess) | У1, З2 | ПК 1.2 | Лабораторная работа |
| 3 | Управление потоками. Создание потоков (CreateThread, _beginthreadex) | У2, З3 | ПК 1.2 | Лабораторная работа |
| 4 | Параллельная обработка потоков. Синхронизация (мьютексы, критические секции, события) | У3, З4 | ПК 1.2, ПК 1.3 | Лабораторная работа |
| 5 | Обмен данными между процессами (IPC): анонимные и именованные каналы | У4, З5 | ПК 1.2 | Лабораторная работа |
| 6 | Передача сообщений (почтовые ящики, очереди сообщений) | У5, З5 | ПК 1.2 | Устный опрос |
| 7 | Сетевое программирование сокетов (Winsock, TCP/UDP, клиент-сервер) | У6, З6 | ПК 1.2 | Лабораторная работа |
| 8 | Динамически подключаемые библиотеки (DLL): создание, загрузка, использование | У7, З7 | ПК 1.2 | Лабораторная работа |
| 9 | Сервисы Windows (службы): создание, установка, запуск, остановка | У8, З8 | ПК 1.2 | Лабораторная работа |
| 10 | Виртуальная память. Выделение памяти процессам (VirtualAlloc, VirtualFree) | У9, З9 | ПК 1.2 | Лабораторная работа |
| 11 | Работа с буфером экрана (консольный ввод-вывод, WriteConsoleOutput) | У10, З10 | ПК 1.2 | Лабораторная работа |

5. ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ ПО МДК

5.1. Типовые задания для оценки знаний и умений

Тема 1.4.1. Программирование на языке низкого уровня

Вопросы для устного опроса:

1. Что такое процесс и что такое поток? В чём их отличие?
2. Какие основные подсистемы управления ресурсами существуют в ОС?
3. Как создать новый процесс в Windows с помощью API?
4. Какие способы синхронизации потоков вы знаете?
5. Чем отличается мьютекс от критической секции?
6. Что такое семафор? Приведите пример использования.
7. Какие существуют механизмы межпроцессного взаимодействия (IPC)?
8. Как организовать клиент-серверное приложение на сокетах (TCP)?
9. Что такое DLL? Как загрузить функцию из DLL динамически?
10. Как выделить виртуальную память с возможностью исполнения кода?

Тестовые задания:

1. **Какой системный вызов Windows создаёт новый процесс?**
 - a) CreateThread
 - b) CreateProcess
 - c) CreateFile
 - d) CreateMutex

Ответ: b
2. **Какой объект синхронизации может использоваться для организации взаимного исключения между потоками одного процесса?**
 - a) Мьютекс
 - b) Критическая секция
 - c) Семафор
 - d) Событие

Ответ: b
3. **Какой протокол обеспечивает надёжную передачу данных с установкой соединения?**
 - a) UDP
 - b) TCP
 - c) IP
 - d) HTTP

Ответ: b
4. **Какой функцией в Windows создаётся поток?**
 - a) CreateProcess
 - b) CreateThread
 - c) CreateFile
 - d) CreateEvent

Ответ: b
5. **Что такое DLL?**
 - a) Динамически подключаемая библиотека
 - b) Статическая библиотека
 - c) Исполняемый файл
 - d) Драйвер устройства

Ответ: a
6. **Какой механизм IPC позволяет передавать данные между процессами через файловую систему?**
 - a) Сокеты

- b) Именованные каналы (Named Pipes)
- c) Разделяемая память
- d) Очереди сообщений

Ответ: b

7. Какой функцией в Windows выделяется виртуальная память?

- a) malloc
- b) new
- c) VirtualAlloc
- d) HeapAlloc

Ответ: c

8. Какой объект синхронизации может счётчик и позволяет нескольким потокам одновременно входить в защищённый участок?

- a) Мьютекс
- b) Критическая секция
- c) Семафор
- d) Событие

Ответ: c

9. Какой тип сокета используется для протокола TCP?

- a) SOCK_DGRAM
- b) SOCK_STREAM
- c) SOCK_RAW
- d) SOCK_RDM

Ответ: b

10. Что такое «сервис Windows»?

- a) Приложение, запускаемое системой в фоновом режиме без взаимодействия с пользователем
- b) Динамическая библиотека
- c) Графическое приложение
- d) Драйвер устройства

Ответ: a

Лабораторные работы (основные):

1. Создание и завершение процессов с использованием CreateProcess.
2. Создание потоков, передача параметров, ожидание завершения (WaitForSingleObject).
3. Синхронизация потоков: критические секции (EnterCriticalSection), мьютексы (CreateMutex), события (CreateEvent).
4. Организация взаимодействия процессов через анонимные и именованные каналы.
5. Разработка простого TCP-чата (сервер и клиент) с использованием сокетов (Winsock).
6. Создание и использование динамической библиотеки (DLL) с экспортом функций.
7. Создание простой Windows-службы (Service) на C++.
8. Выделение виртуальной памяти (VirtualAlloc) и запись/чтение данных.
9. Прямой вывод в консоль через буфер экрана (консольные графические примитивы).

5.2. Критерии оценивания

5.2.1. Критерии оценивания устного ответа

| Оценка | Характеристика ответа |
|--------|---|
| «5» | Ответ правильный, полный, логически выстроен, литературным языком. |
| «4» | Ответ правильный, полный, но есть отдельные затруднения в формулировке выводов. |

| | |
|-----|---|
| «3» | Ответ в основном правильный, но схематичный или с нарушениями последовательности, неполный. |
| «2» | Непонимание основного содержания, грубые ошибки, отсутствие логики и обобщений. |

5.2.2. Критерии оценивания тестовых заданий

| Оценка | Процент выполнения |
|--------|--------------------|
| «5» | 90–100 % |
| «4» | 70–89,9 % |
| «3» | 50–69,9 % |
| «2» | 0–49,9 % |

5.2.3. Критерии оценивания выполнения заданий на лабораторных и практических занятиях

| Оценка | Характеристика выполнения |
|--------|---|
| «5» | Работа выполнена полностью и правильно, сделаны верные выводы. |
| «4» | Работа выполнена правильно с 1–2 несущественными ошибками, исправленными по требованию преподавателя. |
| «3» | Работа выполнена не менее чем наполовину или допущены 3–4 существенные ошибки. |
| «2» | Допущено 5 и более существенных ошибок, которые обучающийся не может исправить. |

5.2.4. Общая классификация ошибок

Грубые ошибки:

- незнание основных понятий, законов;
- неумение выделить главное, обобщить;
- неумение применить знания для решения задач;
- неумение пользоваться справочной литературой;
- нарушение техники безопасности.

Негрубые ошибки:

- неточность формулировок, определений;
- недостаточно продуманный план ответа;
- нерациональные методы работы с литературой.

6. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО МДК

6.1. Вопросы к дифференцированному зачёту (теоретическая часть)

1. Понятие процесса и потока. Многозадачность и многопоточность.
2. Жизненный цикл процесса: создание, выполнение, завершение.
3. Функция `CreateProcess` и её параметры. Запуск дочернего процесса.
4. Поток: создание (`CreateThread`, `_beginthreadex`), завершение, ожидание.
5. Синхронизация потоков: критические секции (`CRITICAL_SECTION`).
6. Синхронизация с помощью мьютексов (`CreateMutex`, `WaitForSingleObject`, `ReleaseMutex`).
7. Синхронизация с помощью семафоров (`CreateSemaphore`).
8. Синхронизация с помощью событий (`CreateEvent`, `SetEvent`, `ResetEvent`).
9. Межпроцессное взаимодействие (IPC): общая память, каналы, сокет, очереди сообщений.
10. Анонимные и именованные каналы (`Pipe`). Создание и использование.
11. Сетевое программирование на сокетах (`Winsock`). Инициализация, создание сокета (`socket`).
12. TCP-сокеты: установка соединения (`connect`), прослушивание (`listen`), принятие (`accept`).
13. UDP-сокеты: без установки соединения (`sendto`, `recvfrom`).
14. Динамические библиотеки (DLL): создание, экспорт функций, загрузка (`LoadLibrary`, `GetProcAddress`).
15. Службы Windows (`Service`): структура, установка, запуск, остановка.
16. Управление виртуальной памятью: `VirtualAlloc`, `VirtualFree`, `VirtualProtect`.
17. Низкоуровневый консольный ввод-вывод: работа с буфером экрана (`WriteConsoleOutput`, `ReadConsoleInput`).
18. Отладка многопоточных приложений: точки останова, окно потоков, `gace conditions`.
19. Классические проблемы синхронизации: производитель-потребитель, читатели-писатели.
20. Безопасность системного кода: обработка ошибок API, утечки ресурсов.

6.2. Практические задания к дифференцированному зачёту (10 вариантов)

На зачёте студент получает билет, содержащий **один теоретический вопрос** (из списка выше) и **одно практическое задание**. Практическое задание выполняется на языке C/C++ (или C# с использованием P/Invoke, если разрешено) в среде Visual Studio. Время выполнения – 45 минут.

Вариант 1

Написать программу, которая создаёт дочерний процесс (например, `notepad.exe`) с помощью `CreateProcess`. Дождаться его завершения и вывести сообщение.

Вариант 2

Создать два потока, каждый из которых увеличивает общую глобальную переменную 1000 раз. Использовать критическую секцию для синхронизации. Вывести финальное значение (должно быть 2000).

Вариант 3

Реализовать программу «производитель-потребитель» с одним производителем и одним потребителем, использующую событие (`Event`) для синхронизации. Производитель генерирует числа, потребитель их выводит.

Вариант 4

Создать именованный канал (`Named Pipe`) для передачи строки от клиента к серверу. Сервер – отдельный процесс, читает строку и выводит её на экран. Клиент – другой процесс, отправляет строку «Hello, Pipe!».

Вариант 5

Разработать простой TCP-сервер, который принимает соединение, читает строку от клиента, переводит её в верхний регистр и отправляет обратно. Клиент – отдельная программа, отправляет строку и выводит ответ.

Вариант 6

Создать DLL, экспортирующую функцию `int Add(int a, int b)`. Написать основную программу, которая загружает эту DLL динамически (`LoadLibrary`, `GetProcAddress`) и вызывает функцию.

Вариант 7

Написать программу, которая выделяет блок виртуальной памяти размером 4096 байт (`VirtualAlloc`), записывает в него строку, читает её и выводит, затем освобождает память (`VirtualFree`).

Вариант 8

Создать два потока, которые одновременно выводят символы ('A' и 'B') в консоль. Использовать мьютекс, чтобы вывод не перемешивался (каждый поток выводит свой символ 10 раз, после чего передаёт управление другому).

Вариант 9

Реализовать простой UDP-сервер, который принимает сообщения от клиента и отвечает «ОК». Клиент отправляет сообщение и выводит ответ. Показать работу в локальной сети (или на одном компьютере через 127.0.0.1).

Вариант 10

Написать программу, которая с помощью функций работы с буфером экрана (`WriteConsoleOutput`) рисует рамку вокруг окна консоли (символами +, -, |). Размеры рамки – 40x10 символов.

Критерии:

| Оценка | Характеристика |
|---------------------------|---|
| «5» (отлично) | Теоретический ответ полный, без ошибок. Практическое задание выполнено полностью, приложение работает корректно на эмуляторе/устройстве, код соответствует стандартам (именование, структура), студент объясняет свои действия. |
| «4» (хорошо) | Теоретический ответ с незначительными неточностями. Практическое задание выполнено, но есть небольшие недочёты (например, не все обработчики событий написаны оптимально, отсутствует обработка краевых случаев), которые не влияют на основную функциональность. |
| «3» (удовлетворительно) | Теоретический ответ неполный, есть ошибки в основных понятиях. Практическое задание выполнено частично (основной функционал работает, но есть критические ошибки или приложение вылетает) или требует помощи преподавателя. |
| «2» (неудовлетворительно) | Теоретический ответ отсутствует или полностью неверен. Практическое задание не выполнено или не работает. Студент не демонстрирует навыков работы в Android Studio. |

7. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

Основные печатные издания

1. Федорова Г.Н. Разработка модулей программного обеспечения для компьютерных систем: учебник. – М.: Академия, 2020. – 384 с. (раздел по системному программированию)

Основные электронные издания

1. Федорова Г.Н. Разработка модулей программного обеспечения для компьютерных систем: электронный учебно-методический комплекс. – М.: Академия, 2021.

Дополнительные источники

1. Харт Дж. М. Системное программирование в среде Windows. – М.: Вильямс, 2018. – 688 с.
2. Рихтер Дж. Windows via C/C++. – М.: Питер, 2019. – 800 с.
3. Таненбаум Э., Бос Х. Современные операционные системы. – СПб: Питер, 2019. – 1120 с. (теоретические основы)

Интернет-ресурсы

- <https://docs.microsoft.com/ru-ru/windows/win32/> – официальная документация Win32 API
- <https://learn.microsoft.com/ru-ru/windows/win32/sync/synchronization> – синхронизация
- <https://learn.microsoft.com/ru-ru/windows/win32/winsock/windows-sockets-start-page-2> – сокеты (Winsock)