

Дата подписания: 18.06.2026 09:53:01

Уникальный программный ключ:

2539477a8ecf706dc9cfff040c417e6683c4a668

Кафедра «Информационных технологий и систем управления»



УТВЕРЖДАЮ
Директор филиала

А.В. Агафонов

"27" мая 2026г.

Методические рекомендации по подготовке и защите курсовой работы по дисциплине

«Системное программирование»

(наименование дисциплины)

Направление подготовки	09.03.01 «Информатика и вычислительная техника» (код и наименование направления подготовки)
Направленность (профиль) подготовки	«Программное обеспечение вычислительной техники и автоматизированных систем» (наименование профиля подготовки)
Квалификация выпускника	бакалавр
Форма обучения	очная, заочная
Год начала обучения	2026

Чебоксары, 2026

Методические рекомендации по подготовке и защите курсовой работы по дисциплине Системное программирование разработаны в соответствии с:

- Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.01 – Информатика и вычислительная техника, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 929 от 19 сентября 2017 г. зарегистрированный в Минюсте 10 октября 2017 года, рег. номер 48489 (далее – ФГОС ВО).

- учебным планом (очной, заочной форм обучения) по направлению подготовки 09.03.01 «Информатика и вычислительная техника».

- рабочей программой дисциплины «Системное программирование».

Автор Пикина Наталия Евгеньевна, кандидат педагогических наук, доцент кафедры информационных технологий и систем управления

(указать ФИО, ученую степень, ученое звание или должность)

Методические рекомендации одобрены на заседании кафедры Информационных технологий и систем управления (протокол № 9 от 22.05.2026 г.).

В Методических рекомендациях изложены методология и методика подготовки курсовых работ по информатике и вычислительной технике, а также требования к их оформлению; кроме того, определены основные обязанности кафедры Информационных систем и технологий и научных руководителей по руководству, даны рекомендации студентам по их защите.

Методические рекомендации предназначены для руководителей курсовых работ, а также для студентов всех форм обучения, обучающихся по направлению подготовки 09.03.01 «Информатика и вычислительная техника» в Чебоксарском институте (филиале) Московского политехнического университета.

Порядок выбора и утверждения темы курсовой работы

Тема определяется студентом самостоятельно на основании перечней направлений научно-исследовательской деятельности, ежегодно утверждаемых кафедрами, и затем формулируется им в первоначальной редакции.

Одна и та же тема не может выполняться несколькими студентами одной и той же группы. В случае совпадения интересов содержание курсовой работы следует уточнить с преподавателем для того, чтобы обеспечить ее исполнение в разных аспектах.

Тема курсовой работы определяется по первой букве ФАМИЛИИ.

Первая буква фамилии	Темы (на выбор)
А	1, 29, 79
Б	2, 30
В	3, 31, 57, 71
Г	4, 32, 80
Д	5, 33, 58
Е	6, 34, 59
Ж	7, 35
З	8, 36
И	9, 37, 60, 72
К	10, 38, 61, 73
Л	11, 39, 62
М	12, 40, 63, 74
Н	13, 41, 64
О	14, 42, 65, 75
П	15, 43, 66, 76
Р	16, 44, 67
С	17, 45, 68, 77
Т	18, 46, 69, 78
У	19, 47
Ф	20, 48
Х	21, 49
Ц	22, 50
Ч	23, 51
Ш	24, 52
Щ	25, 53
Э	26, 54
Ю	27, 55
Я	28, 56, 70

Тематика курсовых работ

1. Разработать программу на ассемблере для арифметического калькулятора с целыми числами.
2. Разработать программу на ассемблере для перевода чисел между системами счисления.
3. Разработать программу на ассемблере для генерации и вывода простых чисел до N.
4. Разработать программу на ассемблере для вычисления чисел Фибоначчи.
5. Разработать программу на ассемблере для проверки введённого пароля.
6. Разработать программу на ассемблере для шифрования строки сдвигом с помощью шифра Цезаря.
7. Разработать программу на ассемблере для дешифрования строки с помощью шифра Цезаря.
8. Разработать программу на ассемблере для вывода текущего времени с обновлением.
9. Разработать программу на ассемблере для вывода текущей даты.
10. Разработать программу на ассемблере для обратного отсчёта с задержкой.
11. Разработать программу на ассемблере для игры «Угадай число».
12. Разработать программу на ассемблере для игры «Камень, ножницы, бумага».
13. Разработать программу на ассемблере для игры «Быки и коровы».
14. Разработать программу на ассемблере для игры «Виселица».
15. Разработать программу на ассемблере для игры «Крестики-нолики».
16. Разработать программу на ассемблере для игры «Пятнашки».
17. Разработать программу на ассемблере для простого текстового лабиринта.
18. Разработать программу на ассемблере для генерации случайного числа.
19. Разработать программу на ассемблере для симуляции броска игрального кубика.
20. Разработать программу на ассемблере для вывода «ёлочки» из звёздочек.
21. Разработать программу на ассемблере для вывода таблицы умножения.
22. Разработать программу на ассемблере для рисования шахматной доски.
23. Разработать программу на ассемблере для рисования гистограммы по введённым данным.
24. Разработать программу на ассемблере для анимации «бегущая строка».
25. Разработать программу на ассемблере для анимации «падающий снег».
26. Разработать программу на ассемблере для анимации «летающий мяч».
27. Разработать программу на ассемблере для вывода меню с выбором и переходом по пунктам.
28. Разработать программу на ассемблере для теста на знание таблицы умножения.
29. Разработать программу на ассемблере для теста «Столицы стран».
30. Разработать программу на ассемблере для простой анкеты.

31. Разработать программу на ассемблере для подсчёта количества слов во введённой строке.
32. Разработать программу на ассемблере для подсчёта гласных букв в строке.
33. Разработать программу на ассемблере для удаления всех пробелов из строки.
34. Разработать программу на ассемблере для замены всех «а» на «б» в строке.
35. Разработать программу на ассемблере для преобразования строки в верхний регистр.
36. Разработать программу на ассемблере для преобразования строки в нижний регистр.
37. Разработать программу на ассемблере для проверки, является ли строка палиндромом.
38. Разработать программу на ассемблере для поиска самого длинного слова в строке.
39. Разработать программу на ассемблере для вывода строки в обратном порядке.
40. Разработать программу на ассемблере для сложения двух многоразрядных чисел.
41. Разработать программу на ассемблере для вычитания двух многоразрядных чисел.
42. Разработать программу на ассемблере для умножения двух 8-битных чисел без MUL.
43. Разработать программу на ассемблере для деления 16-битного числа на 8-битное без DIV.
44. Разработать программу на ассемблере для нахождения НОД двух чисел.
45. Разработать программу на ассемблере для нахождения НОК двух чисел.
46. Разработать программу на ассемблере для проверки числа на простоту.
47. Разработать программу на ассемблере для разложения числа на простые множители.
48. Разработать программу на ассемблере для вычисления факториала числа.
49. Разработать программу на ассемблере для вычисления степени целого числа.
50. Разработать программу на ассемблере для подсчёта единиц в двоичной записи числа.
51. Разработать программу на ассемблере для вывода числа прописью.
52. Разработать программу на ассемблере для вывода дня недели по номеру.
53. Разработать программу на ассемблере для конвертации температуры.
54. Разработать программу на ассемблере для вычисления площади и периметра прямоугольника.
55. Разработать программу на ассемблере для вычисления длины окружности и площади круга.
56. Разработать программу на ассемблере для решения квадратного уравнения.
57. Разработать программу на ассемблере для нахождения максимального

из трёх чисел.

58. Разработать программу на ассемблере для сортировки трёх чисел по возрастанию.
59. Разработать программу на ассемблере для поиска минимального элемента в массиве.
60. Разработать программу на ассемблере для поиска максимального элемента в массиве.
61. Разработать программу на ассемблере для вычисления суммы и среднего арифметического массива.
62. Разработать программу на ассемблере для копирования массива в обратном порядке.
63. Разработать программу на ассемблере для линейного поиска элемента в массиве.
64. Разработать программу на ассемблере для подсчёта отрицательных чисел в массиве.
65. Разработать программу на ассемблере для удаления дубликатов в отсортированном массиве.
66. Разработать программу на ассемблере для циклического сдвига элементов массива влево.
67. Разработать программу на ассемблере для циклического сдвига элементов массива вправо.
68. Разработать программу на ассемблере для простейшего звукового сигнала.
69. Разработать программу на ассемблере для проигрывания короткой мелодии.
70. Разработать программу на ассемблере для морзянки.
71. Разработать программу на ассемблере для симуляции светофора.
72. Разработать программу на ассемблере для вывода битовой карты 8×8 .
73. Разработать программу на ассемблере для вывода гистограммы из символов «|» по введённым данным.
74. Разработать программу на ассемблере для создания простого пиктограммного меню.
75. Разработать программу на ассемблере для вывода всех ASCII-символов от 32 до 127.
76. Разработать программу на ассемблере для проверки, является ли символ цифрой/буквой/другим.
77. Разработать программу на ассемблере для вывода имени студента в рамке из звёздочек.
78. Разработать программу на ассемблере для симуляции работы банкомата.
79. Разработать программу на ассемблере для простого текстового редактора.
80. Разработать программу на ассемблере для вычисления числа π методом Монте-Карло.

3. Структура и содержание курсовой работы

Курсовая работа должна отвечать следующим требованиям к структуре:

- введение;
- основная часть;
- заключение;
- список использованной литературы.

В работе могут быть приложения.

Во введении должны быть указаны следующие положения:

- актуальность избранной темы и причины (обоснование) ее выбора для подготовки курсовой работы;
- обоснование новизны избранной темы;
- степень исследованности (разработанности) темы в отечественной и зарубежной литературе;
- общий обзор технологий и инструментов, используемых при разработке;
- указание на цели и задачи исследования, предмета, объекта исследования, методов.

В основной части студент излагает собранные им в процессе подготовки курсовой работы материалы – содержание научных обсуждений (дискуссий), имевших место по избранной им теме курсовой работы, обзор существующих системных программных решений и низкоуровневых подходов, относящихся к теме, изложение связанных с темой принципов системного программирования, работы с регистрами процессора, сегментацией памяти, обработкой прерываний, организацией ввода-вывода, управлением стеком, использованием системных вызовов, особенностей архитектуры x86/x86-64 (или другой целевой архитектуры). Обязательным условием является самостоятельность обобщения студентом приведенных материалов и формулирования им выводов по итогам проведенного при подготовке курсовой работы исследования. В случае, если в тексте курсовой работы отражается содержание научных обсуждений (дискуссий) по соответствующей теме, студент должен высказать собственное мнение по предмету научной дискуссии и обосновать его.

В случае, если избранная студентом тема курсовой работы предполагает приведение статистических данных или иных справочных данных, указанные статистические и иные данные должны быть приведены студентом со ссылкой на источник их опубликования.

Целесообразно проведение студентом самостоятельного сбора данных посредством применения таких методов, как проведение опроса (анкетирования) определенного круга лиц с последующим анализом его результатов, самостоятельное обобщение статистики, наполнение базы данных тестовыми данными, приближенными к реальным, анализ качества структуры БД на примере конкретных запросов.

В случае, если возможно выдвижение предложений по совершенствованию разработанного системного программного кода или низкоуровневых алгоритмов, студент по итогам проведения исследования или его части может сформулировать данные предложения в виде конкретных рекомендаций по оптимизации использования регистров, уменьшению размера исполняемого кода, снижению количества тактов процессора, улучшению обработки прерываний, оптимизации работы со стеком и памятью, повышению надёжности и отказоустойчивости программы.

В случае обнаружения недостатков в существующих аналогах баз данных,

неэффективных схем данных, избыточности или противоречивости информации, проблем с производительностью запросов это обстоятельство также может быть отмечено студентом.

В случае обнаружения недостатков в существующих системных аналогах, неэффективном использовании процессорного времени, избыточных копированиях данных, некорректной работе с памятью (утечки, переполнение буфера), неоптимальных прерываниях это обстоятельство также может быть отмечено студентом.

Названия глав курсовой работы не должны повторять название (наименование) курсовой работы, а названия параграфов не должны повторять название главы, частью которой они являются.

В заключении студент должен сформулировать выводы по итогам проведенного исследования, в частности:

- отметить основные проблемы, выявленные и исследованные им в процессе подготовки курсовой работы;

- указать предложенные им способы оптимизации системного кода, методы работы с прерываниями, алгоритмы управления памятью, приёмы эффективного ввода-вывода на ассемблере;

- отметить, по каким направлениям целесообразно продолжать научно-практического исследования по данной тематике.

В списке использованных источников должны быть указаны все использованные студентом при подготовке курсовой работы источники, как нормативные, так и теоретические. При этом для подготовки курсовой работы могут быть использованы источники как на бумажных носителях, так и на электронных носителях, включая использование материалов из различных интернет-ресурсов. Обязательным требованием является непременно указание источника и обозначение авторов теоретических источников (воспринятых студентом как на бумажных носителях, так и на электронных носителях).

Все цитаты должны быть забраны в кавычки, в конце цитаты сделана сноска на использованный источник. Плагиат недопустим ни в каких объемах, даже одно предложение может быть плагиатом.

Порядок оформления курсовой работы

Курсовая работа выполняется на компьютере на стандартных листах А4. Текст печатается на одной стороне листа. На странице должно **располагаться 28-30 строк. Междустрочный интервал – 1,5, шрифт текста – 14 (Times New Roman), в таблицах - 12, в подстрочных сносках -10.** Текст печатается строчными буквами (кроме заглавных), выравнивается по ширине с использованием переносов слов. На титульном листе надпись: курсовая работа печатаются 18 шрифтом. Подчеркивание слов и выделение их курсивом внутри самой работы не допускается. Однако заголовки и подзаголовки при печатании текста письменной работы выделяются полужирным шрифтом. Абзацный отступ должен **соответствовать 1,25 см** и быть одинаковым по всей работе.

Ориентировочный объем курсовой работы составляет **30-40 страниц**. В данный объем не входят приложения и список использованных источников. По согласованию с преподавателем объём работы может быть увеличен.

Страницы, на которых излагается текст, должны иметь поля: **левое -30 мм,**

правое - 10 мм, верхнее - 20 мм, нижнее - 20 мм.

В тексте работы «Введение», название глав, «Заключение» и «Список использованной литературы» печатаются (начинаются) с новой страницы.

Расстояние между заголовком и подзаголовком, заголовком и последующим текстом, подзаголовком и предыдущим текстом отделяют двумя полуторными межстрочными интервалами, а между подзаголовком и последующим текстом - одним полуторным межстрочным интервалом.

Главы письменных работ нумеруются арабскими цифрами и должны начинаться с новой страницы (листа). Номер главы состоит из числа: 1, 2 и т.д.

Заголовки (подзаголовки) располагаются центрированным (посередине текста) способом.

Страницы письменных работ должны иметь сквозную нумерацию арабскими цифрами по всему тексту. Номер страницы проставляют в правом верхнем углу поля страницы без точки в конце. Первой страницей письменной работы является титульный лист. Он не нумеруется. В работе второй страницей является содержание.

Титульный лист должен содержать наименование учебного заведения, формы обучения, обозначение характера работы (курсовая), ее тему, фамилию, имя, отчество выполнившего ее студента, номер курса и группы, ученую степень, должность или ученое звание научного руководителя, его фамилию и инициалы, графы «Дата сдачи», «Допустить к защите», «Дата защиты», «Оценка», место и год написания работы.

Оглавление работы, которое следует после титульного листа, должно содержать названия элементов структуры работы и номера листов, с которых они начинаются.

Используемые в работе стандарты, технические спецификации и программные средства при первом упоминании о них необходимо обозначать полным наименованием с указанием в сноске официального источника (например, ГОСТ, документация разработчика, официальный сайт), а в дальнейшем – по усмотрению студента. Если в дальнейшем студент будет использовать в работе сокращённое наименование, то при первом его упоминании необходимо после указания полного наименования указать также то сокращенное наименование, под которым данный объект будет фигурировать в тексте.

При использовании научно-технической литературы по системному программированию и низкоуровневой разработке и цитировании отдельных положений студент обязан осуществлять в сносках ссылки на авторов и источники, откуда он заимствует материал (фамилия и инициалы автора, название работы, место и год издания, конкретная страница, откуда заимствована цитата). При этом цитирование допускается только в ограниченном объеме, оправданном целью цитирования (для обоснования актуальности рассматриваемого вопроса; демонстрации различных подходов к работе с регистрами процессора, обработке прерываний, управлению памятью, организации ввода-вывода на ассемблере, использованию системных вызовов, существующих в науке по проблемам темы, подтверждения или опровержения выдвигаемых студентом тезисов и т.п.).

Прямое цитирование в тексте обязательно оформляется с помощью кавычек. В случае буквального воспроизведения положений научных трудов без указания на их названия и авторов курсовая работа к защите не допускается.

Материал в списке использованной литературы следует сгруппировать следующим образом:

1. Нормативно-технические документы и стандарты (ГОСТы, ISO, ТУ, руководящие документы, в том числе по системному программированию, архитектуре ЭВМ и программной документации – по значимости или в алфавитном порядке. При этом необходимо указывать полное название документа, дату его принятия и источник официального опубликования).

2. Документация на программные средства и инструментарий (перечисляются используемые инструментальные средства: ассемблеры (MASM, NASM, FASM, GAS), компоновщики (linkers), отладчики (GDB, WinDbg, OllyDbg), дизассемблеры, симуляторы, эмуляторы, интегрированные среды разработки – с указанием версий, лицензий и официальных источников документации. При использовании неопубликованных материалов указываются репозитории, номера коммитов или даты обращения).

3. Научно-техническая литература по системному программированию, архитектуре ЭВМ и смежным дисциплинам (системное программное обеспечение, низкоуровневое программирование, операционные системы, организация ЭВМ) в алфавитном порядке по фамилиям авторов. Ссылки должны содержать фамилию и инициалы автора, основное заглавие, сведения к нему относящиеся, сведения об издании, место издания, издательство, дату издания и объем (наименование периодического издания, год и номер выпуска).

В списке использованных источников должны быть указаны только те материалы, на которые имеется ссылка (сноска) в работе.

Если в курсовой работе имеются приложения, их необходимо пронумеровать.

Все листы курсовой работы должны быть пронумерованы.

Нумерация страниц в курсовой работе должна быть сплошной. Студент отвечает за грамотность и аккуратность оформления курсовой работы.

Наличие грамматических, орфографических и пунктуационных ошибок либо небрежное оформление работы может послужить причиной неудовлетворительной оценки работы.

Подстрочные сноски со ссылками на использованные источники должны иметь сплошную нумерацию.

Порядок представления курсовой работы на защиту

Курсовая работа, подготовленная студентом в окончательной форме, должна быть представлена делопроизводителю кафедры в следующем комплекте:

в письменной форме в прошитом, скрепленном виде – 1 экземпляр;

в электронной форме посредством направления на электронный почтовый адрес кафедры Информационных технологий и систем управления k_itsu@chebpolytech.ru – 1 экземпляр.

Делопроизводитель кафедры после регистрации факта и даты сдачи курсовой работы передает ее для проверки научным руководителем.

Передача курсовой работы в электронной форме может быть осуществлена путем направления ее студентом непосредственно научному руководителю по электронной почте.

После поступления курсовой работы на кафедру научный руководитель проверяет ее в течение 14 календарных дней с момента поступления на кафедру,

после чего возвращает ее делопроизводителю со своим отзывом. В отзыве указываются следующие положения:

- наименование учебного заведения, кафедры, формы обучения;
- обозначение характера работы (курсовая), ее тему;
- фамилию, имя, отчество выполнившего ее студента, номер курса и группы;
- ученую степень, должность или ученое звание научного руководителя, его фамилию и инициалы;
- соответствие представленной курсовой работы общим требованиям, указанным в разделе 1 настоящих Методических рекомендаций;
- соответствие структуры курсовой работы требованиям, указанным в разделе 3 настоящих Методических рекомендаций;
- соответствие оформления курсовой работы требованиям, указанным в разделе 4 настоящих Методических рекомендаций;
- указание на основные выводы и предложения, сформулированные студентом в курсовой работе, при наличии в курсовой работе аргументированных предложений по оптимизации использования регистров процессора, уменьшению размера исполняемого кода, снижению количества тактов процессора, улучшению обработки прерываний, оптимизации работы со стеком и памятью, повышению надёжности низкоуровневых программ, а также выявлению недостатков в существующих системных аналогах или неэффективных низкоуровневых решениях – указать это как достоинство рецензируемой работы;
- указание на имеющиеся в курсовой работе недостатки (как по форме, так и по содержанию работы), не препятствующие допуску работы к защите;
- вывод о возможности допуска курсовой работы к защите;
- вопросы к защите;
- предлагаемая форма и дата защиты курсовой работы (устная (очная или дистанционная)).

В случае если поставленные научным руководителем вопросы не ясны студенту, он вправе уточнить их у научного руководителя лично во время его еженедельных консультаций (дежурств на кафедре) или дистанционно через электронную почту.

В случае формулирования научным руководителем вывода о невозможности допуска курсовой работы к защите курсовая работа подлежит подготовке заново с учетом замечаний, указанных научным руководителем, и повторному представлению на защиту в порядке, предусмотренном разделами 3-5, тому же научному руководителю.

Порядок защиты курсовой работы

Защита курсовой работы может проводиться только научному руководителю.

Защита курсовой работы проводится в форме, установленной научным руководителем.

При устной форме защиты курсовой работы студент должен подготовить ответы на вопросы, поставленные ему научным руководителем в отзыве.

Научный руководитель вправе по своему усмотрению задавать студенту дополнительные вопросы для проверки уровня и качества освоения им знаний по теме курсовой работы, а также для дополнительной проверки самостоятельности выполнения курсовой работы.

По итогам защиты научный руководитель определяет, может ли быть защита зачтена, или требуется повторная защита.

По итогам первоначальной или (в случае ее неудачи) повторной защиты курсовой работы научный руководитель ставит отметку о защите курсовой работы в зачетной книжке студента, в ведомости и на титульном листе работы.

После защиты, отзыв и курсовая работа подлежит сканированию самим студентом и заливке в Электронную информационно-образовательную среду (Электронное портфолио) Чебоксарского института (филиала) Московского политехнического университета по адресу <http://students.polytech21.ru/login.php>, после чего работа в письменной форме передается студентом делопроизводителю для хранения в архиве Филиала.

Перечень основной и дополнительной учебной литературы, необходимой для написания курсовой работы

Основная литература:

1. Трофимов, В. В. Алгоритмизация и программирование : учебник для вузов / В. В. Трофимов, Т. А. Павловская. — 4-е изд. — Москва : Издательство Юрайт, 2025. — 108 с. — (Высшее образование). — ISBN 978-5-534-20430-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/562040>.

2. Советов, Б. Я. Информационные технологии : учебник для вузов / Б. Я. Советов, В. В. Цехановский. — 8-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 414 с. — (Высшее образование). — ISBN 978-5-534-20054-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/582766>.

3. Информационные технологии : учебник для вузов / В. В. Трофимов, О. П. Ильина, В. И. Кияев, Е. В. Трофимова ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2026. — 546 с. — (Высшее образование). — ISBN 978-5-534-18340-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/589572>.

4. Информационные технологии : учебник для вузов / В. В. Трофимов, О. П. Ильина, В. И. Кияев, Е. В. Трофимова ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2026. — 546 с. — (Высшее образование). — ISBN 978-5-534-18340-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/589572>.

Прикладная информатика : журнал / гл. ред. А.А. Емельянов. — Москва : Университет Синергия, 2021. — Режим доступа: по подписке. — URL: https://biblioclub.ru/index.php?page=journal_red&jid=618745 . — ISSN 1993-8314. — Текст : электронный.

Журнал технических исследований : сетевой научный журнал / гл. ред. Н. А. Салькова. — Москва : ИНФРА-М, 2020. — URL: <https://znanium.com/catalog/magazines/issues?ref=6de5e665-cd41-11e8-bfa5-90b11c31de4c>. — Текст : электронный.

Согласовано

Подпись и ФИО завкафедрой

« _____ » _____ 20__ г.

И.о. заведующему кафедрой « _____ »

Студента(ки) группы _____

Форма обучения _____

направления подготовки _____

тел. _____

ФИО студента

Заявление

Прошу утвердить тему курсовой работы

(наименование темы)

по дисциплине _____

(дата)

(подпись)

Тема согласована с научным руководителем _____

(дата)

(подпись)

Кафедра Информационных технологий и систем управления

КУРСОВАЯ РАБОТА
по дисциплине «Системное программирование»

Наименование темы

Рег.номер _____

Выполнил: студент _____ курса, группы _____
кафедры информационных технологий и
систем управления _____ формы
обучения по направлению подготовки

Ф.И.О.

Допущена к защите
«__» _____ 202__ г.

подпись

Научный руководитель:

должность, звание

Ф.И.О.

Защита курсовой работы:

Оценка _____

Дата «__» _____ 202__ г.

Подпись научного руководителя _____

Чебоксары 202__ г.

Пример оформления содержания

Содержание

ВВЕДЕНИЕ.....	Ошибка! Закладка не определена.
ТЕХНИЧЕСКОЕ ЗАДАНИЕ.....	5
1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	8
1.1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ И ИСТОРИЯ ТЕРМИНАЛОВ.....	8
1.2 ОСНОВЫ РАБОТЫ С UART	10
1.3 НИЗКОУРОВНЕВЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ	11
2 ПРАКТИЧЕСКАЯ ЧАСТЬ	Ошибка! Закладка не определена.
2.1 РАЗРАБОТКА ЛОГИКИ РАБОТЫ ПРОГРАММЫ.....	Ошибка! Закладка не определена.
2.2 НАСТРОЙКА СРЕДЫ РАЗРАБОТКИ	15
2.3 РАЗРАБОТКА ПРОГРАММЫ.....	17
2.4 ТЕСТИРОВАНИЕ	23
ЗАКЛЮЧЕНИЕ	Ошибка! Закладка не определена.
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	Ошибка! Закладка не определена.
ПРИЛОЖЕНИЯ.....	Ошибка! Закладка не определена.

Образец написания «Введения» курсовой работы**Введение**

В условиях стремительного развития технологий и увеличения сложности программного обеспечения, создание собственных операционных систем (ОС) становится актуальной задачей для разработчиков. Одним из ключевых аспектов разработки ОС является обеспечение эффективного взаимодействия между разработчиком и ядром системы. В рамках данного курсового проекта будет реализован низкоуровневый терминал с базовыми командами через интерфейс UART, который станет важным инструментом для отладки и взаимодействия с ядром разрабатываемой ОС.

Ядро операционной системы выполняет центральные функции, такие как управление процессами, памятью и устройствами ввода-вывода. Однако для эффективной работы с ядром необходимо иметь удобный интерфейс, который позволит разработчикам отправлять команды, получать информацию о состоянии системы и проводить диагностику. Низкоуровневый терминал через UART предоставляет такую возможность, обеспечивая прямой доступ к функционалу ядра и позволяя осуществлять отладку на ранних этапах разработки.

UART является одним из самых простых и распространенных протоколов последовательной передачи данных, что делает его идеальным выбором для реализации терминала в контексте разработки ОС. Он требует минимального количества проводов и не нуждается в сложной синхронизации, что упрощает проектирование аппаратной части системы. Использование UART позволяет разработчикам сосредоточиться на логике работы ядра и взаимодействии с ним, не отвлекаясь на сложности настройки других протоколов.

Целью данной курсовой работы является создание терминала, который будет служить интерфейсом для отправки команд к ядру ОС и получения ответов от него. Это позволит не только упростить процесс отладки, но и повысить эффективность разработки за счет быстрого тестирования различных функций ядра.

Задачи:

1. Изучить работу с UART.

2. Изучить и настроить программу эмуляции QEMU.
3. Настроить среду разработки.
4. Разработать простой код для загрузки программы в процессор.
5. Разработать функции для работы с UART.
6. Разработать терминал.
7. Протестировать программу.

Таким образом, реализация низкоуровневого терминала через UART является важным шагом в процессе создания собственного ядра операционной системы. Этот инструмент не только облегчит взаимодействие с ядром во время разработки, но и станет основой для дальнейших улучшений и расширений функциональности системы в будущем.

Образец написания «Заключения» курсовой работы

Заключение

В заключение, реализация низкоуровневого терминала через интерфейс UART является важным и необходимым шагом в процессе создания собственной операционной системы. Эта работа не только предоставляет разработчикам удобный инструмент для взаимодействия с ядром, но и открывает новые возможности для отладки и тестирования функциональности системы на ранних этапах разработки.

Низкоуровневый терминал служит связующим звеном между разработчиком и ядром операционной системы. Он позволяет отправлять команды непосредственно в ядро, получать информацию о состоянии системы и проводить диагностику. Это особенно важно в контексте разработки ОС, где ошибки могут быть критичными, и быстрая обратная связь необходима для эффективного устранения проблем.

Были реализованы поставленные задачи:

1. Изучена работа с UART.
2. Изучена и настроена программа эмуляции QEMU.
3. Настроена среда разработки.
4. Разработана простой код для загрузки программы в процессор.
5. Разработана функции для работы с UART.
6. Разработан терминал.
7. Протестирована программа.

Успешная реализация данной работы не только упростит процесс отладки, но также создаст прочную основу для дальнейших улучшений и расширений функциональности операционной системы в будущем. Низкоуровневый терминал может быть дополнен новыми командами, функциями или даже графическим интерфейсом.

В конечном итоге, создание низкоуровневого терминала через UART станет важным вкладом в развитие навыков программирования системного уровня, углубит понимание принципов работы операционных систем и повысит качество разрабатываемого программного обеспечения. Это особенно актуально в условиях стремительного развития технологий, где умение создавать собственные решения

становится все более ценным навыком для современных разработчиков.

Пример оформления списка используемой литературы
Список использованной литературы

1. Архитектура компьютера и системное программирование : программа дисциплины / НИУ ВШЭ. – 2025. – URL: <https://www.hse.ru/edu/courses/1071116738> (дата обращения: 09.12.2025). – Текст : электронный.
2. Бизянов, Е. Е. Системное программирование : учебное пособие / Е. Е. Бизянов. – Москва ; Вологда : Инфра-Инженерия, 2025. – 192 с. – ISBN 978-5-9729-1574-3. – URL: <https://profspo.ru/books/143252> (дата обращения: 09.12.2025). – Текст : электронный.
3. Быстров, А. В. Системное программирование в .NET : учебно-методическое пособие / А. В. Быстров ; Новосиб. гос. ун-т. – Новосибирск : ИПЦ НГУ, 2025. – 118 с. – URL: <http://e-lib.nsu.ru/dsweb/Get/Resource-7497/page00000.pdf> (дата обращения: 09.12.2025). – Текст : электронный.
4. Введение в системное программирование : основы и концепции : – URL: <https://habr.com/ru/companies/otus/articles/830000/> (дата обращения: 09.12.2025). – Текст : электронный.
5. Гаврилов, М. В. Архитектура ЭВМ и системное программное обеспечение : учебник для вузов / М. В. Гаврилов, В. А. Климов. – 6-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2025. – 384 с. – (Высшее образование). – ISBN 978-5-534-20334-9 .
6. Жулабова, Ф. Т. Системное программирование. Лабораторные работы : учебное пособие / Ф. Т. Жулабова. – 3-е изд., стер. – Санкт-Петербург : Лань, 2025. – 208 с. – ISBN 978-5-507-52265-1. – URL: <https://e.lanbook.com/book/445238> (дата обращения: 09.12.2025). – Текст : электронный.
7. Иванов, П. С. Межпроцессное взаимодействие в ОС Linux : сокеты, очереди сообщений, разделяемая память : – URL: <https://blog.skillfactory.ru/ipc-linux/> (дата обращения: 09.12.2025). – Текст : электронный.

8. Куль, Т. П. Операционные системы. Программное обеспечение : учебник / Т. П. Куль. – 5-е изд., стер. – Санкт-Петербург : Лань, 2026. – 248 с. – ISBN 978-5-507-51234-8 .
9. Мельников, Д. А. Системное программирование в среде Linux : учебное пособие / Д. А. Мельников. – Екатеринбург : УрФУ, 2025. – 210 с. – ISBN 978-5-7996-4000-0. – URL: <https://elar.urfu.ru/handle/10995/150300> (дата обращения: 09.12.2025). – Текст : электронный.
10. Низкоуровневое программирование на ассемблере x86-64 : – URL: <https://practicum.yandex.ru/blog/assembly-x86-64/> (дата обращения: 09.12.2025). – Текст : электронный.
11. Операционная система Linux : системные вызовы и работа с ядром : – URL: <https://habr.com/ru/companies/otus/articles/830001/> (дата обращения: 09.12.2025). – Текст : электронный.
12. Отладка системного кода : инструменты gdb, valgrind, strace : – URL: <https://habr.com/ru/articles/815000/> (дата обращения: 09.12.2025). – Текст : электронный.
13. Поток и синхронизация в системном программировании : мьютексы, семафоры, условные переменные : – URL: <https://habr.com/ru/companies/otus/articles/830002/> (дата обращения: 09.12.2025). – Текст : электронный.
14. Процессы в UNIX : создание, завершение, управление : – URL: <https://blog.skillfactory.ru/processes-unix/> (дата обращения: 09.12.2025). – Текст : электронный.
15. Работа с файловой системой на уровне системных вызовов : open, read, write, lseek : – URL: <https://habr.com/ru/articles/815001/> (дата обращения: 09.12.2025). – Текст : электронный.
16. Семакин, И. Г. Системное программное обеспечение : учебник / И. Г. Семакин, А. П. Шестаков. – 5-е изд., стер. – Москва : Академия, 2025. – 272 с. – ISBN 978-5-0054-4046-4 .

17. Сигналы в Linux : обработка, отправка, маскирование : – URL: <https://habr.com/ru/companies/otus/articles/830003/> (дата обращения: 09.12.2025). – Текст : электронный.
18. Системное программирование : основные понятия и определения : – URL: <https://www.osp.ru/os/2025/06/13061260> (дата обращения: 09.12.2025). – Текст : электронный.
19. Системные вызовы в операционных системах : классификация и реализация : – URL: <https://practicum.yandex.ru/blog/system-calls/> (дата обращения: 09.12.2025). – Текст : электронный.
20. Шишов, О. В. Технические средства автоматизации и системное программирование : учебное пособие / О. В. Шишов. – 2-е изд., перераб. – Москва : ИНФРА-М, 2025. – 396 с. – ISBN 978-5-16-015283-7.

ОТЗЫВ на курсовую работу

Студент _____
 Курс _____, группа _____, _____ формы обучения
 Направление подготовки _____
 Направленность (профиль) программы _____
 Дисциплина _____
 Наименование темы _____

Руководитель _____

1. Представленная работа состоит из: введения, _____ глав основной части, заключения и списка использованной литературы _____
2. Оценка качества выполнения курсовой работы

№ п/п	Критерии оценки	Оценка (по 5 - балльной шкале)
2.1.	Актуальность тематики работы	
2.2.	Логичность и структурированность работы	
2.3	Самостоятельность изложения и обобщения материала, интерпретации полученных результатов, обоснованность выводов	
2.4	Использование в работе анализа различных информационных явлений, технических процессов, стандартов и моделей, являющихся объектами профессиональной деятельности в области информационных технологий.	
2.5	Качество проведенного исследования (полнота обзора источников, обоснованность гипотез, выбранных методов исследования и данных для анализа)	
2.6	Результаты работы (новизна, теоретическая и практическая значимость и применимость)	
2.7.	Качество оформления работы (общий уровень грамотности, стиль изложения, качество иллюстраций, соответствие требованиям по оформлению)	
2.8	Использование в работе материалов, специально разработанных для информационных технологий	
2.9	Использование в работе соответствующих направлению исследования источников литературы, нормативных документов, результатов научных исследований и публикаций в сфере информационных технологий.	
Рекомендуемая оценка за работу (не обязательно среднее арифметическое из данных оценок)		

3. Замечания по подготовке и выполнению курсовой работы

4. Курсовая работа соответствует (не соответствует) предъявляемым требованиям, компетенции сформированы (не сформированы), заслуживает (не заслуживает) положительной оценки и может (не может) быть допущена к защите (нужное подчеркнуть)

5. Дополнительные комментарии к работе

« _____ » 202__ г.