Минобрнауки России

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Оренбургский государственный университет имени В.А. Бондаренко»

Кафедра математики и цифровых технологий

РАБОЧАЯ ПРОГРАММА

ДИСЦИПЛИНЫ

«Б1.Д.В.Э.2.2 Низкоуровневое программирование»

Уровень высшего образования

БАКАЛАВРИАТ

Направление подготовки
09.03.03 Прикладная информатика
(код и наименование направления подготовки)

Разработка корпоративных систем и цифровых решений
(наименование направленности (профиля) образовательной программы)

Квалификация
Бакалавр

Форма обучения
Очная

Год набора 2026

Рабочая программа дисциплины *«Б1.Д.В.Э.2.2 Низкоуровневое программирование»* рассмотрена и утверждена на заседании кафедры

Кафедра математики и цифровых технологий
наименование кафедры

протокол № _____ от "___" _____ 20__ г.

Заведующий кафедрой
Кафедра математики и цифровых технологий — подпись — А.Е. Шухман
наименование кафедры *подпись* *расшифровка подписи*

Исполнители:
Доцент — подпись — А.И. Сарайкин
должность *подпись* *расшифровка подписи*

должность *подпись* *расшифровка подписи*

СОГЛАСОВАНО:
Председатель методической комиссии по направлению подготовки
09.03.03 Прикладная информатика — личная подпись — И.П. Болодурина
код наименование *личная подпись* *расшифровка подписи*

Заведующий отделом формирования фонда и научной обработки документов
личная подпись — С.А. Бикжигитова
расшифровка подписи

Уполномоченный по качеству института
личная подпись — С.Н. Морозова
расшифровка подписи

№ регистрации _____

© Сарайкин А.И., 2026
© ОГУ, 2026

1 Цели и задачи освоения дисциплины

Цель (цели) освоения дисциплины: формирование у студентов системного мышления и практических навыков, необходимых для разработки, оптимизации и отладки программного обеспечения, тесно взаимодействующего с аппаратным обеспечением (встраиваемые системы, драйверы, операционные системы).

Задачи:

1. Изучить архитектуру целевых аппаратных платформ (RISC-V, ARM) и её влияние на разработку программного обеспечения.
2. Освоить разработку программного кода на языках C и ассемблера для решения задач системного программирования (драйверы, загрузчики, встраиваемое ПО).
3. Овладеть методами и инструментами отладки системного ПО в средах с операционной системой и без неё (Bare-metal), включая JTAG-отладчики, эмуляторы и логические анализаторы.
4. Научиться оценивать производительность кода и оптимизировать его с учётом ограничений по памяти, времени выполнения и энергопотреблению целевой платформы.
5. Освоить работу с периферийными устройствами на низком уровне через регистры управления, прерывания и механизмы прямого доступа к памяти (DMA).

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к дисциплинам (модулям) по выбору вариативной части блока Д «Дисциплины (модули)»

Пререквизиты дисциплины: *Б1.Д.Б.11 Основы программирования, Б1.Д.Б.17 Операционные системы, Б1.Д.Б.18 Вычислительные системы, сети и телекоммуникации, Б1.Д.В.4 Технологии программирования корпоративных систем*

Постреквизиты дисциплины: *Отсутствуют*

3 Требования к результатам обучения по дисциплине

Процесс изучения дисциплины направлен на формирование следующих результатов обучения

Код и наименование формируемых компетенций	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций
ПК*-13 Способен разрабатывать, оптимизировать и отлаживать системное программное обеспечение	ПК*-13-В-1 Разрабатывает низкоуровневый код для встраиваемых систем и драйверов	**Знать:** архитектуру современных процессоров, особенности работы с оборудованием процессора; принципы взаимодействия ПО с аппаратурой. **Уметь:** разрабатывать низкоуровневый код для встроенного программного обеспечения и драйверов; разрабатывать драйверы для аппаратных устройств.

Код и наименование формируемых компетенций	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций
		Владеть: навыками разработки, оптимизации и отладки низкоуровневого кода для встраиваемых систем и драйверов.

4 Структура и содержание дисциплины

4.1 Структура дисциплины

Общая трудоемкость дисциплины составляет 6 зачетных единиц (216 академических часов).

Вид работы	Трудоемкость, академических часов	
	5 семестр	всего
Общая трудоёмкость	**216**	**216**
Контактная работа:	**105**	**105**
Лекции (Л)	34	34
Лабораторные работы (ЛР)	68	68
Консультации	1	1
Индивидуальная работа и инновационные формы учебных занятий	1,5	1,5
Промежуточная аттестация (зачет, экзамен)	0,5	0,5
Самостоятельная работа: *- выполнение курсового проекта (КП);* *- самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий;* *- изучение разделов курса в системе электронного обучения;* *- подготовка к лабораторным занятиям;* *- подготовка к рубежному контролю).*	**111** +	**111**
Вид итогового контроля (зачет, экзамен, дифференцированный зачет)	**экзамен**	

Разделы дисциплины, изучаемые в 5 семестре

№ раздела	Наименование разделов	Количество часов				
		всего	аудиторная работа			внеауд. работа
			Л	ПЗ	ЛР	
1	Введение в низкоуровневое программирование	30	4		8	18
2	Архитектура процессоров и система команд	36	6		12	18
3	Программирование на языке ассемблера	36	6		12	18
4	Взаимодействие с аппаратным обеспечением	38	6		12	20
5	Разработка драйверов и модулей ядра	40	6		12	22
6	Отладка и оптимизация системного ПО	36	6		12	18
	Итого:	216	34		68	114
	Всего:	216	34		68	114

4.2 Содержание разделов дисциплины

№ 1 Введение в низкоуровневое программирование

Роль низкоуровневого программирования в современных системах. Области применения: встраиваемые системы, драйверы, ОС, гипервизоры. Инструментарий: компиляторы (GCC), ассемблеры, компоновщики (GNU Binutils), эмуляторы (QEMU), отладчики (OpenOCD, GDB). Понятие кросс-компиляции.

№ 2 Архитектура процессоров и система команд

Обзор архитектур RISC-V и ARM. Регистровая модель. Система команд: арифметические, логические, загрузки/сохранения, переходы. Режимы адресации. Стек и вызовы функций. Соглашения о вызовах. Кэш-память и её влияние на производительность.

№ 3 Программирование на языке ассемблера

Синтаксис ассемблера. Директивы ассемблера. Написание простых программ на ассемблере. Взаимодействие C и ассемблера. Встраиваемый ассемблер. Оптимизация критических участков кода.

№ 4 Взаимодействие с аппаратным обеспечением

Memory-mapped I/O. Работа с регистрами периферийных устройств. Прерывания: векторы, обработчики, приоритеты. Прямой доступ к памяти (DMA). Таймеры, GPIO, UART, SPI, I2C. Написание простых драйверов для периферии.

№ 5 Разработка драйверов и модулей ядра

Архитектура драйверов в Linux. Модули ядра: загрузка, выгрузка, параметры. Символьные и блочные устройства. Файловые операции. Работа с прерываниями в ядре. Отладка модулей ядра. Особенности разработки для Astra Linux Special Edition.

№ 6 Отладка и оптимизация системного ПО

Методы отладки: JTAG, логические анализаторы, трассировка. Профилирование кода: gprof, perf, встроенные счётчики. Оптимизация по памяти, времени выполнения и энергопотреблению. Рефакторинг низкоуровневого кода. Тестирование встраиваемых систем.

4.3 Лабораторные работы

№ ЛР	№ раздела	Наименование лабораторных работ	Кол-во часов
1	1	Настройка инструментария и кросс-компиляция	4
2	1	Работа с эмулятором QEMU	4
3	2	Изучение системы команд RISC-V/ARM	6
4	2	Работа с регистрами и памятью	6
5	3	Написание программы на ассемблере	6
6	3	Взаимодействие C и ассемблера	6
7	4	Работа с GPIO и прерываниями	6
8	4	Работа с UART/SPI/I2C	6
9	5	Написание простого модуля ядра	6
10	5	Разработка символьного драйвера	6
11	6	Отладка с использованием JTAG/GDB	6
12	6	Профилирование и оптимизация кода	6
		Итого:	68

4.4 Курсовой проект (5 семестр)

Примерные темы курсового проекта:

1. Разработка драйвера для периферийного устройства (датчик температуры, АЦП, ЦАП) на базе микроконтроллера STM32.
2. Реализация простого загрузчика для отладочной платы на базе RISC-V.
3. Разработка модуля ядра Linux для работы с GPIO.
4. Реализация протокола передачи данных (Modbus, CAN) на низком уровне.
5. Оптимизация алгоритма обработки сигналов для встраиваемой системы.

6. Разработка подсистемы ввода-вывода для промышленного контроллера (по материалам АСУ ПРО).

7. Создание драйвера для работы с DMA на платформе ARM.

8. Исследование производительности кода при различных уровнях оптимизации компилятора.

5 Учебно-методическое обеспечение дисциплины

5.1 Основная литература

1. Таненбаум Э. Современные операционные системы = Modern Operating Systems [Текст] / Э. Таненбаум; [пер. А. Леонтьев]. - 2-е изд. - СПб. : Питер, 2007. - 1038 с. : ил. - (Классика computer science). - Парал. тит. л. англ. - Библиогр.: с. 989-1020. - Алф. указ.: с. 1021-1037. - ISBN 978-5-318-00299-1. - ISBN 5-318-00299-4.

2. Антонов, А. С. Технологии параллельного программирования MPI и OpenMP [Текст] : учебное пособие для студентов высших учебных заведений, обучающихся по направлениям ВПО 010400 "Прикладная математика и информатика" и 010300 "Фундаментальная информатика и информационные технологии" / А. С. Антонов; Моск. гос. ун-т им. М. В. Ломоносова. - Москва : Изд-во Моск. ун-та, 2012. - 340 с. : ил. - (Суперкомпьютерное образование). - Библиогр.: с. 333-334. - ISBN 978-5-211-06343-3.

3. Бройдо, В.Л. Архитектура ЭВМ и систем [Текст] : учеб. для вузов / В. Л. Бройдо, О. П. Ильина. - СПб. : Питер, 2006. - 718 с. : ил. - (Учебник для вузов). - Библиогр.: с. 709-712. - Алф. указ.: с. 713-717. - ISBN 5-469-00742-1. Столяров А. В. Программирование: введение в профессию. – М.: МАКС Пресс, 2020. – 512 с.

5.2 Дополнительная литература

1. Алгоритмы: построение и анализ = Introduction to Algorithms [Текст] / Т. Кормен [и др.]; [пер. с англ. И. В. Красикова, Н. А. Ореховой, В. Н. Романова; под ред. И. В. Красикова].- 2-е изд. - Москва ; Санкт-Петербург ; Киев : Вильямс, 2013. - 1296 с. : ил. - Парал. тит. л. англ. - Прил.: с. 1189-1256. - Библиогр.: с. 1257-1276. - Предм. указ.: с. 1277-1290. - ISBN 978-5-8459-0857-5. - ISBN 0-07-013151-1.

2. Гордеев, А. В. Операционные системы [Текст] : учебник / А. В. Гордеев.- 2-е изд. - Санкт Петербург : Питер, 2007. - 416 с. - (Учебник для вузов). - Библиогр.: с. 406-408. - Алф. указ.: с. 409-415. - ISBN 978-5-94723-632-3.

3. Митчелл, М. Программирование для Linux. Профессиональный подход = Advanced Linux Programming [Текст] / М. Митчелл, Д. Оулдем, А. Самьюэл; пер. с англ. и ред. В. Р. Гинзбурга. - М. ; СПб. ; Киев : Вильямс, 2003. - 288 с. : ил. - Парал. тит. л. англ. - ISBN 5-8459-0243-6.

5.3 Периодические издания

1. Программирование (ИПМ им. М.В. Келдыша РАН): журнал. - Москва: издательство «Наука».

2. Вестник компьютерных и информационных технологий: журнал. - Москва: ИД «Спектр».

3. Информационные технологии: журнал. - Москва: Агентство «Роспечать».

5.4 Интернет-ресурсы

− www.elibrary.ru - научная электронная библиотека. Журналы, книги, патенты;

− SCOPUS [Электронный ресурс]: реферативная база данных / компания Elsevier. – Режим доступа: https://www.scopus.com/, в локальной сети ОГУ.

5.5 Программное обеспечение, профессиональные базы данных и информационные справочные системы

1. Операционная система РЕД ОС.
2. Пакет офисных приложений «МойОфис Образование».
3. Университетская платформа электронного обучения «Электронные курсы ОГУ в системе обучения Moodle» (http://moodle.osu.ru).
4. Для работы с ресурсами Интернет - веб-браузер Яндекс https://yandex.ru/.
5. ГАРАНТ Платформа F1 [Электронный ресурс]: справочно-правовая система. / Разработчик ООО НПП «ГАРАНТ-Сервис», 119992, Москва, Воробьевы горы, МГУ, [1990–2026]. – Режим доступа в сети ОГУ http://garant.net.osu.ru
6. Автоматизированная интерактивная система сетевого тестирования - АИССТ (зарегистрирована в РОСПАТЕНТ, Свидетельство о государственной регистрации программы для ЭВМ №2011610456, правообладатель – Оренбургский государственный университет), режим доступа - http://aist.osu.ru.
7. Набор свободных компиляторов для различных языков программирования GCC (GNU Compiler Collection).
8. Набор свободных программ и инструментов командной строки для создания, анализа, изменения и отладки двоичных и объектных файлов – GNU Binutils.
9. QEMU – свободная программа с открытым исходным кодом для эмуляции аппаратного обеспечения и виртуализации.
10. OpenOCD – программа с открытым исходным кодом для внутрисхемной отладки, программирования и тестирования микроконтроллеров, процессоров и FPGA.
11. Бесплатные инструменты для написания и редактирования компьютерного кода – VS Code, Eclipse.
12. Git – свободная и бесплатная распределённая система управления версиями с открытым исходным кодом.

6 Материально-техническое обеспечение дисциплины

Учебные аудитории для проведения занятий лекционного типа, семинарского типа, курсового проектирования, для проведения групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

Аудитории оснащены комплектами ученической мебели, техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Для проведения лабораторных занятий используется компьютерный класс оснащенный:

- персональными компьютерами с возможностью запуска виртуальных машин и эмуляторов;
- отладочными платами на базе архитектур ARM и RISC-V;
- JTAG-отладчиками;
- логическими анализаторами;
- набором периферийных модулей (датчики, исполнительные механизмы, дисплеи).

Помещение для самостоятельной работы обучающихся оснащены компьютерной техникой, подключенной к сети "Интернет", и обеспечением доступа в электронную информационно-образовательную среду ОГУ.