

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Беспалов Владимир Александрович

Должность: Ректор МИЭТ

Дата подписания: 12.10.2023 16:06:37

Уникальный программный ключ:

ef5a4fe6ed0ffdf3f1a49d6ad1b49464dc1bf7354f736d76c8f8bea882b8d602

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет

«Московский институт электронной техники»

УТВЕРЖДАЮ

Проректор по учебной работе

А.Г. Балашов

«18» 05 2023 г.



## РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

### «Программное обеспечение сетевых устройств»

Направление подготовки – 02.03.01 «Математика и компьютерные науки»  
Направленность (профиль) – «Компьютерная математика и математическое моделирование»

## 1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ

Дисциплина участвует в формировании следующих компетенций образовательных программ:

**Компетенция ПК-4** Способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных, управляющих и вычислительных систем» сформулирована на основе профессионального стандарта 06.001 «Программист»

**Обобщенная трудовая функция «D** Разработка требований и проектирование программного обеспечения»

**Трудовые функции: «D/03.6** Проектирование программного обеспечения»

<b>Подкомпетенции, формируемые в дисциплине</b>	<b>Задачи профессиональной деятельности</b>	<b>Индикаторы достижения подкомпетенций</b>
<b>ПК-4.ПОСУ.</b> Способен разрабатывать комплексы программных компонентов сетевых устройств.	Разработка и отладка компонентов сетевого программного обеспечения встраиваемых устройств на основе операционной системы Linux	<b>Знает</b> внутреннюю архитектуру сетевых приложений, методы взаимодействия между сетевыми приложениями, принципы организации системы управления на основе конфигурационной базы данных и протоколов дистанционного управления сетевыми устройствами. <b>Умеет</b> разрабатывать программные компоненты сетевых устройств, которые обеспечивают динамическое управление аппаратными блоками сетевого устройства, осуществляют взаимодействие с другими программными компонентами для совместного решения прикладных задач, а также поддерживают автоматическое управление за счет использования конфигурационной базы данных. <b>Имеет опыт</b> использования внутренних и внешних программных интерфейсов ядра Linux, средств сетевого межпроцессного взаимодействия и типовых структурных элементов сетевых приложений, навыки использования среды сборки и исполнения встроенных сетевых устройств.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина входит в часть, формируемую участниками образовательных отношений Блока ФТД. «Факультативы».

Для изучения дисциплины студент должен владеть основами дискретной математики и программирования на языке С.

## 3. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Курс	Семестр	Общая трудоёмкость (ЗЕ)	Общая трудоёмкость (часы)	Контактная работа			Самостоятельная работа (часы)	Промежуточная аттестация
				Лекции (часы)	Лабораторные работы (часы)	Практические занятия (часы)		
2	4	3	108	32	32	-	44	ЗаО

## 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

№ и наименование модуля	Контактная работа			Самостоятельная работа (часы)	Формы текущего контроля
	Лекции (часы)	Практические занятия (часы)	Лабораторные занятия (часы)		
1. Программное окружение ОС Linux во встроенных системах	8	8	-	10	Защита заданий в рамках лабораторных работ (базовых и дополнительных, рассчитанных на самостоятельную работу)
2. Компоненты программного обеспечения сетевых устройств	14	14	-	20	Защита заданий в рамках лабораторных работ (базовых и дополнительных, рассчитанных на самостоятельную работу)



3. Система управления сетевым устройством	10	10	-	14	Защита заданий в рамках лабораторных работ (базовых и дополнительных, рассчитанных на самостоятельную работу)
---	----	----	---	----	---

#### 4.1. Лекционные занятия

№ модуля дисциплины	№ лекции	Объем занятий (часы)	Краткое содержание
1	1	2	<b>Введение в сетевое ПО. Среда разработки приложений для встроенных систем.</b> Архитектура и функции сетевого устройства. Отличительные особенности сетевого ПО. Кросс-компиляция. Система сборки Automake.
	2	2	<b>Среда исполнения приложений в ОС Linux.</b> Программное окружение процессов в ОС Linux. Файлы и файловые дескрипторы. Командная оболочка ОС Linux. Сетевые интерфейсы.
	3	2	<b>Обмен данными между приложением и ядром операционной системы.</b> Типы файлов. Служебные файлы символьных и блочных устройств. Типы файловых систем. Виртуальные файловые системы proc, sysfs и tmpfs. Подсистема сигналов ввода-вывода общего назначения (GPIO - General Purpose Input/Output).
	4	2	<b>Обмен данными между приложениями с помощью BSD-сокетов.</b> Классификация BSD-сокетов. Адресация процессов в операционной системе и глобальной сети. Поточковая и пакетная передача данных. Поддержание соединения между процессами.
2	5	2	<b>Многозадачность и обработка событий.</b> Проблема многозадачности. Одновременность и параллелизм. Многопоточность и ее свойства. Системные вызовы select и poll. Реализация таймеров с помощью вызова select.
	6	2	<b>Внутренняя структура сетевого приложения.</b> Цикл обработки событий. Конечные автоматы. Двоичные деревья. Списки и словари.
	7	2	<b>Передача данных между сетевыми приложениями через IP-сеть.</b> Коммутация и маршрутизация пакетов в компьютерных сетях. Протокол Ethernet. Алгоритм изучения MAC-адресов.
	8	2	<b>Управление периферийными устройствами.</b>

			Драйверы и устройства в ядре Linux. Представление периферийных устройств в ядре Linux. Обращение к периферийным устройствам через аппаратные регистры. Обработка аппаратных прерываний. Описание структуры встроенной системы с помощью Device Tree.
	9	2	<b>Программные интерфейсы ядра Linux.</b> Доступ к общим ресурсам ядра. Подсистема сигналов ввода-вывода gpiolib. Программный интерфейс символьного устройства.
	10	2	<b>Унифицированное представление периферийных устройств и расширяемые программные интерфейсы ядра.</b> Проблема управления разнородными периферийными устройствами. Команды ввода-вывода и системный вызов ioctl. Исполнение отложенных задач в ядре Linux.
	11	2	<b>Управление ядром Linux с помощью протокола Netlink.</b> Назначение и свойства протокола Netlink. Структура сообщения и атрибуты. Типы запросов Netlink. Управление сетевым стеком ядра.
3	12	2	<b>Автоматическое управление сетевыми приложениями и конфигурационные базы данных.</b> Архитектура системы автоматического управления сетевым устройством. Унифицированное представление конфигурации и статуса устройства в форматах XML и JSON. Язык запросов JSONPath.
	13	2	<b>Управление сетевым приложением по схеме «клиент-сервер».</b> Удаленные вызовы функций (RPC - Remote Procedure Call). Протокол JSON-RPC. Пакетные сокеты с установкой соединения.
	14	2	<b>Управление сетевым приложением по схеме «подписка-рассылка».</b> Обмен сообщениями в сетях с динамической топологией. Очереди сообщений. Регистрация подписки на брокере сообщений. Рассылка автономных уведомлений.
	15	2	<b>Протоколы удаленного управления сетевыми устройствами.</b> Автоматическое и интерактивное управление сетевыми устройствами. Сетевой интерфейс командной строки (CLI). Установка соединений через потоковые сокеты.
	16	2	<b>Удаленное управление сетевыми устройствами в сети Интернет.</b> Доступ к сетевым ресурсам через протокол HTTP. Автоматическое управление сетевыми устройствами по протоколу RESTCONF. Интеграция HTTP-сервера в комплекс сетевого ПО. Доступ к конфигурационной базе данных через протокол CGI.

#### 4.2. Практические занятия

Не предусмотрены

### 4.3. Лабораторные занятия

№ модуля дисциплины	№ лабораторной работы	Объем занятий (часы)	Наименование работы
1	1	2	Разработка консольного приложения для встроенной ОС Linux, реализующего последовательность арифметических операций, заданную в командной строке.
	2	2	Разработка приложения для вывода статуса сетевого интерфейса.
	3	2	Разработка приложения, повторяющего состояние входного сигнала GPIO на выходном сигнале GPIO.
	4	2	Разработка клиентского приложения для учебного протокола передачи строковых значений.
2	5	2	Разработка сервера мультиплексирования текстовых потоков.
	6	2	Разработка приложения для динамического добавления и извлечения строк текста из встроенных списка и словаря.
	7	2	Разработка двухпортового повторителя для Ethernet-кадров.
	8	2	Разработка драйвера аппаратного таймера, управляемого через атрибуты файловой системы sysfs.
	9	2	Разработка драйвера аппаратного таймера, управляемого через символическое устройство.
	10	2	Разработка драйвера для управления сигналами GPIO через команды ioctl.
	11	2	Разработка приложения для установки и считывания состояния сигналов GPIO через протокол Netlink.
3	12	2	Разработка приложения, исполняющего запросы на чтение данных из документа в формате JSON.
	13	2	Разработка клиентского приложения для конфигурационной СУБД.
	14	2	Разработка приложения для отслеживания изменений в конфигурационной СУБД.
	15	2	Разработка сетевого CLI-интерфейса для установки и считывания состояния сигналов GPIO.
	16	2	Разработка транслятора HTTP-запросов в запросы к конфигурационной СУБД.



#### 4.4. Самостоятельная работа студентов

№ модуля дисциплины	Объем занятий (часы)	Вид СРС
1	10	<p>Решение дополнительных задач к лабораторным работам.</p> <p>В дополнение к лабораторной работе 1 необходимо реализовать операции умножения и деления. Все арифметические действия выполняются в таком же порядке, как в языке С.</p> <p>В дополнение к лабораторной работе 2 реализовать конфигурирование сетевого интерфейса. Наборы параметров конфигурации и статуса совпадают. Задание конфигурации осуществляется с помощью пакета iproute2.</p> <p>Решить задачу к лабораторной работе 3 с конфигурируемой задержкой воспроизведения.</p> <p>Решить задачу к лабораторной работе 4 с использованием потоковых Unix-сокетов. Приложение должно восстанавливать целый ответ от сервера из отдельных фрагментов, а также детектировать разрыв соединения с сервером.</p>
2	20	<p>Решение дополнительных задач к лабораторным работам.</p> <p>Разработать сервер мультиплексирования текстовых потоков. Входные потоки принимаются из серверных Unix-сокетов со скоростью не больше 1 строки в секунду для каждого сокета. Мультиплексированный поток выводится через специальный клиентский сокет. При отсутствии места в клиентском сокете приложение буферизирует поток и задерживает его воспроизведение.</p> <p>В дополнение к лабораторной работе 6 реализовать автоматическое удаление устаревших элементов контейнеров. В момент добавления элемента приложение должно устанавливать таймер, при срабатывании которого элемент автоматически удаляется. Если до срабатывания таймера такой же элемент добавляется повторно, время срабатывания таймера откладываетеся.</p> <p>Разработать трехпортовый Ethernet-коммутатор с изучением MAC-адресов. Прием и передача кадров осуществляется через сокеты семейства AF_PACKET. MAC-адрес источника из каждого принятого кадра используется для модификации таблицы адресов. Устаревшие MAC-адреса должны автоматически удаляться из таблицы. Выбор множества выходных портов осуществляется по адресу назначения в соответствии с текущей таблицей адресов.</p> <p>В дополнение к лабораторной работе 8 реализовать обработку аппаратного прерывания, которое таймер генерирует при пересечении программируемого порога. В sysfs должны быть реализованы 2 дополнительных атрибута: один задает значение порога, второй - показывает общее количество</p>

		<p>обработанных прерываний.</p> <p>Разработать драйвер для установки и считывания состояния сигналов GPIO с управлением через набор символьных устройств. Доступные драйверу сигналы GPIO извлекаются из структуры Device Tree. Установка и считывание состояний выводов осуществляется через подсистему gpiolib. Каждый сигнал управляется через отдельное символьное устройство. Запись в устройство устанавливает значение выходного сигнала. Чтение возвращает значение входного сигнала.</p> <p>В дополнение к лабораторной работе 10 реализовать команду для перевода выходного сигнала в состояние периодического мигания с настраиваемым периодом. Мигание должно быть реализовано с помощью механизма отложенной работы ядра Linux.</p> <p>В дополнение к лабораторной работе 11 разработать приложение для мониторинга изменений GPIO через Netlink. Мониторинг осуществляется с помощью подписки на соответствующую группу многоадресной рассылки и чтения уведомлений из Netlink-сокета. Новое состояние GPIO выводится в терминал.</p>
3	14	<p>Решение дополнительных задач к лабораторным работам.</p> <p>В дополнение к лабораторной работе 12 поддержать упрощенные предикаты JSONPath (оператор []). Выражение предиката выполняет сравнение ближайшего дочернего элемента с указанным строковым значением.</p> <p>В дополнение к лабораторной работе 13 реализовать применение настроек с помощью протокола Netlink.</p> <p>В дополнение к лабораторной работе 14 необходимо реализовать применение конфигурации и обновление статуса сетевых интерфейсов. Конфигурация сетевых интерфейсов, полученная в уведомлении от СУБД, применяется с помощью протокола Netlink. Протокол Netlink также используется для мониторинга состояния сетевых интерфейсов. В случае его изменения новый статус записывается в СУБД.</p> <p>В дополнение к лабораторной работе 15 реализовать обмен с СУБД для записи и чтения состояния сигналов GPIO через JSON-RPC.</p> <p>В дополнение к лабораторной работе 16 реализовать поддержку произвольных идентификаторов URI в соответствии со стандартом RESTCONF.</p>

#### 4.5. Примерная тематика курсовых работ (проектов)

Не предусмотрены

### 5. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Учебно-методическое обеспечение для самостоятельной работы студентов в составе УМК дисциплины.

- ✓ Методические указания студентам по изучению дисциплины
- ✓ Материалы для подготовки к выполнению и защите лабораторных работ.



1. Документация на команды и программные интерфейсы операционной системы Linux  
<http://www.opennet/man.shtml>
2. Самоучитель по операционной системе Linux  
<https://studylinux.ru>
3. Герберт Шилдт. Полный справочник по C  
[http://cpp.com.ru/shildt\\_spr\\_po\\_c/index.html](http://cpp.com.ru/shildt_spr_po_c/index.html)
4. Мамаев М.А. Телекоммуникационные технологии: сети TCP/IP  
[https://abc.vvsu.ru/books/tcp\\_ip/](https://abc.vvsu.ru/books/tcp_ip/)
5. Исходный код библиотеки libubox  
<https://git.openwrt.org/project/libubox.git>
6. Исходный код библиотеки jansson  
<https://github.com/akheron/jansson>

## **6. ПЕРЕЧЕНЬ УЧЕБНОЙ ЛИТЕРАТУРЫ**

### **Литература**

1. Керниган Б.В. Язык программирования C / Б.В. Керниган, Д.М. Ритчи. - 2-е изд. - М. : ИНТУИТ.РУ, 2016. - 313 с. - URL: <https://e.lanbook.com/book/100543> (дата обращения: 20.03.2023).
2. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы : Учебник для вузов / В.Г. Олифер, Н.А. Олифер. - 3-е изд. - СПб. : Питер, 2006. - 958 с. - (Учебник для вузов). - ISBN 5-469-00504-6
3. Таненбаум Э. Современные операционные системы : Пер. с англ. / Э. Таненбаум. - 2-е изд. - СПб. : Питер, 2005. - 1038 с. - (Классика Computer Science). - ISBN 5-318-00299-4

## **7. ПЕРЕЧЕНЬ ПРОФЕССИОНАЛЬНЫХ БАЗ ДАННЫХ, ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ**

1. Лань : Электронно-библиотечная система Издательства Лань. - СПб., 2011-. - URL: <https://e.lanbook.com> (дата обращения: 20.03.2023). - Режим доступа: для авторизированных пользователей МИЭТ
2. GitHub : сайт. – На англ. языке. - URL: <https://github.com/> (дата обращения: 20.03.2023).

## **8. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ**

Процесс изучения дисциплины включает следующие виды деятельности.

1. Посещение аудиторных занятий и консультаций преподавателя.
2. Работу по лекционному материалу с подготовкой к практическим занятиям.
3. Выполнение лабораторных работ.

4. Выполнение в полном объеме контрольных мероприятий (защиту лабораторных работ и зачет).

5. Самостоятельную работу, предполагающую изучение рекомендуемой литературы.

Лекционные занятия проводятся в традиционной форме с использованием мультимедийных презентаций. На каждой лекции студенты должны составить краткий конспект по теме лекции. При изучении теоретических материалов необходимо обращать внимание на обоснование выбора рассматриваемых технических решений, их достоинства и недостатки.

Лабораторные работы состоят из подготовительного этапа и защиты. На подготовительном этапе студенту требуется разработать компонент (приложение или драйвер) сетевого программного обеспечения в соответствии с заданием и рекомендациями преподавателя. В каждой работе студенту предлагается на выбор простое и усложненное задание. Студент выполняет только один вариант задания. Перед выполнением работы необходимо изучить материалы лекций, образцы программного кода, предоставленные преподавателем, а также документацию на программные интерфейсы ОС Linux, относящиеся к теме работы. Разработка приложений и драйверов проходит в компьютерной аудитории или дома.

Результатом разработки является программный модуль, скомпилированный для исполнения макете реального сетевого устройства, которым оборудована компьютерная аудитория. Защита лабораторной работы включает демонстрацию работы программного модуля на макете сетевого устройства, предъявление программного кода преподавателю и ответы на вопросы преподавателя по коду программы и по теме соответствующей лекции. На защите студент обосновать выбор алгоритм программы и средств его реализации.

## 9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность учебных аудиторий и помещений для самостоятельной работы	Перечень программного обеспечения
Учебная аудитория	Учебная доска Мультимедийное оборудование (компьютер с ПО и возможностью подключения к сети Интернет и обеспечением доступа в электронно-образовательную среду МИЭТ)	Операционная система Microsoft Windows от 7 версии и выше, Microsoft Office Professional Plus или Open Office, браузер (Firefox, Google Chrome);  Acrobat reader DC GCC, Python, WireShark, Virtual Box, ОС Debian, VS CODE.
Компьютерный класс	Компьютерная техника с возможностью подключения к сети «Интернет» и обеспе-	GCC, Python, WireShark, Virtual Box, ОС Debian, VS CODE.

	чением доступа в электронную информационно-образовательную среду МИЭТ	
Помещение для самостоятельной работы обучающихся	Компьютерная техника с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду МИЭТ	Операционная система Microsoft Windows от 7 версии и выше, Microsoft Office Professional Plus или Open Office, браузер (Firefox, Google Chrome);  Acrobat reader DC GCC, Python, WireShark, Virtual Box, ОС Debian, VS CODE.

## **10. ФОНДЫ ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕРКИ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ/ПОДКОМПЕТЕНЦИЙ**

ФОС по подкомпетенции ПК-4.ПОСУ. Способен разрабатывать комплексы программных компонентов сетевых устройств.

Фонды оценочных средств представлены отдельными документами и размещены в составе УМК дисциплины электронной информационной образовательной среды ОРИ-ОКС// URL: <http://orioks.miet.ru/>.

## **11. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

### **11.1. Особенности организации процесса обучения**

Данный курс посвящен изучению архитектуры, алгоритмов и протоколов программного обеспечения сетевых устройств. Целью курса является формирование компетенций, необходимых для разработки и отладки компонентов сетевого программного обеспечения встраиваемых устройств на основе операционной системы Linux.

Задача курса «Программное обеспечение сетевых устройств» состоит в следующем:

- познакомиться с принципами построения сетевого программного обеспечения,
- изучить методы взаимодействия компонентов сетевого ПО,
- изучить методы автоматизированного управления сетевыми устройствами,
- изучить и освоить среду разработки ПО для встраиваемых устройств,
- освоить разработку приложений и драйверов сетевой операционной системы,
- освоить средства управления и отладки встраиваемых устройств на основе ОС Linux,
- изучить и освоить программные интерфейсы библиотек и ядра ОС Linux.



Лекции и лабораторные занятия проводятся контактно в соответствии с расписанием. Посещение лекций и лабораторных занятий обязательно. Дополнительной формой контактной работы являются консультации (их посещать необязательно).

Перечень доступных студентам учебно-методических материалов приведен в п. 5, 6, 7.

Подробное описание организации процесса обучения, системы контроля и оценивания изложено в «Методических рекомендациях студентам по изучению дисциплины».

### **11.2. Система контроля и оценивания**

Система контроля включает мероприятия текущего контроля и промежуточную аттестацию. Текущий контроль состоит из сдачи и защиты лабораторных работ. Промежуточная аттестация проходит в форме зачета.

Для оценки успеваемости студентов по дисциплине используется балльная накопительная система. Баллами оцениваются: защита лабораторных работ (в сумме до 80 баллов) и сдача зачета (в сумме до 20 баллов). По сумме баллов выставляется итоговая оценка по предмету. Структура и график контрольных мероприятий приведены ниже в таблице (см. также журнал успеваемости на ОРИОКС, <http://orioks.miet.ru/>).

По сумме баллов выставляется итоговая оценка по предмету. Структура и график контрольных мероприятий доступен в ОРИОКС// URL: <http://orioks.miet.ru/>.

#### **Разработчик:**

Преподаватель кафедры ВМ-1, к.т.н. Трещановский

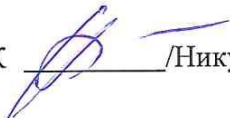
/П.А. Трещановский/

Рабочая программа дисциплины «Программное обеспечение сетевых устройств» по направлению подготовки 02.03.01 «Математика и компьютерные науки», направленность (профиль) «Компьютерная математика и математическое моделирование», разработана на кафедре ВМ-1 и утверждена на заседании кафедры 25.04 2023 года, протокол № 11

Заведующий кафедрой ВМ-1  /А.А. Прокофьев/

### ЛИСТ СОГЛАСОВАНИЯ

Рабочая программа согласована с Центром подготовки к аккредитации и независимой оценки качества

Начальник АНОК  /Никulina И.М./

Рабочая программа согласована с библиотекой МИЭТ

/Директор библиотеки  /Филиппова Т.П./