

Контрольные вопросы

2024-04-19

Разработка данных учебных материалов выполнена авторским коллективом (О. А. Афонина, Д. А. Булах, Я. А. Кириленко) при поддержке [Альянса RISC-V](#) и допускается к использованию под лицензией [CC BY 4.0](#).

Глава 1. Введение в RTOS и встраиваемые системы

1. Что такое встраиваемые системы?
2. Какие аппаратные компоненты обычно входят в состав встраиваемой системы?
3. Какие программные компоненты обычно входят в состав встраиваемой системы?
4. Какую роль печатная плата и программное обеспечение драйвера играют в поддержке аппаратного обеспечения?
5. Чем отличаются системы реального времени от систем общего назначения?
6. Каковы требования к операционным системам реального времени?
7. Что в системах реального времени ценится больше: объём работы, выполняемый в единицу времени, или скорость/предсказуемость отклика?
8. Какие виды алгоритмов планирования задач чаще всего используются в операционных системах реального времени?
9. Какие примеры операционных систем реального времени с открытым и закрытым исходным кодом можно привести?

Глава 2. FreeRTOS

1. Допускает ли лицензия, под которой распространяется FreeRTOS, безвозмездное копирование, изменение и распространение ПО?
2. Как модульность влияет на разработку программного обеспечения для встраиваемых систем?
3. Почему важно использовать абстракцию информации о времени при разработке ПО для встраиваемых систем?
4. Каким образом управление временем простоя системы влияет на эффективность работы встраиваемой системы?

5. Где в дистрибутиве находятся файлы, специфичные для конкретных компиляторов?
6. За что отвечает файл `FreeRTOSConfig.h`?
7. Правда ли, что динамическое распределение предоставляет больший контроль над управлением памятью?
8. Какая из пяти схем управления памятью, определённых FreeRTOS, скорее приведёт к фрагментации?
9. За что отвечает планировщик FreeRTOS?
10. Что такое задача в контексте FreeRTOS?
11. Какие механизмы межзадачного взаимодействия доступны во FreeRTOS?
12. Какие шаги нужно выполнить для создания нового приложения с использованием FreeRTOS?
13. Какие файлы необходимо включить в проект для работы с FreeRTOS?
14. Какие настройки необходимо обновить в файле `FreeRTOSConfig.h` для нового проекта?

Глава 3. Портирование FreeRTOS на другую программную или аппаратную платформу

1. Какие файлы ядра FreeRTOS обязательно должны быть в проекте для нового контроллера?
2. Какие файлы, специфичные для используемого компилятора, необходимо включить в проект?
3. Что обязательно нужно скорректировать в скрипте компоновщика при переносе на новый микроконтроллер?
4. Как настроить прерывание таймера в FreeRTOS?
5. Как управлять использованием ПЗУ и ОЗУ в FreeRTOS?

Глава 4. Процессоры RISC-V

1. Что такое RISC-V?
2. В чем отличия между RISC-V и другими архитектурами?
3. Верно ли утверждение, что в RISC-V у всех инструкций фиксированная длина?
4. За что отвечает интерфейс среды выполнения (EEI) ПО RISC-V?

5. Что такое hart?
6. Какие наборы инструкций RISC-V являются базовыми?
7. Какие регистры определены как необходимые в RISC-V ISA?
8. Есть ли машинные инструкции RISC-V, выполнение которых не требует обращения к памяти?
9. Какими средствами RISC-V обрабатывает исключения и прерывания?
10. Какие инструменты поддерживаются на машинах Linux для разработки программного обеспечения под RISC-V?
11. Какие инструменты используются для компиляции и запуска программных приложений на аппаратных моделях RISC-V?
12. Какие шаги нужно выполнить для установки toolchain на машины Windows и Linux?

Глава 5. Портирование приложений FreeRTOS на процессоры RISC-V

1. Какие шаги нужно выполнить для портирования FreeRTOS на процессоры RISC-V?
2. Какой заголовочный файл необходим для порта RISC-V, требующего расширения архитектуры?
3. В каком случае в конфигурационном файле значения параметров `configMTIME_BASE_ADDRESS` и `configMTIMECMP_BASE_ADDRESS` должны быть установлены в 0?
4. Как указать ядру FreeRTOS, какой обработчик внешних прерываний ему нужно вызвать?

Глава 6. Создание приложений под FreeRTOS и RISC-V

Вопросов нет.