

Контрольные вопросы

2024-04-01

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

Глава 1. Общая информация о языке ассемблера

1. Что такое набор инструкций (instruction set)?
2. Чем отличаются архитектуры CISC и RISC?
3. Почему язык ассемблера может отличаться для различных микропроцессоров?
4. Как называется самая быстрая память внутри микропроцессора?
5. Какие основные этапы можно выделить у конвейера типичного процессора с RISC-архитектурой?
6. Что называется конфликтом конвейера?

Глава 2. Инструменты разработчика

1. Чем объектный файл отличается от исполняемого?
2. В чём отличие ассемблера от языка ассемблера?
3. Как компоновщик получает исполняемый файл из нескольких объектных?
4. Состояние каких модулей процессора позволяет отслеживать Ripes?
5. Чем симулятор отличается от эмулятора?
6. В каких режимах может работать Qemu?
7. Как с помощью objdump посмотреть информацию об исполняемом файле?
8. Как отладчиком подключиться к запущенному процессу в Qemu?
9. Какие команды отладчика необходимо использовать для выполнения программы?
10. Какие команды позволяют просматривать значения переменных/регистров в процессе отладки?

Глава 3. Набор непривилегированных инструкций RISC-V

1. Какие наборы инструкций RISC-V являются базовыми?
2. Какой набор инструкций поддерживает процессор с архитектурой RV128IMFDV?
3. Для чего используются регистры x_{10} - x_{17} ?
4. Чем отличаются различные форматы/типы инструкций?
5. Можно ли выполнить арифметическую операцию над значениями, не загруженными в регистр?
6. Закодируйте инструкцию `andi x1, x0, 7`
7. Среди арифметических инструкций есть инструкции `add`, `sub`, `addi`. Почему нет инструкции `subi`?
8. Инструкцией какого типа является `lb x2, 0(x1)`?
9. Можно ли явно менять значение регистра PC?
10. Какая инструкция используется для совершения системного вызова?
11. Какой из барьеров памяти реализует инструкция `fence`?
12. Можно ли в архитектуре RV32IM с помощью одной инструкции получить результат умножения $46340 * 46341$?

Глава 4. Программирование на языке ассемблера

1. Что такое метки?
2. Что происходит с метками при ассемблировании?
3. Для чего используются функции переразмещения?
4. Для чего предназначена секция `.bss`?
5. Как определить точку входа программы?
6. Что такое псевдо-инструкция?
7. Какие из перечисленных инструкций являются псевдо-инструкциями? `sub`, `por`, `subw`, `ecall`, `mv`, `sw`, `slt`.
8. В чём преимущество позиционно-независимого кода?
9. Что специфицирует соглашение о вызовах?
10. В чём отличие пользовательского уровня, аппаратного и уровня супервизора?

11. Для чего используется стек?
12. Как реализовать `if-then` конструкцию на языке ассемблера?
13. Есть ли специальные инструкции для организации циклов?
14. Можно ли в коде функции менять значение регистра `x1`?
15. Чем принципиально отличается условный и безусловный переход?

Глава 5. Использование системных библиотек

1. Где можно посмотреть, какие номера каким системным вызовам соответствуют?
2. С помощью какой инструкции осуществляется системный вызов?
3. Что регламентирует ABI, кроме соглашения о вызовах?
4. Как должна быть организована программа на языке ассемблера, использующая `glibc`?
5. Как передать Qemu информацию о расположении динамических библиотек?