

Спецификация Binary Digital Small Machine

Оглавление

1. Виртуальная машина.....	3
1.1. Общие сведения.....	3
1.2. Память.....	3
1.3. Процессор	3
1.4. Использование	3
2. Язык программирования	3
2.1. Общие сведения.....	3
2.2. Обработка команд	3
2.3. Перечень инструкций	4
2.3.1. Add	4
2.3.2. Assign.....	4
2.3.3. Div	4
2.3.4. Goto	4
2.3.5. Load.....	4
2.3.6. Mod.....	4
2.3.7. Mul	4
2.3.8. Nop.....	4
2.3.9. Page.....	4
2.3.10. Read	5
2.3.11. Save.....	5
2.3.12. Sub	5
2.3.13. Write	5
2.4. Байт-код	5

1. Виртуальная машина

1.1. Общие сведения

Binary Digital Small Machine (далее BDSM) – регистровая виртуальная машина.

1.2. Память

Команды и данные в BDSM хранятся отдельно. Память команд представляет собой массив, содержащий до 65536 команд, индексация элементов начинается с 0. Память данных – 256 страниц по 256 байт каждая. Единоновременно доступна только одна страница.

1.3. Процессор

Процессор BDSM имеет 3 регистра: 32-разрядный reg, 16-разрядный ip, 8-разрядный dp. Регистр reg – основной регистр BDSM, регистр-аккумулятор; регистр ip содержит индекс текущей команды в массиве команд; регистр dp содержит индекс текущей страницы памяти.

1.4. Использование

Для выполнения программы нужно запустить исполняемый файл виртуальной машины с параметрами командной строки:

bdsvm -p (--plain) <имя файла> – для выполнения из исходного кода

bdsvm -b (--byte) <имя файла> – для выполнения из байт-кода

2. Язык программирования

2.1. Общие сведения

BDSM имеет ассемблероподобный язык программирования, компилируемый в собственный байт-код.

Программа для BDSM представляет собой текст, в котором каждой непустой строке, не являющейся комментарием, соответствует команда. Комментарий – строка, начинающаяся с символа одинарной кавычки (').

Команда := Имя_инструкции [Модификатор_указателя] [Модификатор_типа]
[Аргумент]

Модификатор_указателя := '&'

Модификатор_типа := 'c'|'b'|'w'|'d'

Аргумент := 32-разрядное целое число

Остаток строки после аргумента игнорируется

Значение по умолчанию для модификатора типа – 'b', для аргумента – 0.

Типы:

'c' – символ, 8-бит, используется для ввода/вывода текста

'b','w','d' – байт, слово, двойное слово; 8-, 16-, 32-разрядные числа соответственно

2.2. Обработка команд

Сначала, если присутствует модификатор указателя, то вместо аргумента берется содержимое ячейки с номером <аргумент> с учетом типа. Затем содержимое регистра обрезается в соответствии с модификатором типа, берутся младшие байты регистра. Далее непосредственно выполняется инструкция.

2.3. Перечень инструкций

2.3.1. Add

Допустимые модификаторы: &,b,w,d

Принимает аргумент: да

Прибавляет к регистру reg значение аргумента, результат записывает в reg

2.3.2. Assign

Допустимые модификаторы: &,b,w,d

Принимает аргумент: да

Присваивает регистру reg значение аргумента

2.3.3. Div

Допустимые модификаторы: &,b,w,d

Принимает аргумент: да

Осуществляет целочисленное деление регистра reg на значение аргумента, результат записывает в reg

2.3.4. Goto

Допустимые модификаторы: &,b,w,d

Принимает аргумент: да

Осуществляет переход на команду с индексом <аргумент>, если регистр не пуст

2.3.5. Load

Допустимые модификаторы: &,b,w,d

Принимает аргумент: нет

Заменяет содержимое регистра (адрес) на ячейку памяти с соответствующим адресом.

Адрес обрезается до байта, значение из памяти до модификатора типа

2.3.6. Mod

Допустимые модификаторы: &,b,w,d

Принимает аргумент: да

Записывает в reg остаток от деления reg на аргумент

2.3.7. Mul

Допустимые модификаторы: &,b,w,d

Принимает аргумент: да

Умножает регистр reg на значение аргумента, результат записывает в reg

2.3.8. Nop

Допустимые модификаторы: b

Принимает аргумент: нет

Ничего не делает

2.3.9. Page

Допустимые модификаторы: b

Принимает аргумент: да

Меняет номер текущей страницы памяти на значение аргумента

2.3.10. Read

Допустимые модификаторы: c,b,w,d

Принимает аргумент: нет

Принимает значение указанного типа с клавиатуры ('с' – одиночный символ, в противном случае число)

2.3.11. Save

Допустимые модификаторы: &,b,w,d

Принимает аргумент: да

Записывает содержимое регистра в память по адресу <аргумент>. Если модификатор типа не 'b', то записывает в несколько подряд идущих байт (порядок байт – little-endian)

2.3.12. Sub

Допустимые модификаторы: &,b,w,d

Принимает аргумент: да

Вычитает из регистра reg значение аргумента, результат записывает в reg

2.3.13. Write

Допустимые модификаторы: c,b,w,d

Принимает аргумент: нет

Выводит содержимое регистра на экран ('с' – одиночный символ, в противном случае число)

2.4. Байт-код

Первые два байта байт-кода – беззнаковое 16-битное целое, указывающее количество команд в коде. За ними идут непосредственно команды. Каждая команда представляет собой 5 байт. Первый байт – байт инструкции, следующие 4 – аргумент, 32-битное целое число. Байт инструкции формируется следующим образом:

Instruction_byte := instruction_code | magic | type_code | pointer_code,

где '|' – побитовое «или»

instruction_code – из таблицы:

Instruction_code	Имя инструкции
0x00	Assign
0x10	Save
0x20	Load
0x30	Page
0x40	Read
0x50	Write
0x60	Add
0x70	Sub
0x80	Mul
0x90	Div
0xA0	Mod
0xB0	Goto
0xC0	Nop

magic – 0x08

type_code – 0x00, 0x02, 0x04, 0x06 для 'с', 'b', 'w', 'd' соответственно

pointer_code – 0x01, если используется модификатор указателя; 0x00 в противном случае