

Практикум на ЭВМ. Лабораторная работа № 5: 8 факультет, 1 курс, 1 семестр 2011/12 учебного года
Программирование машин Тьюринга

Когда вице-президента фирмы IBM спросили, почему на персональных компьютерах IBM PS/2 нет кнопки Reset, он ответил: «Наши компьютеры никогда не зависают. Да и шесть долларов, в которые обходится эта кнопка, не липине!»

Составить программу машины Тьюринга в чётвёрках, выполняющую заданное действие над словами, записанными на ленте. Отладку и тестирование проводить в среде интерактивного действующего макета **tu4**. Алфавит МТ определяется заданием. Использование дополнительных (несобственных) букв (кроме λ) нежелательно. Необоснованное использование несобственных букв считается недочётом и является достаточным основанием для снижения оценки.

В начальном состоянии головка МТ находится на пустой ячейке непосредственно справа от записанных на ленте аргументов – слов входного сообщения. В конечном состоянии головка МТ должна находиться на пустой ячейке непосредственно справа от результата (последнего преобразованного или вновь сгенерированного слова результирующего сообщения).

Вычисления в программе, как правило, должны быть нормированными (аргументы после работы программы сохраняются на ленте в неизменном виде и не остаётся промежуточных результатов); ненормированные вычисления, особенно в простых случаях, считаются недочётом и являются достаточным основанием для снижения оценки.

Перед составлением алгоритма для машины Тьюринга необходимо подготовить тесты для него – представительный набор различных входных сообщений, для которых известен правильный ответ, включая значения на границах области определения вычислимой функции и за её пределами.

В отчёте по данной работе необходимо привести диаграмму Тьюринга, эквивалентную реализованной программе.

Примеры программ: ASCII-крутилка и ненормированное сложение двоичных чисел без знака:

00, , - , 00	00, , < , 01	03, 1, < , 03	05, , > , 00	04, , 1, 05	07, , , 07 стоп
00, - , \, 00	01, 0, 1, 02 <i>перенос -1</i>	03, , < , 04	00, 1, > , 00	01, , > , 06	
00, \, , 1, 00	02, 1, < , 01 <i>перенос -1</i>	04, 0, 1, 05 +1	00, 0, > , 00	06, 1, > , 06	
00, , /, 00	01, 1, 0, 03 -1	05, 0, > , 05	04, 1, 0, 02 <i>перенос +1</i>	06, , < , 07	
00, /, , 01	03, 0, < , 03	05, 1, > , 05	02, 0, < , 04 <i>перенос +1</i>	07, 1, , 06 <i>стирание</i>	

Варианты заданий (назначаются преподавателем):

№ Действие (звездочками помечены более трудные задачи)

- 1* Вычисление поразрядной конъюнкции двух двоичных чисел.
- 2* Вычисление поразрядной дизъюнкции двух двоичных чисел.
- 3* Обмен местами двух двоичных чисел.
- 4 Нормированное вычисление суммы двух двоичных чисел без знака.
- 5** Задача 4 с логарифмической сложностью.
- 6 Генерация двух чисел из чётных и нечётных разрядов двоичного числа.
- 7* Генерация двух чисел из разрядов двоичного числа, находящихся на чётных и нечётных позициях.
- 8 Обмен местами разрядов двоичного числа, находящихся на чётных и нечётных позициях.
- 9 Зеркальное отражение цифр двоичного числа относительно его середины
- 10 Вычисление логического произведения (&& в Си) двоичных чисел.
- 11*** Вычисление наибольшего общего делителя двух чисел в натуральной системе счисления.
- 12*** Вычисление наименьшего общего кратного двух чисел в натуральной системе счисления.
- 13*** Проверка делимости на три
- 14 Вычисление двоичного логического сдвига второго числа влево на число разрядов, равное первому.
- 15 Вычисление двоичного логического сдвига первого числа вправо на число разрядов, равное второму.
- 16* Вычисление двоичного арифметического сдвига второго числа влево на число разрядов, равное первому.
- 17* Вычисление двоичного арифметического сдвига первого числа вправо на число разрядов, равное второму.
- 18* Вычисление двоичного циклического сдвига второго числа влево на число разрядов, равное первому.
- 19* Вычисление двоичного циклического сдвига первого числа вправо на число разрядов, равное второму.
- 20* Выделение разрядов первого двоичного числа по маске, заданной вторым числом.
- 21* Выделение разрядов второго двоичного числа по маске, заданной первым числом.
- 22 Закодировать двоичное число азбукой Морзе.
- 23* Умножение двух чисел в кардинальной системе счисления {}.
- 24 Кодирование числа в римской записи по Цезарю (в алфавите {I, V, X, L, C, D, M})
- 25 Умножение однозначных чисел в усечённой римской системе счисления.
- 26* Проверить палиндромию двоичного числа.
- 27* Вычисление двоичного логарифма двоичного числа.
- 28 Натурализация двоичного числа в позиционной записи (перевод в натуральную систему счисления {}).
- 29** Двоичное сложение двоичного и четверичного числа.
- 30 Восстановление целого числа в восьмеричной системе счисления по обратному коду.
- 31 Восстановление целого числа в восьмеричной системе счисления по дополнительному коду.

№	Действие (звездочками помечены более трудные задачи)
32	Уменьшение на единицу целого неотрицательного числа в восьмеричной системе счисления.
33	Увеличение на единицу целого неотрицательного числа в восьмеричной системе счисления.
34	Получение двоичного числа, противоположного данному, в обратной кодировке.
35	Получение двоичного числа, противоположного данному, в дополнительной кодировке.
36*	Вычисление разности двух двоичных чисел без знака, при условии, что первое число больше второго.
37**	Задача 36 с логарифмической сложностью.
38*	Задача 36, ответ — модуль разности.
39**	Задача 37, ответ — модуль разности.
40	Копирование троичного числа со знаком.
41	Реверс троичного числа со знаком (запись цифр в обратном порядке).
42*	Зеркальное отражение двух двоичных слов относительно промежутка между ними.
43	Перевод числа из двоичной системы счисления в восьмеричную.
44	Перевод числа из восьмеричной системы счисления в двоичную.
45	Перевод числа из троичной системы счисления в девятеричную.
46	Перевод числа из девятеричной системы счисления в троичную.
47*	Перевод числа из двоичной системы счисления в восьмеричную с логарифмической сложностью.
48*	Перевод числа из восьмеричной системы счисления в двоичную с логарифмической сложностью.
49*	Перевод числа из троичной системы счисления в девятеричную с логарифмической сложностью.
50*	Перевод числа из девятеричной системы счисления в троичную с логарифмической сложностью.
51	Проверка делимости на 9.
52*	Проверка делимости на 11.
53**	Четверичное сложение двоичного и четверичного числа.

Варианты заданий составлены проф. Зайцевым В.Е., доц. Сошниковым Д.В., ст. преп. Сеницким П.А., Перетягиным И.А. и асп. Макаровым Н.К.

Первый макет МТ разработан в 1987 г. студ. Лукашевичем С. Ю. Описание подготовлено проф. Зайцевым В. Е., доц. Журавлевой Т. Э. и ст. преп. Сеницким П. А. на основе курсовой работы Лукашевича С. Ю. Перенос на DEC Alpha выполнил Сеницкий П. А., модификацию для программ большого размера в окнах X Window System, четырёчную и пакетную версии реализовали Лебедев А. В. и Измайлова А.А. Примеры переработаны Перетягиным И. А., Марухиным А. В., Чечериной С. В. и др. В нынешних версиях макетов ослаблены ограничения на размер программы и длину ленты.

Для *домашних* работ могут использоваться и другие системы Тьюринговских вычислений в среде MS-DOS и MS Windows: tu05m, DDT, мультимодельная система Рыбакова К. А., программы Семёнова А. В. («4», на MS Visual Basic 6.0!), Гуренкова М. Г. В среде UNIX для DEC Alpha есть и пятёрочная версия макета (*tu5* и *tutu*). Есть версия *tu4* для FreeBSD (выполнимый файл, кодировка сообщений КОИ-8), а также для Linux (Intel). Наконец, в рамках проекта GNU существует *gturing*, хороший экранный (в среде GNOME!) интерпретатор МТ с возможностью редактирования текста программы, однако, задаваемой в пятёрках, причём в качестве разделителя полей используется пробел. Существуют самые разнообразные макеты МТ: для MS Excel, Lego-версия <http://legoofdoom.blogspot.com> и iPhone-симулятор <http://mobile.clauss-net.de/Turing>. Видео вполне натуральной машины Тьюринга можно найти здесь <http://aturingmachine.com/>

Вопросы к зачёту по лабораторным работам №№5-7. ВВОД И ПРЕДСТАВЛЕНИЕ ДАННЫХ В ЭВМ

1. Коды ASCII и КОИ-8: характеристики, состав, структура.
2. Альтернативная и основная кодировки (в сравнении с ASCII).
3. Кодировки ISO 8859-5 и CP 1251.
4. Понятие о кодах EBCDIC, ДКОИ.
5. Кодировки Unicode, UTF-8. ISO 8859-5 и 10646.
6. Понятие о клавиатурных раскладках. Основные принципы.
7. Раскладки QWERTY и ЙЦУКЕН и соответствие между знаками кириллицы и латинского алфавита для обычной и фонетической латино-кириллических раскладок.
8. Раскладки Дворака и Diktor.
9. Позиционные системы счисления.
10. Представление целых чисел в ЭВМ.
11. Перевод чисел из одной системы счисления в другую.
12. Особенности целочисленной арифметики в ЭВМ.
13. Научная (экспоненциальная) форма записи числа. Машиное представление с плавающей точкой.
14. Различия представлений числовых и текстовых данных в ЭВМ.
15. Использование калькуляторов ОС UNIX (bc) и MS Windows для операций с числами в различных системах счисления.

Литература

1. Бауэр Ф.Л., Гооз Г. Информатика. –М.: Мир, 1976, 1990. Глава 1 и приложения A; Глава 6 и приложение B (D).
2. Вороцук А.В. Основы ЦВМ и программирование. –М.: Наука, 1978. с. 23-39, 59-80.
3. Форсайт Дж., Мальcolm M., Моулер К. Машинные методы математических вычислений. –М.: Мир, 1980, с. 22-26
4. Карапес С.Б. и др. Машинные алгоритмы обработки информации. –М.: МАИ, 1987. с. 3-16.
5. Танненбаум Э. Многоуровневая организация ЭВМ. –М: Мир, 1979.
6. UNIX on-line manuals.
7. Уэзерелл Ч. Этюды для программистов. –М.: Мир, 1982.

Краткое описание действующего макета машин Тьюринга

Интерактивный полноэкранный макет машин Тьюринга *tu4* реализован на ЭВМ DEC (Compaq, HP) LSI-11, PDP-11, VAX-11, Alpha в среде HP Tru64 UNIX или FreeBSD с терминалами VT-1xx/2xx/3xx/4xx/5xx/XTerm. Запуск макета осуществляется из интерпретатора команд UNIX:

tu4 имя_файла_с_программой.tu

Макет реализует следующий вариант определения МТ:

- лента МТ «полубесконечна» вправо;
- программа МТ состоит из четвёрок $\langle q, a, v, q' \rangle$ где q – символ старого состояния, a – обозреваемая буква, v – символ действия: записываемая буква или команда перемещения, q' – символ нового состояния;
- поддерживаются следующие команды перемещения:
 - $>$ – сдвиг на одну ячейку вправо (знак $>$ реализует символ движения вправо на одну ячейку r);
 - $<$ – сдвиг на одну ячейку влево (знак $<$ реализует символ движения влево на одну ячейку l);
 - $=$ – сохранение текущего положения;
- # – останов макетной МТ (классическая команда *останова* (q, a, a, q) также реализована).

Ограничения реализации для текстовых терминалов 80x24: объём программы – до 168 команд, длина активного участка ленты – 80 ячеек. С программами из более чем 168 команд необходимо работать в окне эмулятора текстового терминала на графическом мониторе в среде X Window System. При этом *перед* вызовом ***tu4*** необходимо соответствующим образом увеличить размеры окна текстового терминала и уменьшить (в особо тяжёлых случаях) размер шрифта в этом окне. Для корректного отображения русских букв на терминале может потребоваться перекодировка программы.

Программы МТ хранятся в текстовых файлах с расширением **.tu** и кодируются по следующим правилам:

- состояния МТ обозначаются двузначными неотрицательными десятичными (а в некоторых версиях – шестнадцатеричными) числами от 00 до 99 (FF соответственно). МТ всегда начинает работать с нулевого состояния;
- буквы рабочего алфавита выбираются из знаков, имеющихся на клавиатуре и отображаемых на терминале, за исключением знаков $<$ $>$ $=$ $#$, изображающих символы действия МТ. Несобственная буква λ кодируется пробелом;
- каждая команда располагается в отдельной строке, начиная с 1-й позиции. Компоненты команды разделяются запятыми. В команде не должно быть пробелов, кроме тех, которые имеют смысл пустых ячеек ленты МТ;
- в тексте программы не должно быть пустых строк.

Ввод и редактирование текста программы МТ можно осуществлять с помощью самого интерпретатора МТ, в котором для этого имеются все необходимые средства; переход в режим редактирования: автоматически при ошибке в программе, по **Tab** в процессе выполнения программы и по **Return/Enter** в других случаях. Поскольку программа МТ является текстовым файлом, она также может быть набрана в любом редакторе текстов *со строгим соблюдением формата*.

Во время работы макета МТ экран терминала имеет следующую структуру:

- в верхней строке отображается текущее состояние ленты МТ;
- во второй строке содержится подсказка по горячим клавишам;
- в остальных строках в несколько столбцов отображается текст программы МТ.

Макет МТ предоставляет пользователю следующие возможности:

- пошаговая интерпретация команд программы;
- быстрая интерпретация программы без остановок;
- редактирование программы МТ, в том числе, интерактивное создание новой программы.

Макет МТ поддерживает следующие горячие клавиши:

- | | |
|---|-----------------------|
| • переход к пошаговой интерпретации программы | <пробел> |
| • переход к быстрой интерпретации программы | Return/Enter |
| • прекращение интерпретации и переход к редактированию, | |
| а также запись редактируемой программы на диск | Tab |
| • прекращение работы макета МТ | Ctrl / C |

Для протоколирования работы МТ необходимо использовать пакетную версию интерпретатора, вывод которой – строго текстовый. Текст отлаженной программы поместить в файл типа **tu**, например с именем **15.tu**, а входные данные к ней – в файл **15.tst**, и в режиме протоколирования вызвать пакетную версию интерпретатора МТ:

turun имя-файла.tu имя-файла-с-тестами.

Каждый тест занимает две строки. Первая строка должна содержать исходное состояние ленты МТ(слова, обрамлённые пробелами), вторая указывает начальную рабочую ячейку знаком ^, предварённым необходимым числом пробелов.