

Описание автокода ЭЦВМ М-20 (первое поколение советских ЭЦВМ)

Автор:

Стефанков Дмитрий Викторович

Версия 0: 27 февраля 2015 года

Версия 2: 10 мая 2015 года

1. Общее назначение

Автокод М-20 предназначен для трансляция программ на символическом языке (языке ассемблера) в машинный код целевой системы.

Данная реализация автокода М-20 не имеет промышленного качества (т.е. не реализует большинство возможностей и не сообщает о многих ошибках) и больше предназначена для учебных и демонстрационных целей системы символического кодирования. Тем не менее при помощи автокода можно создавать рабочие программы, которые проще сопровождать, чем программы в машинном коде.

Данное описание является кратким руководством по автокоду М-20.

В случае возникающих сомнений используйте примеры или посмотрите текст автокода. В автокоде не используется реализация разбора соответствующей грамматики. Возможно в будущем, так и будет сделано. Сейчас используется позиционный разбор полей одной строки с определенными ограничениями. В любом случае смотрите исходные тексты автокода, если есть какие-то сомнения.

Примечание.

Нужно было реализовать автокод крайне быстро. Поэтому был выбран позиционный разбор и автокод был создан в два дня. И он даже вполне прилично работает.

2. Возможности автокода

Автокод М-20 предоставляет следующие возможности:

- символическое представление машинных операций;
- символическое представление адресов (меток);
- восьмеричную форму записи числа;
- десятичную форму записи числа;
- псевдооперации (директивы);
- выходной файл в текстовом формате эмулятора М-20;
- выходной файл объектного листинга программы М-20;
- входной файл символического кодирования для М-20;
- загружаемые таблицы сообщений, директив, кодов операции;
- русские кодировки (cp866, cp1251, koi8-r, utf-8);
- поддерживает Unix/Linux и Microsoft Windows платформы;

3. Порядок работы с автокодом

Нужно создать файл на языке автокод М-20. Обыкновенно он имеет расширение «**.a20**».

После этого выполнить трансляцию файла.

Пусть файл автокода имеет следующее имя «**hello.a20**».

Тогда строка трансляции будет примерно такой:

```
> autocode_m20.exe -vp -e 1 -i hello.a20 -o hello.m20 -l hello.l20 >hello.p20
```

autocode_m20.exe - программа-транслятор автокода

hello.a20 - входной файл на языке автокода

hello.m20 - выходной файл в формате эмулятора М-20

hello.l20 - выходной файл объектного листинга

hello.p20 - выходной файл фаз разбора программы автокода

-v - опция увеличения глубины сообщений

-p - опция выдачи фаз разбора

Все опции командной строки автокода можно посмотреть так:

```
> autocode_m20.exe -h
```

После трансляции нужно посмотреть объектный листинг. Убедитесь, что все переменные разрешены верны, введенные числа верны и т.д. Если необходимо, то внесите нужные изменения и вновь транслируйте файл.

Если трансляция выполнена верна, то можно передать выходной файл формата эмулятора М-20 для исполнения эмулятором М-20.

Еще раз повторим: автокод М-20 не обнаруживает ваших ошибок, он их просто пропускает и обыкновенно записывает в эту ячейку (или часть ячейку) нуль.
Поэтому проверяйте трансляцию имен и меток на нуль, когда вам нужен точно не нуль.

Строка автокода ориентирована на формат машинной инструкции М-20.

4. Формат строки автокода

Текстовые строки автокода имеет следующие форматы:

- (1) **[метка] директива [комментарий]**
- (2) **[метка] операция [операнд1] [операнд2] [операнд3] [комментарий]**
- (3) **[метка] число [комментарий]**
- (4) **комментарий**

Символы комментария: “*” и “;” .

“*” можно использовать только в первой позиции.

“;” можно использовать только в любой позиции.

Все, что начинается с символа комментария далее игнорируется.

Числа могут быть в восьмеричном или десятичном формате.

Пример восьмеричного формата (аналогично эмулятору М-20):

1 02 4000 0000 0000

2 50 0013 7000 7600

1

2 22

3 33 3333

4 44 4444 4444

5 55 5555 5555 5555

Пример десятичного формата (аналогично эмулятору М-20):

=10

=1E-1

Числа могут быть и командами (приказами).

Текст автокода может использовать русские или английские названия для директив и кодов операций. Поскольку работать можно только с одной кодировкой, то директивы и коды операций должны быть записаны в единой кодировке.

На имена меток и адресов данное ограничение не распространяется.

Операция может быть числом или символическим именем.

Операнды могут числами или символическими именами.

Максимальная длина имени метки – 32 символа.

Максимальная длина символического имени адреса – 64 символа.

Максимальная длина имени абсолютной величины – 64 символа.

Максимальное число меток в программе и символических имен – 512.

Максимальное число абсолютных величин в программе – 512.

Максимальное число строк в файле – 10000.

Допускается смешивать имена машинных инструкций (кодов операций) в короткой и длинной формах представления.

Символ “*” может использовать в операндах как текущий счетчик.

Рекомендуется использовать следующий стиль:

(1) метки начинаются с @;

(2) директивы и абсолютные величины используют заглавные буквы;

(3) символические имена адресов используют прописные буквы;

(4) используется только одна кодировка для представления;

Имена меток, символических имен, абсолютных величин могут состоять из букв, цифр и неслужебных символов как @, _ и некоторые другие.

Допускаются операции сложения «+» и вычитания «-» для символических адресов и абсолютных величин. Пробелов между операндами и знаками операций не должно быть.

Примеры:

num1+num2

***+5**

число1-АБС_ВЕЛ_1

число2-10

Для задания признака адреса нужно адресное выражение заключать в круглые скобки.

Примеры.

пер_если_ра_мнш_a1 СЧЕТЧИК_ЗОН-1, @РАЗМЕТКА, (1)

слож_команд ЧТЕНИЕ_КМД, (МЛ_РАЗМЕТКА_ДАнные), ЧТЕНИЕ_КМД

слож_он (0020),(0021),(число3) ; 2+2=4?

5. Директивы автокода

Директивы (псевдо-операции) автокода позволяют программисту упрощать написание программ.

Ниже приводится список директив автокода М-20.

Английское название	Русское название	Синтаксис	Назначение
.NAME	.ИМЯ	имя-дир. строка	.имя программы
.TITLE	.НАЗВАНИЕ	имя-дир. строка	Назначение программы
.START	.СТАРТ	имя-дир. число	начальный адрес программы
.FINISH	.ФИНИШ	имя-дир. [число]	конец программы [начальный адрес старта программы]
.ADDRESS	.АДРЕС	имя-дир. число	замена текущего счетчика
.ABS	.АБС	имя-дир. число	абсолютная величина
.MEM	.ПАМ	имя-дир. число	резервировать ячейки памяти
.AUTHOR	.АВТОР	имя-дир. строка	имя автора
.LIST	.ПЕЧАТЬ	имя-дир.	вывод строк в объектный листинг
.NOLIST	.НЕПЕЧАТЬ	имя-дир.	нет вывода строк в объектный листинг

Все числа должны быть восьмеричными и в диапазоне (0-7777).

Только директива «.ABS» допускает восьмеричный и десятичный формат представления и число будет храниться как 45-разрядное машинное слово.

6. Загружаемые таблицы автокода

Имена табличных файлов фиксированы и явно «зашиты» в программу.

Файлы содержат таблицы сообщений, директив и кодов операций.

autocode_m20_eng.tab	- таблицы на английском языке
autocode_m20_dos_cp866.tab	- таблицы на русском языке (кодировка cp-866)
autocode_m20_win_cp1251.tab	- таблицы на русском языке (кодировка cp-1251)
autocode_m20_unix_koi8r.tab	- таблицы на русском языке (кодировка koi8-r)
autocode_m20_rus_utf8.tab	- таблицы на русском языке (кодировка utf-8)

Максимальная длина сообщения - 128 символов.

Максимальная число сообщений - 1024.

Максимальная длина имени директива (псевдооперации) – 32 символа.

Максимальная число директив - 128.

Максимальная длина имени короткой формы машинной инструкции - 32 символа.

Максимальная длина имени длинной формы машинной инструкции - 128 символов.

Число инструкций должно быть 64.

7. Пример программы на автокоде

Следующая программа выполняет простейшее сложение.

Файл hello.a20.

```
*  
* Sample file for M-20 autocode  
* encoding: english,ASCII-7  
*  
* Copyright (c) 2015 Stefankov  
*  
    .NAME    hello.a20
```

```

        .TITLE      Hello Program for M-20
        .AUTHOR     Stefankov Dmitry

*
*
*
TWO1      .ABS      =2
TWO2      .ABS      1 02 4000 0000 0000
TWO3      .ABS      1024000000000000

        .START 20

*
num1: TWO1
num2: TWO2
num3: TWO3
*

        .START 40
sum1: .MEM 1
sum2: .MEM 1
sum3: .MEM 1
*
*

        .ADDRESS 100
START:
        add_rn num1, num2, sum1      ; 2+2=4?
        add_rn num1, num3, sum2      ; 2+2=4?
        add_rn num2, num3, sum3      ; 2+2=4?
        stop_77
        stop_17

        .FINISH 100

```