**Программный эмулятор ЭЦВМ М-20  
(первое поколение советских ЭЦВМ).  
Описание форматов файлов для устройств ввода-вывода.**

**Автор:   
Дмитрий Викторович Стефанков**

**Версия 0: 29 ноября 2014 года.  
Версия 5: 10 мая 2015 года.**

**1. Введение**  
Данное руководство содержит описание форматов файлов для устройств ввода-вывода эмулятора ЭЦВМ М-20.  
Общие сведения по М-20 могут найдены в литературе, список которой приведен в общем описании проекта по М-20.

**2. Соглашения об именах файлов**  
Для удобства и единообразия рекомендуется использовать следующие расширения файлов:  
  
**.simh** или **.ini** - конфигурационный файл эмулятора SIMH (текстовой формат)  
**.m20** - команды и данные М-20 для загрузки в МОЗУ с дальнейшим исполнением  
(в некотором смысле аналог инженерного пульта)  
**.cdr** - файл ввода с перфокарт (текстовой формат)  
**.cdp** - файл вывода на перфокарты (текстовой формат)  
**.drum** или **.drm** – образ МБ (бинарный формат)  
**.mt** - образ МЛ (бинарный формат)  
**.lst** - файл печати (текстовой формат)  
**.a20** – файл автокода М-20 (текстовой формат)

**3. Формат файла эмулятора SIMH (.simh или .ini)**  
Файл имеет текстовой формат.  
Полное описание данного формата можно найти в соответствующем руководстве SIMH.  
  
Ниже приводятся примеры наиболее часто используемых директив.  
Все числа даются в восьмеричном формате.  
  
; Загрузить код и данные из файла  
**load test\_0017.m20**  
  
; Дамп памяти в диапазоне указанных адресов  
**ex 100-105**

; Дизассемблировать в диапазоне указанных адресов  
**ex -m 100-105**

; Запуск загруженной программы   
; (адрес запуска должен быть указан в файле или задается вручную)  
**run**; Начальная загрузка с ЧУ (чтение перфокарт)  
**boot cdr**; Вывести состояние триггера w **ex tsw**; Изменить состояние переменной  
**de BCDPRINT 1**

; Завершить работу с эмулятором  
**quit**

; Установить значение параметра  
; (здесь включить отладку для ЦПУ и МБ и вывести отладочные сообщения в файл)  
**set console debug=test\_0101\_debug.txt  
set cpu debug  
set drum debug**; Вызвать внешнюю команду  
**! del test\_0102\_debug.txt**; Соединить (назначить или ассоциировать ) файл с устройством **att drum0 test\_0102.drum0**; Вывести сообщение на консоль **echo Start**; Вывести текущее состояние очередей и затраченных тактов **show queue  
show time**

; Строка комментария  
**; This is comment**

**4. Формат файла загрузки и исполнения для М-20 (.m20)**  
Файл имеет текстовой формат.  
Все числа даются в восьмеричном формате (если не оговорено иное).  
  
Строки комментариев начинаются с символа “;”.  
Пустые строки игнорируются.  
  
Формат изменения текущего адреса: **“:AAAA**”  
Пример.  
**:0521**  
**:1030  
:0001**

Формат установки начального адреса исполнения: **“@AAAA**”  
Пример.  
**@0100 ; Старт**

Формат ввода 45-разрядного кода или числа: **“O OO OOO OOO OOO**”  
Пример.  
**0 50 4040 0005 0177**

Формат ввода десятичного числа: “**=decimal-number**”  
Примеры.  
**=1  
=-1  
=10  
=1958  
=1.5E+30**

Формат ввода двоично-десятичного числа: “**# binary-coded-decimal-number**”.  
Каждая цифра – это цифра двоично-десятичного числа от 0 до 9 (группа в 4 бита)  
Первое число содержит в разряде 0 знак порядка (экспоненты), в разряде 1 знак числа (мантиссы), в разряде 2 признак числа.  
Следующее число – это экспонента без знака (безнаковый порядок числа),   
величина в пределах от 0 до 19.  
Следующие числа – это группа мантиссы (9 цифр).  
Примеры.  
**# 0 02 256 000 000 ; 25.6  
#1 03 300 000 000 ; 0.0003  
# 2 01 590 000 000 ; -5.9  
# 3 01 500 000 000 ; -0.05  
# 2 00 500 000 000 ; -0.5**  
Формат ввода двоично-десятичного числа: “[**+|-][+|-] [+|-]exp mantissa**”.  
Первый символ – признак числа («+» - нет признака, «-» - есть признак)  
Второй символ – знак числа («+» - положительное число, «-» - отрицательное число)  
Третий символ с последующим числом – это экспонента со знаком (порядок числа), величина в пределах от 0 до 19.  
Следующие числа – это группа мантиссы (9 цифр).  
Каждая цифра – это цифра двоично-десятичного числа от 0 до 9 (группа в 4 бита)  
Примеры.  
**-+ +00 500 000 000  
-- -03 100 000 000  
++ +00 314 159 265  
++ +00 100 000 000  
++ +00 200 000 000  
++ +00 300 000 000  
++ +01 100 000 000  
+- +01 100 000 000  
+- +11 124 381 413 ; -12438141361,326  
++ -03 462 318 497 ; 0.00046231849671  
+- +02 147 121 899 ; -14,712189915**

**5. Формат файла ввода с перфокарт (.cdr)**

Устройство присоединяется к файлу командой **attach**.  
Пример.  
**att -r cdr test\_0060.cdr**  
  
Файл определен как устройство с последовательным доступом.Файл имеет текстовой формат.  
Все числа даются в восьмеричном формате (если не оговорено иное).  
  
Строки комментариев начинаются с символа «;».  
Пустые строки игнорируются.

Остальные строки должны иметь следующий формат:  
<основной маркер><пробел><число><пробел><вспом.маркер>  
  
Расшифровка значений маркеров:  
О=0, В=0 (неверная комбинация)  
О=1, В=0 (числовой код)  
О=0, В=1 (адресный код)  
О=1, В=1 (завершение ввода с контрольной суммой)

Числовой код вводится в восьмеричном виде или в десятичном виде   
(см. описание формата загрузки для М-20).  
Адресный код вводится в восьмеричном виде (на месте адреса 1).  
Контрольная сумма вводится в восьмеричном виде.

**Пример 1.  
; Sample data for M-20 card reader  
; Format of data  
; M addr VM M=0,VM=1, addr=45-bit word, but only address 1 is using  
; M code VM M=1,VM=0, code=45-bit word  
; M sum VM M=1,VM=1, sum=45-bit word  
; data at 0020  
0 0 00 0020 0000 0000 1  
1 1 02 4000 0000 0000 0  
1 123456707654321 0  
1 1 14 6000 0000 0000 0  
1 0 77 0000 0000 0000 0  
; chksum  
1 0 00 0000 0000 0000 1  
; end of data  
  
  
Пример 2.  
; data at 0020  
0 0 00 0020 0000 0000 1  
1 1 02 4000 0000 0000 0  
1 1 02 4000 0000 0000 0  
1 1 03 4000 0000 0000 0  
1 0 00 0000 0000 0000 0  
  
; real checksum at 0040  
;  
  
; data at 0041  
0 0 00 0041 0000 0000 1  
1 1 00 0000 0000 0000 0  
1 1 00 0000 0000 0000 0  
1 1 00 0000 0000 0000 0  
1 1 00 0000 0000 0000 0  
  
; code at 0050  
0 0 00 0050 0000 0000 1  
1 0 01 0020 0021 0030 0  
1 0 21 0020 0021 0031 0  
1 0 41 0020 0021 0032 0  
1 0 61 0020 0021 0033 0  
1 0 77 0000 0000 0000 0  
  
; code at 0060  
0 0 00 0060 0000 0000 1  
1 =-1 0  
1 =1 0  
1 =2 0  
1 =3 0  
  
; code at 0070  
0 0 00 0070 0000 0000 1  
1 ++ +01 500 000 000 0  
1 +- -02 100 000 000 0  
1 -+ +01 314 159 265 0  
1 -+ +01 100 000 000 0  
1 -+ +01 200 000 000 0  
1 -+ +01 300 000 000 0  
  
; chksum  
1 5 51 4231 0104 0147 1  
;1 0 00 0000 0000 0000 1  
;1 1 53 4231 0104 0147 1  
  
; end of data**

**6. Формат файла вывода на перфокарты (.cdp)**

Устройство присоединяется к файлу командой **attach**.  
Пример.  
**att cdp test\_0111.cdp**  
Файл определен как устройство с последовательным доступом.Файл имеет текстовой формат.  
Все числа даются в восьмеричном формате (если не оговорено иное).  
По умолчанию перфорация в восьмеричном виде.  
По команде “**de BCDPRINT 1**” осуществляется перфорация в десятичном виде   
(двочно-десятичное представление).  
  
**Пример 1 (двоично-десятичная перфорация из СП 013).  
; common codes section  
1 11111 111111111 0  
1 00022 222222222 0  
1 11133 333333333 0  
1 00004 444444444 0  
1 11115 555555555 0  
1 00026 666666666 0  
1 11137 777777777 0  
1 00008 888888888 0  
1 11119 999999999 0  
1 00000 000000000 0  
1 1113 0  
1 1113 0  
1 01010 123456789 0  
1 10101 23456789 0  
1 01012 3456789 0 0  
1 10103 456789 01 0  
1 01014 56789 012 0  
1 10105 6789 0123 0  
1 01016 789 01234 0  
1 10107 89 012345 0  
1 01018 9 0123456 0  
1 10109 01234567 0  
  
; end-of-input marker and checksum  
1 10003 987 1**

**Пример 2 (восьмеричная перфорация из СПП 0013).  
; common codes section  
1 7 21 0421 0421 0421 0  
1 0 42 1042 1042 1042 0  
1 7 63 1463 1463 1463 0  
1 0 04 2104 2104 2104 0  
1 7 25 2525 2525 2525 0  
1 0 46 3146 3146 3146 0  
1 7 67 3567 3567 3567 0  
1 0 10 4210 4210 4210 0  
1 7 31 4631 4631 4631 0  
1 0 00 0000 0000 0000 0  
1 7 77 7777 7777 7777 0  
1 7 77 7777 7777 7777 0  
1 2 20 0443 2126 3611 0  
1 5 01 1064 2547 4237 0  
1 2 22 1505 3170 4760 0  
1 5 03 2126 3611 7401 0  
1 2 24 2547 4237 0022 0  
1 5 05 3170 4760 0443 0  
1 2 26 3611 7401 1064 0  
1 5 07 4237 0022 1505 0  
1 2 30 4760 0443 2126 0  
1 5 11 7401 1064 2547 0  
  
; end-of-input marker and checksum  
1 4 03 7755 6272 4607 1**

**7. Формат файла печати (.lst)**

Устройство присоединяется к файлу командой **attach**.  
Пример.  
**att lpt sample\_0004.lst**  
Файл определен как устройство с последовательным доступом.Файл имеет текстовой формат.  
Все числа даются в восьмеричном формате (если не оговорено иное).  
По умолчанию печать в десятичном виде (двоично-десятичном).  
По команде “**set lpt OCTHELPFMT**” печать в восьмеричном виде.  
По команде “**de LPTWIDTH 1**” меняется количество чисел при выводе одной строки.  
По команде “**ex DPTYPE 1**” выбирается один из вариантов для десятичной печати.  
  
**Пример 1 (десятичная печать без + и -).   
00001187500000  
00001180000000  
01000100000000  
01002205000000  
00002924000000  
00000000000000**  
**Пример 2 (десятичная печать).   
---11 111111111   
+++22 222222222   
---33 333333333   
+++04 444444444   
  
Пример 3 (восьмеричная печать).   
101400000000000  
101400000000000  
102400000000000**

**8. Формат файла образа МБ (.drum или .drm)**Устройство присоединяется к файлу командой **attach**.  
Пример.  
**att drum0 test\_0102.drum0  
att drum1 test\_0102.drum1**Файл имеет двоичный формат (бинарный формат) и состоит из 45-ти разрядных кодов.  
Для упрощения работы каждый код хранится в виде 64-разрядного слова (8 байт).  
Файл определен как устройство с произвольным доступом.  
МБ можно трактовать как внешнюю память из 4096 кодов.   
При записи 4096 кодов (нужно отключать контрольное суммирование) или писать только 4095 кодов с включенным контрольным суммированием (ячейка 0 имеет всегда значение 0, поэтому ее содержимое можно не писать и не считывать).   
Эмулятор устройства автоматически пропускает чтение/запись по адресу 0, т.е. эффективно реализуя чтение/запись группы длиной 1-4095 кодов (эту опцию **AUTO\_SKIP\_ZERO\_ADDRESS** можно отключить).

**9. Формат файла образа МЛ (.mt)**

Устройство присоединяется к файлу командой **attach**.  
Пример.  
**att mt0 test\_0091.mt0  
att mt1 test\_0091.mt1**Файл имеет двоичный формат (бинарный формат) и состоит из 45-ти разрядных кодов.  
Для упрощения работы каждый код хранится в виде 64-разрядного слова (8 байт).  
Файл определен как устройство с произвольным доступом.При форматировании МЛ используется следующий формат для разметки зон:  
8-байт – номер и размер зоны  
(нижние 4-ти байта – это номер, старшие 4-ти байта – это размер зоны в кодах)  
Далее следуют число кодов в зоне (т.к. N\*8 байт)  
Затем резервный код для контрольной суммы всей группы (1\*8 байт).  
Итого размер зоны в кодах: 1+N+1.  
Итого размер зоны в байтах: (1+N+1)\*8.  
Максимальный размер зоны равен 4095 кодам, а контрольная сумма для зоны будет записана в 4096-м коде (резервном коде).  
Просмотр ленты всегда начинается с начала ленты независимо от вида операции (чтение или запись). Разряд реверса ленты значения не имеет.  
Разметка ленты начинается с текущей позиции. Если файла нет, то начальная позиция равна 0 и далее можно размечать с разными номерами и размерами для зон МЛ.Эмулятор устройства автоматически пропускает чтение/запись по адресу 0, т.е. эффективно реализуя чтение/запись группы длиной 1-4095 кодов (эту опцию **AUTO\_SKIP\_ZERO\_ADDRESS** можно отключить).

**10. Формат файла автокода (.a20)**

Формат файла описан в руководстве по автокоду М-20.