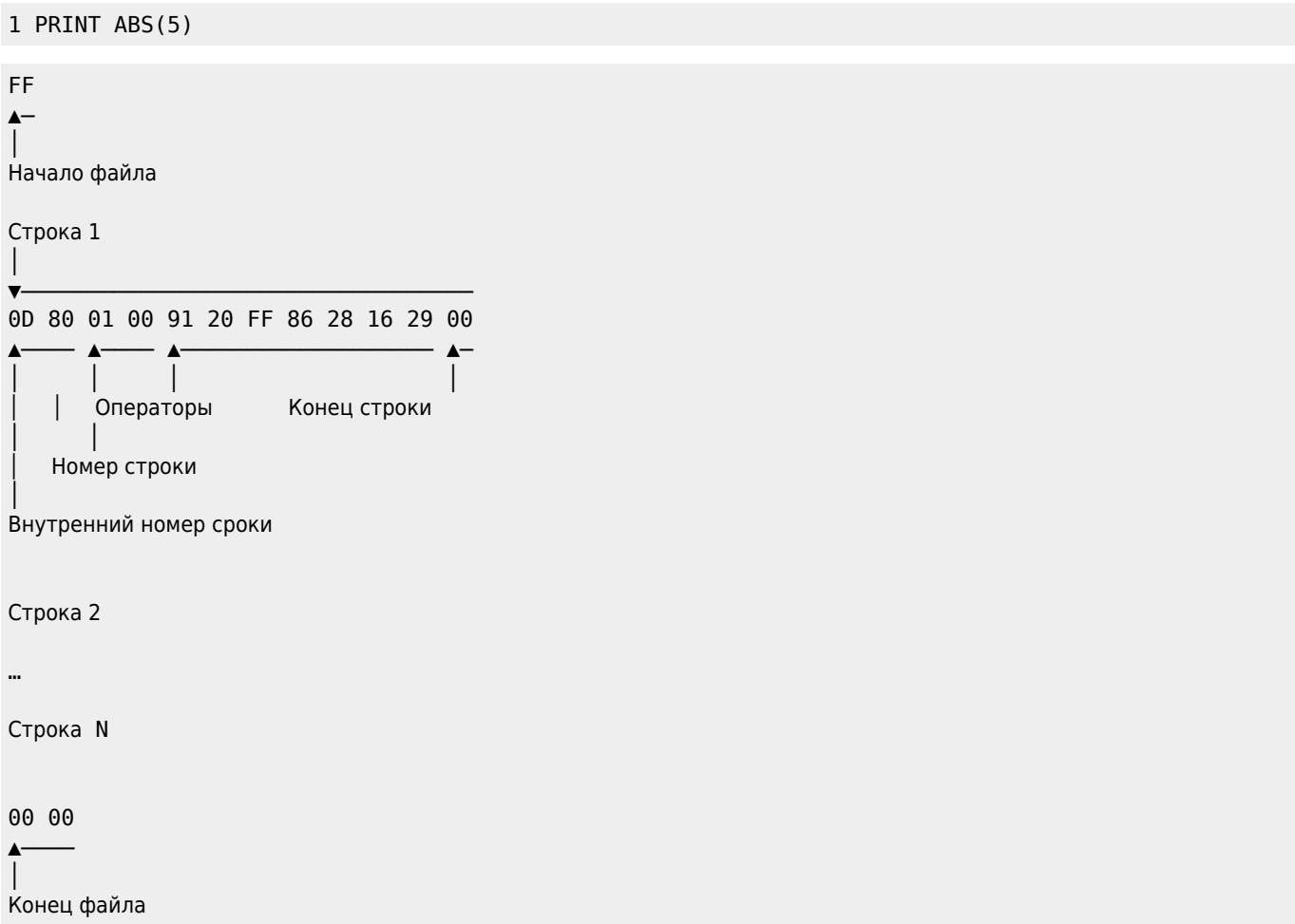


MSX BASIC — Промежуточный код

Для хранения программ MSX BASIC в ОЗУ и компактной записи программ используется перевод лексем языка в внутренние коды. В этом виде программа хранится в таблице PIT (Program Instruction Table).

Формат упакованного файла

Структура файла на примере:



Внутренний номер строки, и номер строки это 2 байта в обратном порядке: младший, старший.

Числа (переменные)

Название, тип	Префикс	Длина (байт)	Пример		Примечание
			значение	байты	
Номер строки в операторах перехода	0x0E	2	520	0E 08 02	2 байта в обратном порядке: младший, старший
Восьмеричное число	0x0B	2	&O10	0B 08 00	
Шестнадцатеричное число	0x0C	2	&H2A0	0C A0 02	
Целое число 256-32767 (%)	0x1C	2	1000	1C E8 03	
Целое число 10-255 (%)	0x0F	1	11	0F 0B	

Десятичная число одинарной точности

Десятичная число одинарной точности (6 цифр) (!)

- Префикс: 0x1D
- Длина (байт): 4
- В первом байте значение второго полубайта — положение десятичной точки

Примеры:

Значение	Байты
.123456	1D 40 12 34 56
1.23456	1D 41 12 34 56
12.3456	1D 42 12 34 56
123.456	1D 43 12 34 56
1234.56	1D 44 12 34 56
12345.6	1D 45 12 34 56
123456!	1D 46 12 34 56

Десятичная число двойной точности

Десятичная число двойной точности (14 цифр) (#)

- Префикс: 0x1F
- Длина (байт): 8
- Если значение второго байта меньше 0x4F, то значение его второго полубайта — положение десятичной точки, иначе это порядок числа для [экспоненциальной записи](#).

Примеры:

Значение	Байты
.12345678901234#	1F 40 12 34 56 78 90 12 34
1.2345678901234#	1F 41 12 34 56 78 90 12 34
12.345678901234#	1F 42 12 34 56 78 90 12 34
123.45678901234#	1F 43 12 34 56 78 90 12 34
1234.5678901234#	1F 44 12 34 56 78 90 12 34
12345.678901234#	1F 45 12 34 56 78 90 12 34
123456.78901234#	1F 46 12 34 56 78 90 12 34
1234567.8901234#	1F 47 12 34 56 78 90 12 34
12345678.901234#	1F 48 12 34 56 78 90 12 34
123456789.01234#	1F 49 12 34 56 78 90 12 34
1234567890.1234#	1F 4A 12 34 56 78 90 12 34
12345678901.234#	1F 4B 12 34 56 78 90 12 34
123456789012.34#	1F 4C 12 34 56 78 90 12 34
1234567890123.4#	1F 4D 12 34 56 78 90 12 34
12345678901234#	1F 4E 12 34 56 78 90 12 34

Экспоненциальная запись

Порядок числа = Значение второго байта - 65

Примеры:

Значение	Байты
1.2345678901235E+14	1F 4F 12 34 56 78 90 12 35
1.2345678901235E+15	1F 50 12 34 56 78 90 12 35
1.2345678901235E+16	1F 51 12 34 56 78 90 12 35
1.2345678901235E+17	1F 52 12 34 56 78 90 12 35
1.2345678901235E+18	1F 53 12 34 56 78 90 12 35

Обычный текст

Название	Префикс
Кавычки (")	0x22
Комментарий (REM)	0x8F
Комментарий (')	0x3A8FE6

Внутренние коды служебных слов

Варианты:

- [Без префикса](#)
- С префиксом
 - [0xFF](#)
 - [0x3A](#)

Без префикса

Слово / символ	Код(ы)
AND	0xF6
ATTR\$	0xE9
AUTO	0xA9
BASE	0xC9
BEEP	0xC0
BLOAD	0xCF
BSAVE	0xD0
CALL	0xCA
CIRCLE	0xBC
CLEAR	0x92
CLOAD	0x9B
CLOSE	0xB4
CLS	0x9F
CMD	0xD7
COLOR	0xBD
CONT	0x99
COPY	0xD6
CRSLIN	0xE8
CSAVE	0x9A
DATA	0x84

Слово / символ	Код(ы)
DEF	0x97
DEFDBL	0xAE
DEFINT	0xAC
DEFSNG	0xAD
DEFSTR	0xAB
DELETE	0xA8
DIM	0x86
DRAW	0xBE
DSKI\$	0xEA
DSKO\$	0xD1
END	0x81
ERASE	0xA5
ERL	0xE1
Слово / символ	Код(ы)
ERR	0xE2
ERROR	0xA6
EQU	0xF9
FIELD	0xB1
FILES	0xB7
FN	0xDE
FOR	0x82
GET	0xB2
GOSUB	0x8D
GOTO	0x89
INKEY\$	0xEC
INPUT	0x85
INSTR	0xE5
IMP	0xFA
IPL	0xD5
IF	0x8B
KEY	0xCC
KILL	0xD4
LET	0x88
LINE	0xAF
LIST	0x93
LFILES	0xBB
LLIST	0x9E
LOAD	0xB5
LOCATE	0xD8
LPRINT	0x9D
LSET	0xB8
MAX	0xCD
MERGE	0xB6

Слово / символ	Код(ы)
MOD	0xFB
MOTOR	0xCE
NAME	0xD3
NEW	0x94
Слово / символ	Код(ы)
NEXT	0x83
NOT	0xE0
OFF	0xEB
ON	0x95
OPEN	0xB0
OR	0xF7
OUT	0x9C
PAINT	0xBF
PLAY	0xC1
POINT	0xED
POKE	0x98
PRESET	0xC3
PRINT	0x91
PSET	0xC2
PUT	0xB3
READ	0x87
RENUM	0xAA
REM	0x8F
RESTORE	0x8C
RESUME	0xA7
RETURN	0x8E
RSET	0xB9
RUN	0x8A
SCREEN	0xC5
SET	0xD2
SAVE	0xBA
SPC	0xDF
SPRITE	0xC7
SOUND	0xC4
STEP	0xDC
STOP	0x90
STRING\$	0xE3
SWAP	0xA4
Слово / символ	Код(ы)
TAB	0xDB
THEN	0xDA
TIME	0xCB
TO	0xD9

Слово / символ	Код(ы)
TRON	0xA2
TROFF	0xA3
USING	0xE4
USR	0xDD
VARPTR	0xE7
VDP	0xC8
VPOKE	0xC6
WAIT	0x96
WIDTH	0xA0
XOR	0xF8
0	0x11
1	0x12
2	0x13
3	0x14
4	0x15
5	0x16
6	0x17
7	0x18
8	0x19
9	0x1A
>	0xEE
=	0xEF
<	0xF0
+	0xF1
-	0xF2
*	0xF3
/	0xF4
^	0xF5
\	0xFC

0xFF

Слово / символ	Код(ы)
ABS	0xFF86
ASC	0xFF95
ATN	0xFF8E
BIN\$	0xFF9D
CDBL	0xFFA0
CINT	0xFF9E
CHR\$	0xFF96
COS	0xFF8C
CSNG	0xFF9F
CVD	0xFFAA

Слово / символ	Код(ы)
CVI	0xFFA8
CVS	0xFFA9
Слово / символ	Код(ы)
DSKF	0xFFA6
EOF	0xFFAB
EXP	0xFF8B
FIX	0xFFA1
FPOS	0xFFA7
FRE	0xFF8F
HEX\$	0xFF9B
INP	0xFF90
INT	0xFF85
LEN	0xFF92
LEFT\$	0xFF81
LOC	0xFFAC
Слово / символ	Код(ы)
LOG	0xFF8A
LOF	0xFFAD
LPOS	0xFF9C
MID\$	0xFF83
MKD\$	0xFFB0
MKI\$	0xFFAE
MKS\$	0xFFAF
OCT\$	0xFF9A
PAD	0xFFA5
PDL	0xFFA4
PEEK	0xFF97
POS	0xFF91
Слово / символ	Код(ы)
RIGHT\$	0xFF82
RND	0xFF88
SGN	0xFF84
SIN	0xFF89
SPACE\$	0xFF99
SQR	0xFF87
STICK	0xFFA2
STRIG	0xFFA3
STR\$	0xFF93
TAN	0xFF8D
VAL	0xFF94
VPEEK	0xFF98

0x3A

Слово / символ	Код(ы)
ELSE	0x3AA1
'	0x3A8FE6

Скрипты

code-to-text.py

Скрипт работает в [Python](#) 2 и 3

[code-to-text.py](#)

Читает файл в бинарном формате и сохраняет его содержимое в текстовом формате

Параметры:

1. Исходный файл в бинарном формате
2. Выходной файл в текстовом формате

Пример запуска:

```
./code-to-text.py file.bas file_a.bas
```

Тесты

Запуск скрипта с тестовым файлом:

```
./code-to-text.py code.bas code.txt
```

Сравнение файлов:

```
diff code.txt code_a.bas
```

code.bas	Тестовый файл в бинарном формате, создан командой: save "code.bas"
code_a.bas	Тестовый файл в текстовом формате, создан командой: save "code_a.bas",a

Ссылки

[MSX Datapack](#)

[MSX2 Technical Handbook](#)

[Konamiman/MSX2-Technical-Handbook Chapter2 3.3 Storage format of BASIC programs](#)

[Konamiman/MSX2-Technical-Handbook Appendix2 Math-Pack](#)

 [Число одинарной точности](#)

 [Число двойной точности](#)

 [Проект на GitHub](#)

http://sysadminmosaic.ru/msx/basic_intermediate_code/basic_intermediate_code

2023-06-12 00:27

