MSX X-BASIC

X-BASIC , известный как MSX-BASIC KUN , KUN BASIC или TURBO BASIC (Turbo-BASIC) это run-time компилятор для MSX BASIC, который обычно используется для ускорения существующих программ на MSX BASIC и создания демо и игр.

x-basic-2.1.rom, оригинал

Его развитием является NestorBASIC.

KUN BASIC

MSX-BASIC-KUN (Компилятор BASIC)

автор J.Suzuki 1989
этот документ и примеры составлены Sho Endo
перевод на английский выполнен LASP[1)]
текст перепечатан из LASP без изменений Нестором Сориано
корректура и перевод на русский — ATroubleshooter

MSX-BASIC-KUN это потрясающий компилятор BASIC. Он компилирует программу на MSX BASIC в ОЗУ за считанные секунды и она работает в 15-100 раз быстрее!! X-BASIC может компилировать почти все операторы и функции MSX BASIC и поддерживает строки и числа с плавающей запятой. Попробовав его, вы поймете, что вам теперь ни к чему учить ассемблер Z-80. Игры в реальном времени, компьютерная графика, демонстрационные программы могут быть написаны, сочетая простоту MSX BASIC и скорость машинного кода.

Применение

1. Установка и Общие сведения

Данный компилятор продается в Японии в виде картриджа за 4500 йен. Для использования, вам нужно просто вставить его в слот. Также, вы можете найти этот компилятор в MSX2+ компьютерах Sanyo, в ПЗУ. А ещё, он может быть загружен в ОЗУ (с 4000h по 7FFFh) из MSX-DOS, что мы сейчас и проделаем. Просто запустите `X.BAT` из MSX-DOS.

```
A>x
set40 xbasic.bin
basic
.
.
Ok
```

Скачать X-BASIC 1.0 в формате .bin и утилиту SET40.COM

Теперь вы в режиме Бейсика, всё как обычно — за исключением того, что вам доступны две новые команды:

```
CALL RUN
CALL TURBO ON/OFF
```

`CALL` может быть записано как «_» (подчеркивание). Будем использовать такую запись далее.

`_RUN` это команда для компиляции и запуска всей программы в памяти. Если она находит ошибку она останавливается и выдает сообщение.

`_TURBO ON` это оператор, определяющий начало турбо-блока.
`_TURBO OFF` обозначает конец турбо-блока.

Турбо–блок это часть программы, которую вы хотите исполнять быстро. Если, программа в целом, содержит какие-то некомпилируемые операторы, вы можете определить блок для компиляции, используя эту конструкцию.

Пример:

```
100 SCREEN 8:DEFINT A-Z
110 BLOAD"PICTURE",S
120 _TURBO ON
130 FOR X=0 TO 255
140 LINE(X,0)-(X,211),0
150 NEXT X
160 _TURBO OFF
170 GOTO 170
```

Эта программа не может быть запущена с помощью _RUN, потому что BLOAD это одна из команд, которые не могут компилироваться. Если запустить это с помощью RUN, участок программы с 130–й по 150–ю строки будет исполнен быстро.

Поскольку _RUN«FILE» не поддерживается, вам придется добавить _TURBO ON и _TURBO OFF в начале и в конце программы, если вы хотите использовать RUN«FILE» и получить эффект.

```
100 FOR I=0 TO 999
110 ...
 .
 .
 .
890 'END OF THE PROGRAM
```

Итак, это может быть запущено через _RUN, или, после добавления 10 _TURBO ON и 900 _TURBO OFF, через RUN«FILE».

Если попробовать запустить посредством _RUN программу, содержащую _TURBO ON/OFF, произойдет ошибка.

_TURBO ON/OFF нельзя писать в одну строку с другими операторами.

Конструкции _TURBO ON/OFF не могут быть вложенными. Но можно иметь много турбо–блоков в программе.

Переменные и массивы существуют отдельно для турбо–блоков и остальной программы. После завершения турбо–блока, переменные и массивы, используемые в турбо–блоке, будут потеряны. Только переменные целочисленного типа могут быть определены как общие.

```
100 DEFINT A-Z:DIM C(2),D(2)
110 A=1:B=2:C(0)=3:D(0)=4
120 _TURBO ON(A,C())
130 DIM D(2)
140 PRINT A,B,C(0),D(0)
150 A=5:B=6:C(0)=7:D(0)=8
160 _TURBO OFF
170 PRINT A,B,C(0),D(0)

RUN
 1 0 3 0
 5 2 7 4
Ok
```

Числа с плавающей запятой, используемые в компиляторе, имеют особый трехбайтовый формат. Его точность примерно 4.5 знака. Двойная точность не поддерживается.

Массивы должны статически объявляться в начале.

Этот компилятор работает с программой на BASIC в ОЗУ, и помещает сгенерированный объектный код и переменные в незанятую часть ОЗУ. Поэтому есть ограничение на размер исходной программы примерно в 10Кб. Большие массивы, строковые переменные (каждая занимает 256 байт), CLEAR ???,&H???? сделает ситуацию ещё напряженней, как вы понимаете. Скомпилированные объектные файлы не могут быть сохранены как самостоятельные программы.

Поддерживаются прерывания, такие как KEY(1) ON, OFF и т.п. Но их использование понизит качество исполняемого объектного кода в плане размера и скорости.

Некоторые операторы могут работать некорректно.

```
100 GOTO 130
110 A=3/2
120 END
130 DEFINT A-Z
140 GOTO 110
```

Если запустить это через RUN, то A будет 1. Если запустить через _RUN, A будет 1.5. В случае интерпретатора, DEF??? сработает в порядке его очереди исполнения, тогда как в противном случае это зависит от номера строки.

Мало-мальски сложная строковая операция легко может вызвать ошибку «String formula too complex». Потому что данный компилятор имеет только один уровень стэка для этого. Разбейте длинную строковую формулу на несколько небольших, в таком случае.

Если запустить посредством _RUN бесконечную программу, вы не сможете её прервать. Сделайте выход по нажатой клавише.

```
100 GOTO 100 'Нажмите ресет или выключите питание чтобы остановить
```

```
100 IF INKEY$="" THEN 100
110 END
```

будет лучше.

2. Отличия от интерпретатора MSX-BASIC

Список операторов, команд и функций, которые не могут быть откомпилированы:

```
AUTO
BASE
BLOAD
BSAVE
CALL
CDBL
CINT
CLEAR
CLOAD
CLOAD?
CLOSE
CONT
CSAVE
CSNG
CVD
CVI
CVS
```

```
DEFFN
DELETE
DRAW
DSKF
EOF
ERASE
ERL
ERR
ERROR
EQV
FIELD
FILES
FPOS
FRE
```

```
GET
IMP
INPUT#
```

```
KEY LIST
LFILES
LINEINPUT#
LIST
LLIST
LOAD
LOC
LOF
LPRINT USING
LSET
MAXFILES
MERGE
MOTOR
MKD$
MKI$
MKS$
NAME
```

```
NEW
ON ERROR GOTO
ON INTERVAL GOSUB (из-за ошибки)
OPEN
PLAY
PRINT#
PRINT USING
PUT KANJI
RENUM
RESUME
RSET
SAVE
SPC
TAB
TRON
TROFF
WIDTH
```

Список тех, которые имеют ограничения:

`CIRCLE`	Только окружности, никаких дуг или эллипсов.
`COPY`	Только пересылки в видеопамяти.
`DEFDBL`	То же что и `DEFSNG`.
`DIM`	Должен определяться только вначале программы либо внутри блока компиляции.
`INPUT`	Может принимать только одну переменную.
`KEY`	Только для `ON KEY GOSUB`, `KEY(n) ON/OFF`.
`LOCATE`	Обе координаты должны быть определены. Параметр переключения курсора недоступен.
`NEXT`	Имя переменной цикла после `NEXT` не может быть опущено.
`ON`	`ON STOP GOSUB`, `ON INTERVAL GOSUB` не поддерживаются.
`PRINT`	Разделители обрабатываются иначе. Нет выравнивания для цифр.
PUT	Только `PUT SPRITE`
RUN	Переменные не инициализируются.
`SCREEN`	Только номер режима и размер спрайтов.
SET	Только `SET PAGE`

STOP	Такое же поведение как у END
USR	Параметром может быть только Intrger
VARPTR	Дескриптор файла не может быть передан в качестве параметра.

В остальном, нет существенной разницы.

Мы видим, что, как правило, не могут быть скомпилированы команды и функции ввода–вывода, а также команды редактирования. Конечно же они доступны в интерактивном режиме, и вне турбо–блоков. Вы можете редактировать, отлаживать и сохранять программы в MSX BASIC и запускать их, используя _RUN.

Если вы хотите использовать PRINT# для вывода символов на GRP:, используйте его вне турбо-блока. Иначе, изучите пример PRINT.TRB.

Если нужно использовать PLAY, используйте расширение CompileBGM, и выводите звук с помощью USR(n).

3. Новые возможности, которые были добавлены

Появились 3 специальные команды, которые вводятся как строка комментария, начинающаяся с определенного символа:

- #I — Означает INLINE. Позволяет написать короткую программу в машинных кодах.

```
100 DEFINT A-Z
110 K=1
120 '#I &H2A,K
130 '#I &HF3,&HCD,@150,&HFB
140 END
150 'SUB
160 RETURN
```

```
120 означает LD HL,(K) ; K должна быть простой переменной целого типа.
130 означает DI
             CALL @150 ; Будьте осторожны, эта строка не будет переименована с помощью RENUM.
             EI
```

- #C — Означает CLIP (отсечение). В экранных режимах с 5 по 8 (за исключением PAINT и CIRCLE), это будет включать и отключать отсечение.

```
10 SCREEN 8
20 '#C-
30 LINE(0,0)-(255,255) 'Y CLIPPED
40 IF INKEY$="" THEN 40
50 '#C+
60 LINE(0,0)-(255,255) 'NOT CLIPPED
70 IF INKEY$="" THEN 70
```

- #N — Проверка на переполнение в ходе NEXT.

```
10 FOR I%=0 TO &H7FFF
20 NEXT I%
```

Эта программа завершится ошибкой «Overflow error» в MSX BASIC. А будучи запущена путем _RUN, она войдет в бесконечный цикл. Если указано #N+, то она завершится нормально. Этот прием тоже снизит эффективность объектного кода. Лучше не использовать без реальной нужды. Чтобы сбросить, нужно указать #N-.

К сведению: В Sanyo MSX-2+ вы найдете новую команду: CALL BC Эта команда задействует BASIC COMPILER.

Данный текст представляет собой синтез переводов двух источников, датированных 1986 и 1989 годами.

Примечание 1
Довольно сомнительно, что текст был переведен LASP (бумажный каталог-прейскурант, выходивший в Испании). Этот текст в английском варианте существовал с 1986 года, и был, с минимальными изменениями, напечатан в LASP в 1989 году.

Ссылки

MSX-BASIC-KUN (BASIC COMPILER) | MSX INFORMATIE & LINSTINGS COLLECTIE

MSX-BASIC-KUN (BASIC COMPILER) | MSX2-Technical-Handbook

X-BASIC

Трактат о X-Basic'e

1)
Смотри Примечание 1

https://sysadminmosaic.ru/msx/x-basic/x-basic

2023-07-07 21:18