BCBASIC Documentation

William R Cooke

August 2, 2013

Contents

1 Introduction **5**
1.1 Why are We Here? 5

2 What is a Compiler **7**

3 The BCBASIC Language **9**
3.1 Blast from the Past 9
3.2 Example Program 9
3.3 Data Types 9
3.4 Expressions and Operators 10
3.4.1 Operator Precedence 10
3.5 Statements 10

4 Overview of the Compiler **11**

5 The Lexical Analyzer **13**

6 Parsing: Your English Teachers' Revenge **15**

7 Semantics and Symbols **17**

8 The Intermediate Language: Quadruples **19**

9 Appendix: Language Grammar **21**

10 Appendix: Language Reference **23**

Chapter 1

Introduction

1.1 Why are We Here?

A compiler is often like an oracle. You prepare your source code, feed it to the compiler with some magic incantations, and (hopefully) get a working machine language program out the other side. A lot of programmers don't really know much about how a compiler works and are often amazed when leaving out a single character causes the compiler to spit out a long string of strange error messages. "It should have KNOWN I meant to put a semicolon there!"

Chapter 2

What is a Compiler

Chapter 3

The BCBASIC Language

3.1 Blast from the Past

BCBASIC is intended to have the flavor of "classic" BASIC but modernized to make it more useful and productive. To my mind, "classic" BASIC means Microsoft BASIC, as was used on almost all early personal computers. Much of the style and syntax are borrowed from those early BASICs, with a lot of structure added. I borrowed heavily from C when adding to the language, but I tried to be careful not to turn it into a veiled form of C. This chapter is a reference manual for the language itself. A lot of people, including Microsoft, have extended BASIC to make it more modern, more structured, and even object oriented. To me, these aren't BASIC; they look like some horrid cross of Java structure with Pascal syntax and a few BASIC keywords thrown in for effect. We will try to avoid that.

I have made a few departures from BASIC. Two big ones are: there will not be a GOTO, and there are subroutines with parameters and local variables.

3.2 Example Program

What does a BCBASIC program look like? If you have programmed in older versions of BASIC (NOT Visual Basic) it should look mostly familiar.

3.3 Data Types

There are six basic data types in BCBASIC. They are shown in the chart below. They type of a variable is indicated by a special character attached to the end of the name.

Type Character Size Example BYTE @ 8 byt@ INT % 16 i% WORD (none) 16 w LONG INT & 32 l& SINGLE ! 32 s! STRING $ (varies) a$

In addition to simple variable, there are single dimensioned arrays of each type and functions that return each type.

Ordinary variables do not need to be declared. They are implicitly declared by their first use. Arrays and certain special types must be declared prior to use. The DIM statement is the general purpose declaration statment.

DIM a%(25) DIM b@ AS EEPROM DIM c AS STATIC

ROM EEPROM STATIC VOLATILE BYTE WORD INT LONG STRING SINGLE

3.4 Expressions and Operators

3.4.1 Operator Precedence

Unary: -, , !, NOT

3.5 Statements

Chapter 4

Overview of the Compiler

Chapter 5

The Lexical Analyzer

Chapter 6

Parsing: Your English Teachers' Revenge

Remember diagramming sentences in 8th grade English? You've sworn your whole life it would never be useful. Guess who is going to have the last laugh. Your English teacher.

A programming language isn't a real language like English, German, or Latin, but it has a lot of similarities, especially in how the parts are put together. Consider this description of one simple type of sentence:

Sentence is : subject predicate .

It says that a sentence is a subject, followed by a predicate, followed by a period. Now let's define what a subject and a predicate might be:

Subject is : noun or : article noun or : article adjective noun Predicate is : verb or : verb adverb

Now we need to define some more things:

Noun is : box or : boy or : car or : ship or : planet Article is : A or : An or : The Adjective is : brown or : big or : heavy or : fancy or : mysterious

Chapter 7

Semantics and Symbols

Chapter 8

The Intermediate Language: Quadruples

Chapter 9

Appendix: Language Grammar

program stlist END sublist EOF stlist statement: statement assignst — dimst —forst — ifst — whilest — repeatst — subst — breakst — continuest — returnst

assignst var = exp dimst DIM vardec specifier forst FOR ident = exp TO exp [STEP exp] stlist NEXT ifst IF exp THEN stlist [ELSE stlist] END whilest WHILE exp stlist (END—WEND) repeatst REPEAT stlist UNTIL exp subst ident(paramlist) breakst continuest returnst

exp

term factor mulop factor factor := numconst — charconst — constant — var — funcall — (exp)

sublist := subprog subprog := procedure — function — isr procedure := SUB [importlist] stlist END

ABS INTERRUPT SUB FUNC AT AS VAL LEFT$ MID$ RIGHT$ REPEAT RETURN IF THEN ELSE WHILE WEND UNTIL FOR NEXT DIM STR$ STATIC BYTE WORD INTEGER LONG SINGLE ROM EEPROM LEN REM BREAK CONTINUE IMPORT EXPORT

Chapter 10

Appendix: Language Reference