GAS Tutorial - 3. Syntax

Young W. Lim

2016-03-01 Tue

Outline

1 Syntax

Based on

"Using as", Dean Elsner, Jay Fenlason & friends

Symbols

- 1+ characters chosen from the set of
  - all letters (both upper and lower case)
  - digits
- ‘.$′ On most machines, you can also use $

in symbol names; exceptions are noted in Chapter 9 [Machine Dependencies], page 79.

- may not begin with a digit
- case is significant
- no length limit
- delimited
  - by characters not in that set
  - by the beginning of a file

Statements

- A statement ends at a newline character (‘’) or line separator character(;).
- A statement begins with zero or more labels
- A statement optionally followed by a key symbol which determines what kind of statement it is.
    - the symbol begins with a dot ‘.’ : an assembler directive
    - the symbol begins with a letter : an assembly language instruction:
- A label is a symbol immediately followed by a colon (:)

Number Constants

- Integers : int in the C language
  - binary : ‘0b’ or ‘0B’
  - octal : ‘0’
  - decimal : starts with a non-zero digit
  - hexdecimal : ‘0x’ or ‘0X’
- Bignums : int more than 32 bits
  - the same syntax and semantics as int
- Flonums : floating point numbers
  - The digit ‘0’
  - A letter e
  - (‘+-’)(int part).(frac part)(exp part)

Configuring listing output : '–listing'

This feature combines the input source file(s) with a hex dump of the corresponding locations in the output object file
The format of this listing can be controlled by directives inside the assembler source

- .list
- .title
- .sbttl
- .psize
- .eject

Join Text and Data Sections : ‘-R’

- treat all data-section data in the text section.
- the data section part of your object file is zero bytes long
- generate shorter address displacements

TTTT