ASM programming language 0.0

Kajetan Rzepecki

2011-10-08

Contents

Part I

Lexical

Chapter 1

Source code

ASM source text can use one of the following formats:

- ASCII
- UTF-8
- UTF-16BE
- UTF-16LE
- UTF-32BE
- UTF-32LE

UTF-8 is a superset of traditional 7-bit ASCII. One of the following UTF BOMs (Byte Order Marks) can be present at the beginning of the source text:

- ASCII - no BOM
- UTF-8 - EF BB BF
- UTF-16BE - FE FF
- UTF-16LE - FF FE
- UTF-32BE - 00 00 FE FF
- UTF-32LE - FF FE 00 00

1.1 Character set

```
Characters = {Character};
Character  = ? any Unicode character ?;
```

Any valid Unicode character is also a valid ASM sourcecode character.

1.2 Whitespace

```
Whitespace = Space, {Space};
Space      = \U0009 | \U000B | \U000D | " " | EndOfLine;
```

Whitespace are a pretty cool guy. Eh separates tokens from eachother and doesn't afraid of anything.

1.3 End of line

```
EndOfLine = \U000A | EndOfFile;
```

1.4 End of file

```
EndOfFile = \U0000 | ? physical end of file ?;
```

ASM source text is terminated by whichever comes first.

Chapter 2

Comments

```
Comment      = LineComment | ExpressionComment;
CommentStart = "#";
```

ASM supports two kinds of comments - line comments and expression comments.

2.1 Line comments

```
LineComment = CommentStart, CommentOpt, Characters, EndOfLine;
CommentOpt  = "!" | "#" | " ";
```

Line comments start with CommentStart coupled with a comment option and span until the EndOfLine:

- Line comment options
  CommentOpt determines the purpose of a line comment:
  - " " - starts a regular comment.
  - "!" - starts a shebang comment.
  - "#" - starts a documentation comment.
- Shebang
  ASM suports Unix shebang notation. First line starting with "#!" is simply discarded.

  Example:

  ```
  #!/bin/asm

  (var foo 'bar)
  ```
- Documentation Comments
  Line comments with a hash option are considered documentation comments and can be used to generate code documentation automatically.

  Examples:

[]javascript This is a documentation comment. (var of-this-variable)

This too is a documentation comment.

(var of-this-other-variable) So is this.

See more here.

2.2 Expression comments

```
ExpressionComment = CommentStart, ? any syntactically valid ASM
expression ?;
```

Expression comments in ASM are useful to comment out entire S-expressions and are an equivalent of block comments in imperative languages. There are no different kinds of expression comments and their sole purpose is aiding the programmer in debugging and writting selfexplanatory code.

Examples:

[]javascript (if (some-condition) (some buggy-piece-of-code) (1) (a-quick-fix) else (2) (else-clause))

In (1) the programmer commented out some buggy piece of code and supplied a quick fix. (2) demonstrates the use of expression comments as means of clarifying the code.