

Assignment 2

Michael Hunsinger

September 13, 2014

1 Change log

- Changed the Reader from the bufio package to the bytes package. The bufio Reader is made to read streaming data and cannot perform seeks.
- Changed `compiler.go` to a `scanner.go`, so it is more appropriately named for future implementations.
- Added a `parser.go` file to implement a compiler's parser functionality.
- Added the ability to pass in the file on the command line.

2 Go Documentation

This implementation of the scanner uses Google's new language Go. There are instructions on how to setup Go, a description of the file structure, and how to compile and run the program.

2.1 Install Go

Download the appropriate installation from Google's Go website, <http://golang.org/doc/install>, there is additional documentation located on website as well.

2.2 Go's Workspace

Extract files from the tarball into the desired location. Inside the root folder you will find four directories

- `bin` compiled executables, along with sample micro program files
- `doc` documentation
- `pkg` package objects (the compiler package is located in here)
- `src` source files
 - `compiler` source files pertaining to the compiler package
 - `main` source files pertaining to the main package (the driver file)

We must also setup the `GOPATH` to ensure proper compilation of the files. Follow the steps below to set `GOPATH` in a *unix environment.

```
$ cd ../01
$ export GOPATH=$HOME/your/path/here/01
```

2.3 Compiling Source Files

There are two steps to compile and the executable; building the compiler package and then build the executable.

```
$ cd ../01
$ go build compiler
$ go install main
```

Now there is an executable in the `bin` folder.

2.4 Running the Program

You can run the executable that was compiled. Ensure you are in the directory where the `sample.micro` file is located.

```
$ cd ../02/bin
$ ./main sample.micro
```

This will run scan the `sample.micro` file. There is also a `sample2.micro` file in the `bin` folder that uses some of the tokens found in the extended Micro language. If you wish to scan this file, you will need to change file name in `../src/main/main.go` on line 18.

3 Sample Input and Output

There are still some existing issues when outputting larger files (`sample2.micro`) for instance doesn't print quite right.

`sample.micro` input

```
BEGIN A := B + (72 - C); END
```

`sample.micro` output

```
<system goal> --> BeginSym <program> EndSym EofSym
<system goal> --> BeginSym <statement list> EndSym EofSym
<system goal> --> BeginSym <statement> EndSym EofSym ahead AssignOp
<system goal> --> BeginSym <ident> := <expression> ; EndSym EofSym
<system goal> --> BeginSym Id := <expression> ; EndSym EofSym
<system goal> --> BeginSym Id := <primary> <add op> <expression> ; EndSym EofSym
<system goal> --> BeginSym Id := <ident> <add op> <expression> ; EndSym EofSym
<system goal> --> BeginSym Id := Id <add op> <expression> ; EndSym EofSym
<system goal> --> BeginSym Id := Id PlusOp <expression> ; EndSym EofSym
<system goal> --> BeginSym Id := Id PlusOp <primary> ; EndSym EofSym
<system goal> --> BeginSym Id := Id PlusOp ( <expression> ) ; EndSym EofSym
<system goal> --> BeginSym Id := Id PlusOp ( <primary> <add op> <expression> ) ; EndSym EofSym
<system goal> --> BeginSym Id := Id PlusOp ( IntLiteral <add op> <expression> ) ; EndSym EofSym
<system goal> --> BeginSym Id := Id PlusOp ( IntLiteral MinusOp <expression> ) ; EndSym EofSym
<system goal> --> BeginSym Id := Id PlusOp ( IntLiteral MinusOp <primary> ) ; EndSym EofSym
<system goal> --> BeginSym Id := Id PlusOp ( IntLiteral MinusOp <ident> ) ; EndSym EofSym
<system goal> --> BeginSym Id := Id PlusOp ( IntLiteral MinusOp Id ) ; EndSym EofSym
```