

Solution Review: Analyzing Input from Keyboard

This lesson discusses the solution to the challenge given in the previous lesson.

Environment Variables



Key:	Value:
GOROOT	/usr/local/go
GOPATH	//root/usr/local/go/src
PATH	//root/usr/local/go/src/bin:/usr/local/go...

```
package main
import (
    "fmt"
    "bufio"
    "os"
    "strings"
)
var nrchars, nrwords, nrlines int

func main() {
    nrchars, nrwords, nrlines = 0, 0, 0
    inputReader := bufio.NewReader(os.Stdin)
    fmt.Println("Please enter some input, type S in the new line to stop: ")
    for {
        input, err := inputReader.ReadString('\n')
        if err != nil {
            fmt.Printf("An error occurred: %s\n", err)
            return
        }
        if input == "S\n" { // Windows it is "S\r\n", on Linux it is "S\n"
            fmt.Println("Here are the counts:")
            fmt.Printf("Number of characters: %d\n", nrchars)
            fmt.Printf("Number of words: %d\n", nrwords)
            fmt.Printf("Number of lines: %d\n", nrlines)
            os.Exit(0)
        }
        Counters(input)
    }
}

func Counters(input string) {
    nrchars += len(input) - 2 // -2 for \r\n
    // count number of spaces, nr of words is +1
    nrwords += strings.Count(input, " ") + 1
    nrlines++
}
```

Click the **RUN** button, and wait for the terminal to start. Type `go run main.go` and press ENTER.

The variables `nrchars`, `nrwords` and `nrlines` contain the counts we must make: they are declared at **line 8**, and initialized to **0** at **line 11**. Then, we construct at **line 12** a new buffered reader to read from the keyboard. **Line 14** starts an infinite loop:

- The keyboard input is read in a string input until a new line is found at **line 15**. At **line 16**, we check the error variable `err`: if it contains an error, we jump out of the loop.
- Then, we check if the input contains the letters `S` at **line 20**. If so, we print out the *counts* found (from **line 21** to **line 24**) and exit the program at **line 25**.
- At **line 27**, the function `Counters` is called with input as a parameter.

The function of `Counters` is defined as follows:

- `nrchars` is incremented with the length of the input string, `-2` for `\r\n` on Windows, or `-1` for `\n` on Linux (see **line 32**).
- `nrwords` is incremented with the number of words; this is the number of spaces in input + 1 (see **line 34**).
- `ReadString` just reads one line, so `nrlines` is simply incremented with **1** (see **line 35**).

That's it for the solution. In the next lesson, you'll see how Go provides support for reading from a file.