Lab 2: Bash Scripting

Due September 27th, 2016

In this lab, you will learn about the basics of Bash scripting by writing a configuration file parser in Bash.

Bash is the name of a scripting language that is commonly used within the Linux operating system. It is primarily used to automate tasks on the terminal. Many programs in Linux also use **configuration files** as a powerful way for users to set options and specifications for the program. Linux config files commonly end with a *.conf* extension and have the following syntactical properties:

1. Variables can be declared using the syntax ⟨name⟩=⟨value⟩
   Note that there are no spaces on either side of the equal sign. This is important, as many configuration files are whitespace-sensitive.

2. Comments are supported in configuration files and begin with a # symbol, identical to how comments work within a Bash script.

You will write a Bash script called **configParser.sh** that will accept the name of a configuration file as an argument, read through it, and create a series of environment variables that match the variables declared in the configuration file. In Linux, environment variables are used by the operating system to declare useful information for use at any time. For example, the PATH environment variable declares a series of directory paths for the operating system to check when you run a terminal command(this is why you can run *ls* on the terminal without declaring its full path every time, which is most likely */usr/bin/ls*). To create or change an environment variable on your system, you can use the *export* command in the terminal like so:

export TESTING=hello

To check the value of an environment variable, you can use the *echo* command followed by the name of the environment variable with a dollar sign

prepended to it. So to check the value of the TESTING variable above, you would type:

echo $TESTING

To list all environment variables on your system, type ***env*** at the terminal and press enter.

Notice that the standard format for environment variables in Linux is all-caps. **This is a requirement for your script!** Even if a variable is declared in the config file in lowercase, your script should convert it to all-caps before declaring it as an environment variable.

Your parsing script should be able to handle an arbitrary number of variable declarations. For this lab, you can assume that any configuration files given to your script will be properly formed(that is, you do not need to worry about error handling in this lab). Your script must also be able to handle full-line comments within the given configuration file. You do not need to worry about partial-line comments(comments that appear on the same line as a valid variable declaration). For each line in the configuration file, your script should create an environment variable of the same name with the same value.

Here is an example of a configuration file that your script would have to parse:

#Example Configuration File
#Advanced CS
hello=goodbye
TESTING=foo
bEsTeVeR=AdvancedCS

After running your script on this file, you should be able to run the following commands and see the following output:

```
$ echo $HELLO
   goodbye
$ echo $TESTING
   foo
$ echo $BESTEVER
   AdvancedCS
```

Your script will be graded according to the following table:

Successfully read a configuration file given as an argument.	5
For each line of the config file, create an appropriate environment variable.	10
Handle full-line comments within the config file.	10
TOTAL	25 pts

If you have any questions, contact your instructor at `acryker@gmail.com`. All due dates are final unless you choose to use one or more of your late days. Your lab submission **must** consist of a single tarball containing all of the relevant files for your lab. Submission of single code files or anything other than a valid tarball file ending in *.tar* will result in a zero for that lab. Refer to the syllabus for more information.