

Bash

Author: Rachid Ounit, Brian Crites

In this lab you will create your first Shell scripts, and especially some scripts to manage C++ files. Get each section checked off by your TA after you have completed every section. Several sections mention tutorials that you should read. If you find that any tutorial does not adequately explain the concept then I encourage you to look for additional tutorials.

Introduction

In this part, you will create and run your first Shell script, printing in the standard output “Hello world!”.

1. Read the manual for the commands echo and chmod.
2. Create your first Shell script named “hello.sh” (remember to use vim). In the first line of the file, write “#!/bin/sh”, this commented line indicates that this file is a shell script. Then, type ‘echo “Hello world!”’. Save your file.
3. Execute your script: ./hello.sh. What is the result?
4. Make your file executable (i.e., chmod 711 hello.sh). Execute hello.sh.

*If Shell is a new language to you then feel free to consult online tutorials (see next page) **before** moving to the next parts.*

Parameters

You can pass parameters to your Shell scripts. For example, \$1 is the first parameter passed (similarly for \$2, etc.) and \$# is the number of parameters passed.

1. Write a Shell script, “a.sh”, that outputs only the first parameter. Run it with several test cases.
2. Write a Shell script, “b.sh”, that outputs your first and last name concatenated.

Empty C++ main file

1. Read the manual for the command cat.
2. Create a header text file entitled “<Firstname>_<SID>.txt”, where <Firstname> is your first name and <SID> your Student ID. Its content is your first name, last name, email address, and SID. Use rigorously the format provided in the example file.
3. Write a script named “createEmptyMainFile.sh” that automatically create a source file (“main.cc”), in the working directory, that contains the content of “<Firstname>_<SID>.txt” at the top, and its last two lines are:

```
int main(int argc, const char** argv)
{ }
```

Empty C++ class files

Write a script “createEmptyClassFiles.sh” that takes in parameter the class name (e.g., “MyClass”) and produces in the working directory two files “MyClass.cc” and “MyClass.hh”, the source and header files respectively. In addition, the last lines of “MyClass.hh”/“MyClass.cc” are:

MyClass.h

```
#ifndef MyClass_hh
#define MyClass_hh

class MyClass
{
    public:
        MyClass();
        ~MyClass();

    private:
};
#endif
```

MyClass.cc

```
#include "../MyClass.hh"

MyClass::MyClass()
{}

MyClass::~~MyClass()
{}

```

If and Else

You can use if and else logic in your Shell scripts as well, however the syntax is slightly different from the if/else syntax in C++.

1. [Read this tutorial on using if/else in bash](#)

2. [Read this tutorial on using comparison operators in bash](#)
3. Write a program called largest.sh that takes three numbers from the command line, and echoes the largest number. This program should guard against having too few or too many parameters passed in, and should echo an error message with proper usage if the number of parameters is incorrect

Loops

1. [Read this tutorial on using loops in bash](#)
2. [Read this tutorial on using arrays in bash](#)
3. Write a program called sort.sh that sorts an array of integers in ascending order (this array can be defined at the top instead of read in from command line, you can use any sorting method you wish however you must implement your own sorting functionality) and echoes each number in order.

Tutorials

Vim:

<http://linuxconfig.org/vim-tutorial>

<https://wiki.archlinux.org/index.php/Vim>

Shell:

http://www.gnu.org/software/bash/manual/html_node/

<http://tldp.org/LDP/Bash-Beginners-Guide/html/>

<http://mywiki.woledge.org/BashGuide>