**IS 340 – Operating Systems**

**HOP02D– BASH – Your First Shell Program**

07/10/2019 Developed by Kevin Wang

07/12/2019 Reviewed by Clark Ngo and Bill Kaghan

03/20/2020 Reviewed by Kim Nguyen

10/03/2020 Reviewed by Kim Nguyen

Center for Information Assurance (CIAE) @City University of Seattle (CityU)

**Before You Start**

* This exercise assumes that the user is working with the Ubuntu 18.04 distribution. If you are working with a different Linux distribution, the set of shell commands may vary from those available in Ubuntu 18.04.
* Students will use the EC2 Ubuntu virtual machine that they created in the Module 1 exercise.
* All commands and code discussed in this exercise will run in the Ubuntu console.
* Some steps are not explained in the tutorial**.** If you are not sure what to do:
  1. Consult the resources listed below and experiment in the Ubuntu console and try to solve the problem yourself. (The tutorial will provide reminders.)
  2. If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

Students will be able to:

* Create a script file
* Modify file permissions
* Execute echo and printf commands
* Run a bash script

**Resources**

# Linux command line: bash + utilities

<https://ss64.com/bash/>

(You can use this reference to find descriptions of all the Bash commands that you will use in this and future hands-on exercises. It is recommended that you consult this reference frequently until you are familiar with frequently used commands.)

* Nano/Basics Guide

<https://wiki.gentoo.org/wiki/Nano/Basics_Guide>

**(NOTE: Your directory and file names might be different from screenshots, as they are subject to be used for a different course)**

**Preparation**

1. Connect to your Ubuntu instance (If you have done this, move to step 2):

Open a command prompt

Syntax: ssh -i LOCATION\_OF\_YOUR\_KEY ubuntu@PULIC\_DNS

Example:

>>>ssh -i key.pem [ubuntu@ec2-33-222-101-222.us-west-2.compute.amazonaws.com](mailto:ubuntu@ec2-33-222-101-222.us-west-2.compute.amazonaws.com)

2) Navigate to the folder you just cloned in HOP02C, the path should look similar to below path:

>>> cd IS340/HOP01-YourGithubUsername/Module 2

**Creating, saving, and running the first Bash Script**

1. Create your first script file by typing the following command in the console command line:

>>> echo echo Hello World! > HelloWorld.sh

Note:

We are using the > to send the output of echo command to a file called HelloWorld.sh



1. Run the script file by typing follow command:

>>> bash HelloWorld.sh



1. Add your directory to the path:

>>> export PATH=$PATH: ~/IS340/HOP01-YourGithubUsername/Module 2

[NOTE: the path above is an example, you need to replace YourGithubUsername with your actual Github username]



1. Give the execution permission to your file:

>>> chmod +x HelloWorld.sh



1. Now we can execute the file without typing the bash command:

>>> HelloWorld.sh

**Use nano to edit the file**

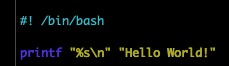
1. Type the following command to create a file with the nano editor:

>>> nano HelloWorld2.sh



You should see an editor UI like this

1. Type the following commands in the editor:



The first line is a shebang, which tells the shell what program to interpret the script with, when executed. In this example, the script is to be interpreted and run by the bash shell.

For the second line, the first part “%s\n” define how to print the text (%s means printing as string. \n means print a line break after the text)

The second part is the actual text that we want to print.

More format interpterion can be found from <https://wiki-dev.bash-hackers.org/commands/builtin/printf>

1. Hit the control + x key to quit the editor. You will see as below:



Then hit y key to confirm. The following message will appear as shown below:



Hit the enter key to save the changes

1. Run the script by typing:

>>> bash HelloWorld2.sh

**Push your work to GitHub**

Run the following commands to push your work to the GitHub repository:

>>> git add .

>>> git commit -m “Submission for HOP2”

>>> git push origin master