

COMP-206 Introduction to Software Systems, Winter 2020

Mini Assignment 3: Advanced Bash Scripting

Due Date Feb 7, 23:55

This is an individual assignment. You need to solve these questions on your own. Use the discussion forum on Piazza if you have any questions. You can also reach out to the course email address, utilize TA/Instructors office hours as necessary. Late penalty is -5% per day. Even if you are late only by a few minutes it will be rounded up to a day. Maximum of 2 late days are allowed.

You MUST use `mimi.cs.mcgill.ca` to create the solution to this assignment. You must not use your Mac command-line, Windows command-line, nor a Linux distro installed locally on your laptop. You can ssh or putty from your laptop to `mimi.cs.mcgill.ca`, or you can go to the third floor of Trottier and use any of those labs to ssh to `mimi` to complete this assignment. All of your solutions should be composed of commands that are executable in `mimi.cs.mcgill.ca`.

Questions in this exercise requires you to turn in scripts. Instructors/TAs upon their discretion may ask you to demonstrate/explain your solution. No points are awarded for commands that do not execute at all or programs that do not compile in `mimi`. (Commands/programs that execute/compile, but provide incorrect behavior/output will be given partial marks.) All questions are graded proportionally. This means that if 40% of the question is correct, you will receive 40% of the grade.

Please read through the entire assignment before you start working on it. You can loose up to 3 points for not following the instructions.

Labs A through D will provide some background help for this mini assignment.

Unless otherwise stated, all the names of the scripts and programs that you write, commands, options, input arguments, etc. are implied to be case-sensitive.

Total Points: 20

Ex. 1 — Customizing your login shell (9 Points)

For this question you will need to:

- Determine which login script your shell uses when you ssh or putty.
- Modify your login script as specified below.
- Comment the login script with your changes.
- Upload the login script as part of your assignment submission.

Modify your login script in all the following ways:

1. **(1 Point)** Display `Welcome to <hostname>!` Here you should replace `<hostname>` with the name of the host to which you login (explore the `hostname` command and see how you can incorporate it's output into this message).
2. **(2 Points)** List the total number of sessions in which you are logged into that particular host (Explore a combination of `who`, `grep` and `wc` commands to accomplish this task). The message should be `You have <X> login sessions to this host`. Replace `<X>` with actual number of sessions that you have logged into that system.
3. **(1 Point)** Alias `comp206` so that typing it executes `cd ~/Projects/COMP206` (the folder you created in mini 1 assignment).

- 4.(2 Points) Set your shell's prompt to include your user name, host name, current directory and time (all dynamic - see sample screenshot below for an expected outcome).
- 5.(1 Point) Execute the **fortune** command to display any random quote.
- 6.(1 Point) Plus one other thing of your liking (make sure to comment this one for the TA).

(1 Point) for writing comments on the script describing the intentions of respective commands/statements.

Below is an expected screen shot of what happens when you login. In this instance, I logged in and got a prompt which has time included in it. I executed the **pwd** and **cd** commands only to show that the next prompt that the shell gives has the time and directory automatically updated. You do not have to do execute these commands in your updated profile.

```
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-66-generic x86_64)
There are 6 users logged in.

Please contact help@cs.mcgill.ca for assistance or if any software is missing.

Last login: Wed Jan 22 11:39:51 2020 from 132.206.51.22
Welcome to teach-vw4!
You have 2 login sessions to this host.
"Any excuse will serve a tyrant."
      -- Aesop
jdsilv2@teach-vw4[11:40:11]:~$ pwd
/home/2013/jdsilv2
jdsilv2@teach-vw4[11:40:20]:~$ cd /etc
jdsilv2@teach-vw4[11:49:49]:/etc$
```

I recommend you start out by implementing/testing each idea in the shell prompt first before making changes to your actual login script.

For this exercise, you should turn in your modified login script.

Ex. 2 — Using shell environment variables(7 Points)

In this exercise, you will learn how to use a set of environment variables that are setup in a “configuration script”, in your main program script, so that your main script can be run without modification by users of your program. You have been provided with a configuration script **a3config** that is a simple text file which contains the environment variables that we will be using. The contents of this file is shown below for discussion. You can use **vi** to edit this file.

```
$ cat a3config
# You may edit this file for your testing.
# You should only change the values of the variables for the purpose of your testing
# However, TAs will use their own a3config file to test your script with.
# Therefore, do not add any other code here that is required for your script's execution.
DIRNAME=/tmp/__206__$LOGNAME/mydir
EXTENSION=msg
SHOW=true
```

Your task is to write a script **finder.sh** that will check if there are files in the directory whose value is stored in the variable **DIRNAME** such that the files have the extension as indicated in the variable **EXTENSION**. If there are such files, you should print a list of them. Further, if the variable **SHOW** has the value **true**, the contents of the file must be displayed to screen.

In order to accomplish this, you will have to execute the contents of **a3config** within **finder.sh**. Explore the **source** or **dot** (.) commands to accomplish this.

If your **finder.sh** script cannot find **a3config**, then it should throw an error message and terminate the script with a non zero exit code.

```
$ ./finder.sh
Error cannot find a3config
$ echo $?
```

1

The `echo $?` command is included above to merely show you the exit code from the script was non zero. If either of the variables `DIRNAME` or `EXTENSION` is not present in `a3config` or if their values are empty, the script should display an error message and terminate with a non zero exit code.

```
$ ./finder.sh
Error DIRNAME and EXTENSION must be set
$ echo $?
2
```

If the directory mentioned in `DIRNAME` does not exist, then the script should throw an error message and terminate with a non zero exit code.

```
$ ./finder.sh
Error directory /nosuchdir/dir1 does not exist
$ echo $?
3
```

If the script cannot locate any files in the directory that have the expected extension, then it should display the following message, but still terminate with a zero exit code.

```
$ ./finder.sh
Unable to locate any files with extension msg in /etc
$ echo $?
0
```

If the script is able to locate files in the directory, and the value of `SHOW` is `true` then it should list the names of the files, along with the contents of each file and terminate with a zero exit code.

```
$ ./finder.sh
/home/2013/jdsilv2/comp206/Mini3/first.msg
Greetings to all of you!
/home/2013/jdsilv2/comp206/Mini3/second.msg
Howdy ?
$ echo $?
0
```

If the script is able to locate files in the directory, but the value of `SHOW` is not set to `true` or if the `SHOW` variable itself is missing from `a3config`, then it should list the names of the files, but do not display the contents of files and terminate with a zero exit code.

```
$ ./finder.sh
/home/2013/jdsilv2/comp206/Mini3/first.msg
/home/2013/jdsilv2/comp206/Mini3/second.msg
$ echo $?
0
```

1. Your `finder.sh` must be executable using the bash shell in `mimi`.
2. (1 Point) For properly commenting throughout the source code.
3. (1 Point) Detecting and displaying an error message and terminating with non zero exit code when the script cannot find `a3config`.
4. (1 Point) Detecting and displaying an error message and terminating with non zero exit code when any of the variables `DIRNAME` and `EXTENSION` are not present `a3config` or is empty.
5. (1 Point) Displaying an error message and terminating with non zero exit code when the directory does not exist.
6. (1 Point) Displaying a message and terminating with zero exit code when the directory does exist but cannot locate any files with the expected extension.

- 7.(1 Point) Displaying only a list of files (not contents) and terminating with zero exit code when the directory does exist and is able to locate files with the expected extension, however the variable `SHOW` is not set in `a3config` or is not set to the value `true`.
- 8.(1 Point) Displaying a list of files and their contents and terminating with zero exit code when the directory does exist and is able to locate files with the expected extension, and the variable `SHOW` is set in `a3config` to the value `true`.

For this exercise, you should turn in the `finder.sh` script that you wrote.

Ex. 3 — Defining functions in shell(4 Points)

In this exercise, you will update the script `showusr` that is given to you (you can use `vi` to edit this file), to write a function `fname`, that accepts a userid (such as your socs unix account) as the argument and sets an environment variable `FNAME` with the value that is the first name of the userid (explore the `pinky` command to get the first name of the userid). The function will also return an integer code to indicate its status (see examples below). Do not use the `exit` command in the function !

- 1.(1 Point) For properly commenting throughout the source code.
- 2.(1 Point) If `fname` is not passed an argument, set the variable `FNAME` to `ERROR` and return a value of 1. Below is an example of me “loading” the function to shell environment and then executing it (with no argument). I am using the `echo` command to check the return code as well as the value stored in `FNAME`.

```
$ . showusr
$ fname
$ echo $? $FNAME
1 ERROR
```

- 3.(1 Point) if the argument passed to `fname` cannot be found among the sessions logged in, it should set `FNAME` to `NOTFOUND` and return 0.

```
$ fname nosuchid
$ echo $? $FNAME
0 NOTFOUND
```

- 4.(1 Point) if the userid passed as argument to `fname` is found among the sessions logged in, it should set `FNAME` to the first name of that user and return 0.

```
$ fname jdsilv2
$ echo $? $FNAME
0 Joseph
```

You should turn in your modified `showusr` script.

WHAT TO HAND IN

Turn in the modified login script, and the scripts `finders.sh`, and `showusr` named properly (3 items). You do not have to zip all of the files together. The files must be uploaded to mycourses under assignment 3. DO NOT turn in `a3config`. TAs will use their own `a3config`.

MISC. INFORMATION

There is a tester script `mini3tester.sh` that is provided with the assignment that you can use to test how your `finder.sh` and `showusr` is behaving. TAs will be using the exact same tester script to grade your assignment. You can test your modified login script by doing a login to mimi as usual.

```
$ ./mini3tester.sh
```

It is recommended that when you start writing your program, test it yourself first with the above examples. Once you are fairly confident that your program is working, you can test it using the tester script.

You can compare the output produced by running the mini tester on your scripts to that produced by the mini tester on the solution which is given in `mini3tester.out.txt`.

You have to implement only the scenarios discussed in the above examples and as required to pass the tester script given to you. You are free to explore for sophisticated approaches, but not obliged to.

FOOD FOR THOUGHT!

The following discussion is meant to encourage you to search independently for creative and optimal ways to perform rudimentary tasks with less effort and/or make your program robust and sophisticated. It does not impact the points that you can achieve in the above questions.

- The command `pinky` can be used to find a person's information if they have a session running in the host. Can you find another command that can be used for the same purpose, even if the person is not logged into the host?