

CIS241 Lab 5: Get Started with Bash Scripting

Introduction to Bash Shell Script

1. Set your own environment variables in start files.

- a) Put the following line at the bottom of the file `~/.bash_profile`:

```
PATH=$PATH:.
```

```
# This will add current directory "." In your PATH variable. Therefore, you don't need to prefix  
# "./" to run a program "a.out" in current directory.
```

Question 01: Can you add the line as `"PATH=.:$PATH"`? What could be the difference?

- b) Put the following line at the bottom of the file `~/.bashrc`:

```
alias rm='rm -i' # interactive rm by default
```

- c) You can make other changes to your environment by locating corresponding commands in those files.
- d) When you have made changes to those files, you can either log out and log in again or run the following commands to make those changes effective.

```
source ~/.bash_profile
```

```
source ~/.bashrc
```

- e) Please test the effects of your changes.

Question 02: How do you test the effects of your changes? Do they work as you expected?

2. Fill in the blank below so as to display the output of the command that **currTime** represents. Run the two commands to verify your answer.

```
currTime=$(date +%r)
```

```
echo "Current Time is: _____"
```

3. Create a directory and do the following in that directory:

- a) Create a text file name **classes** with the following contents:

```
cis162 proj1
```

```
cis163 proj2
```

```
cis162 proj2
```

cis263 proj1

cis163 proj3

- b) Write a Bash script “**readfile.sh**” that reads file classes line by line and displays the first field of each line (i.e., the course numbers) on the screen.
- c) In the body of the loop you just created to do the task in b), add one or more commands to create a subdirectory in your current working directory with the first field of the current input line as its name if that sub-directory does not exist. If you run the script after this is done, you will see three sub-directories in your current working directory, namely, cis162, cis163 and cis263.

You may want to use man command to find the option of mkdir (man mkdir), which does not display an error message if you attempt to create a directory that already exists.

- d) Again, in the body of the loop, add one or more commands to create a file with the second field of the current line as its name in the sub-directory named after the first field of that line. You may want to use **touch filename** to create a file.

#Hints:

The following code will read a line field-wise from file “**classes**”

```
while read -r field1 field2
```

```
do
```

```
    # commands to process field1 and / or field2
```

```
done < “classes”
```

A good reference for utility “read” can be found from the following link:

<https://www.cyberciti.biz/faq/unix-howto-read-line-by-line-from-file/>

- 4. Write a Bash script “**select.sh**” to allow the following pipeline to
 - a) Display the number of regular files (excluding directory files) in your current working directory if the script is run with the option “**f**”:

ls | bash select f | wc -l

- b) Display the number of directories (excluding regular files) in your current working directory if the script is run with the option “**d**”:

ls | bash select d | wc -l

Note: Counting the number of regular files or the number of directories is done by command wc -l. You do not want to do it in the **select.sh** script. Also, you want to make sure that the **select.sh** script uses input data from the **ls** command.

Hint:

The following code will read from the stdin

```
while read x
```

```
do
```

```
    # command to process data x
```

```
done
```

When all the above are done successfully, show them to your instructor. If you cannot demo your work with your instructor, submit it through Black Board by due time, including the answers to Question 01, Question 02 and Bullet 2 as well as the scripts **readfile.sh** and **select.sh**.