

ECE 364 Lab 01 Handout

Conditionals, Loops, and Basic Commands in Bash

August 21, 2017

Passing this lab will satisfy course objective CO1.

Instructions

- Work in your Lab01 directory
- Copy all files from `~ee364/labfiles/Lab01` into your Lab01 directory. You may use the following command: `cp ~ee364/labfiles/Lab01/* ./`
- Remember to add and commit all files to SVN. **We will grade the version of the file that is in SVN!**
- Name and spell your scripts exactly as instructed. When you are required to generate output, make sure it matches the specifications exactly. Your scripts may be graded by a computer.

Tiny Shell (40 pts)

Introduction

Your task is to write a mini version of a shell that can execute a small subset of commands.

Implementation Details

Your script, `tiny_shell.bash`, should repeatedly ask the user what command they would like to execute, with the prompt "Enter a command: ". The list of commands available to you, and the expected responses are given below:

- `hello`
Prints "Hello <username>" to the terminal.
- `quit`
Prints "Goodbye" and exits.
- `compile`
Attempts to compile all files ending in ".c" in the current directory using the command `gcc -Wall -Werror <somefile.c> -o <somefile.o>` such that ".o" binaries with the same basename are created. For each compilation, you must check the return code of the gcc command and print a message indicating if the compilation succeeded or failed for that file.
- `whereami`
Prints the pathname of the current working directory.
- If the given command is none of the above, print an appropriate error message.

Sample Output

```
$ ./tiny_shell.bash
Enter a command: hello
Hello ee364ta

Enter a command: compile
Compilation succeeded for: a.c
cc1: warnings being treated as errors
bad.c: In function main:
bad.c:2: error: implicit declaration of function echo
Compilation failed for: bad.c
Compilation succeeded for: b.c
Compilation succeeded for: works.c

Enter a command: whereami
/home/ecegridfs/a/ee364ta/Lab01

Enter a command: someCommand
Error: unrecognized input

Enter a command: quit
Goodbye
```

Mobile Games (40 pts)

Introduction

Purdue has conducted a survey recently to find out the number of hours in a day each student is spending on six mobile games Angry-Birds-2, Candy-Crush-Saga, Clash-of-Clans, Hill-Climb-Racing-2, Subway-Surfers, Temple-Run.

In your Lab01 directory, you are provided with a file called `data.txt` that contains a list of the names of all participants, along with how many hours each spent in a day on a given mobile game. The format of each line of the file is as follows:

```
<participant's name>,<mobile game>,<number of hours in a day>
```

Implementation Details

You have been assigned the task of collecting results from `data.txt` and printing out statistical data for each game, outlined in the requirements below. You decide to use your newfound knowledge of Bash to write a script called `game_stats.bash` that meets the given requirements.

1. The script should accept two arguments:
 - (a) Name of the data file
 - (b) Name of the mobile game
2. If the correct number of arguments are not provided, print an appropriate message and exit with a return code of 1.
3. If the first argument is a non-existent file, print an error message and exit with a return code of 2.
4. Print the total number of students who plays the given mobile game.
5. Print the total number of hours spent on a given mobile game.
6. Print the name of the student who spent the highest amount of the time in a given mobile game, as well as the number of hours spent.
7. Print the name of the student who spent the least amount of the time in a given mobile game, as well as the number of hours spent.

Sample Output

Note: Your output must match the sample output exactly. Your script may be tested with a different data file.

```
$ ./game_stats.bash
```

```
Usage: ./game_stats.bash <file> <game>
```

```
$ ./game_stats.bash someFile Temple-Run
```

```
Error: someFile does not exist
```

```
$ ./game_stats.bash data.txt Angry-Birds-2
```

```
Total students: 48
```

```
Total hours spent in a day: 107
```

```
Harry Mitchell spent the highest amount of time in a day: 7
```

```
Tim Mahoney spent the least amount of time in a day: 1
```

```
$ ./game_stats.bash data.txt Subway-Surfers
Total students: 59
Total hours spent in a day: 134
John B. Shadegg spent the highest amount of time in a day: 8
Connie Mack spent the least amount of time in a day: 1
```

```
$ ./game_stats.bash data.txt Temple-Run
Total students: 50
Total hours spent in a day: 114
Brian Holtz spent the highest amount of time in a day: 10
Eddie Adams spent the least amount of time in a day: 1
```