Assignment 3: bash scripting exercise 1: `logincount`

Eventually, you will want to automate repetitive tasks to run in your terminal instead of having to type and execute every command manually. At other times, you only want certain commands to execute based on your specified conditions or on a set of items.

You will be able to do this by ***bash scripting***. When you write a bash script, you are really writing a program in the `bash` ***programming language.*** `bash` is not just a shell, but a programming language as well, and a bash script can just as well be called a *bash program.* It is mentioned now because very soon you will begin writing programs in another programming language, *Python;* this is the first in a sequence of small steps to eventually master programming in Python.

We have been calling your programs *shell scripts.* A script is a program, make no bones about it. Scripts are programs written in a ***scripting language,*** which is a special kind of programming language. All scripting languages are programming languages, but not *vice versa.* The distinction will be explained in a later lecture. In this case, `bash` is both a programming language and a scripting language.

Instructions

The `last` command lists information about who has logged into the computer on which it is run. In particular, it has a column with the username, the terminal on which the user was connected, the Internet address (the IP address) from which they connected to the computer, and the date and time that they logged in and then logged out if they did. If they logged out it also displays the total time they were logged in. For example, this is an entry for Dr. Weiss, whose username is `sweiss` when he is logged into `cslab12`.

```
sweiss pts/11 146.95.214.131 Thu Sep 14 13:05 - 14:27 (01:22)
```

If the username is too long, it is truncated. The truncation of usernames in the standard output of `last` can be avoided with an option to display the full username displays the full username in that column – you will need to read the man page for the `last` command to understand how it works.

Write a bash script named `logincount` that takes a list of usernames as its command line arguments and displays on the screen, for each user name, a message of the form

```
Number of times that <username> logged into this machine is <N>
```

where <N> is to be replaced by the number of records that the last command output that match <username> exactly.

For example, if I execute the script

```
./logincount sweiss Trami.Dang67 fake.student
```

it should output something like

```
Number of times that sweiss logged into this machine is 7
Number of times that Trami.Dang67 logged into this machine is 21
```

Notice that if a username given, as an argument, is not a valid username, *nothing is printed for that name.*

Valid usernames: it is well worth the effort to spend some time exploring everything in the `student.accounts` directory. Learn the absolute pathname of this directory.

- Did you notice that it is the parent directory for each user on the Linux Lab?
- Recall from Assignment 1 that each user receives their own home directory in the fiiesystem. For each Linux Lab account, the username is used as the name of the user's home directory.

The validity of usernames can be checked with an expression to only display the message if a user has a home directory within `student.accounts` – you will need to read the man page for the `test` command to understand how it works.

On the other hand, if no usernames are given when running logincount, it is considered a ***user error,*** and the command should display an error message to the user instead:

```
Usage: logincount <list of usernames>
```

Grading Rubric

This assignment is graded on a 100 point scale.

The script will be graded on its correctness foremost. This means that it does exactly what the assignment states it must do, in detail. Correctness is worth 70% of the grade. Then it is graded on its clarity, simplicity, and efficiency, worth 30% of the grade.

The objectives when writing any script, are

- *clarity* – it should be easy to understand by someone with a basic knowledge of UNIX
- *efficiency* – it should use the least resources possible
- *simplicity* – it should be as simple as possible

Submitting Requirements

- **Due date:** Sunday October 4, 11:59PM Eastern Standard Time
- Late submissions will be docked total points at the rate of 1 point for every day it is late. No submissions will be accepted after Sunday October 11, 11:59PM Eastern Standard Time.
- Accepted formats:
    - Screenshot image(s) in JPEG, PNG, or TIF of the script, its execution for a sampling various input and the resulting output of each execution.
    - A zip archive file containing the `logincount` script.
      The steps to create a zip archive and to secure-copy files is outlined in the tutorials called '`zip` command for beginners' and '`scp` command for beginners', that are located in the Course Materials section on the course Blackboard.