CPSC 4240

Stuart Jackson

Bash Questions

Question 1:

Reference 3 is the Bash Guide for Beginners. It includes explanations of the basic features of Bash and Bash scripts as well as serving as a general guide for new users. There are sections on what Bash is, on writing and debugging scripts, the Bash environment and how to work in it, regular expressions and how to use them, and several sections detailing specific techniques common in Bash scripting.

Reference 4 is the Advanced Bash-Scripting guide. This document serves several functions including: teaching new users the basics of shell scripting, teaching more advanced levels of scripting, usage as a textbook, a reference, and general resource for scripting. It includes sections on most topics covering scripting, from the very basic to the much more advanced.

Reference 5 is a reference manual for Bash. It includes a list of the features of shells in general, and Bash in particular. It includes a list of commands, ways to combine commands, parameters, and other general functions of a shell. There are also sections of command line editing, job control, how to install Bash, and differences between Bash and Bourne. The document is meant to serve as a reference, so focuses less on teaching the material and more on simply presenting it in a clean fashion.

While similar, each of the three references has a different focus. Reference 5 focuses on usage as a reference, plainly stating the various features of Bash and shell scripting. Reference 3 is a beginner's guide, focusing on instructing new users on the basics of Bash scripting. Finally, reference 4 is also a beginner's guide, but also serves to instruct more advanced concepts and techniques as well as serving as a reference.

Question 2:

Single quotes preserve the literal value of all the characters between them.

Ex: **echo 'Try something like \"this\"'**

Try something like \"this\"

Double quotes preserves the literal value of all characters between them, except for the dollar sign, backticks, and the backslash (when followed by a dollar sign, backtick, double quote, backslash, or newline.)

Ex: **echo "Try something like \"this\""**

Try something like "this:

Backticks take the surrounded characters and calls them as a command.

Ex: **echo `date`**

Thu Feb 23 12:1727 EST 2017

Question 3:

p='fc -s'

fc is used to find previously executed commands, and the -s flag is used to find commands that begin with certain characters. This simplifies the process, so that you only need to type 'p' and the desired starting characters.

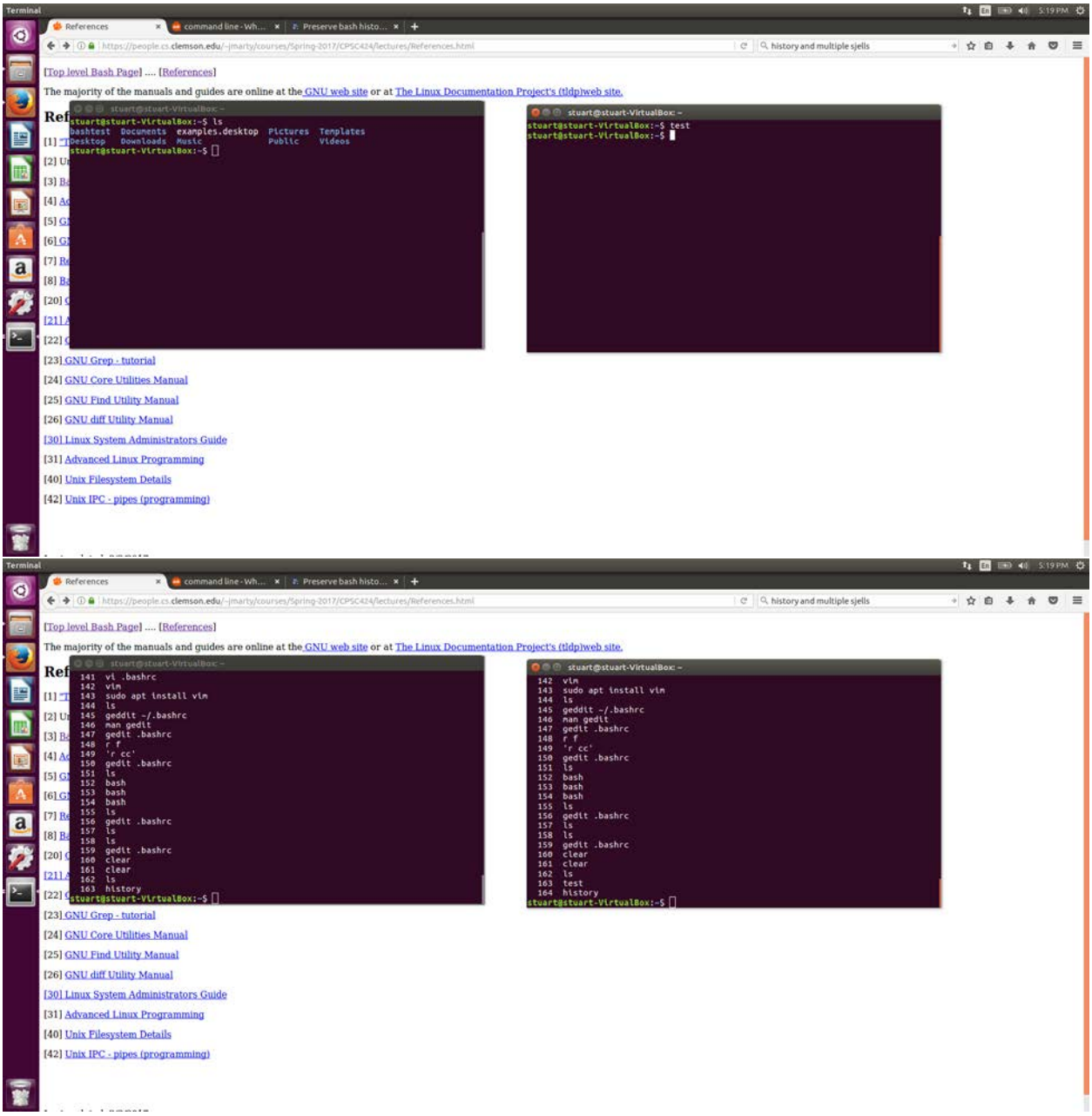
Searching online, I found the following fix added to .bashrc:

export HISTCONTROL=ignoredups:erasedups

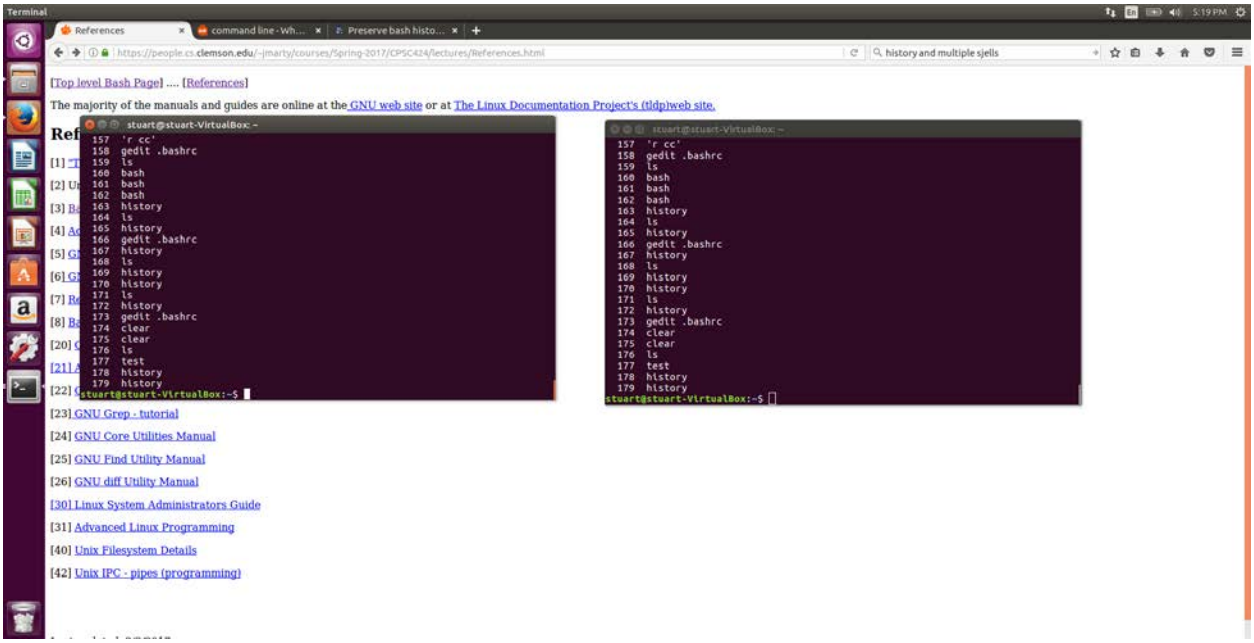
shopt -s histappend

export PROMPT_COMMAND="${PROMPT_COMMAND:+$PROMPT_COMMAND$'\n'} history -a; history -c; history -r"

These commands, when added to .bashrc, help with using history between terminals, though it is not without issues.


Terminal
References
command line - Wh...
Preserve bash histo...
https://people.cs.clemson.edu/~jmarty/courses/Spring-2017/CPSC424/lectures/References.html
history and multiple sjells
[Top level Bash Page] .... [References]
The majority of the manuals and guides are online at the GNU web site or at The Linux Documentation Project's (tldp)web site.
stuart@stuart-VirtualBox: ~
stuart@stuart-VirtualBox:~$ ls
bashtest Documents examples.desktop Pictures Templates
Desktop Downloads Music Public Videos
stuart@stuart-VirtualBox:~$
stuart@stuart-VirtualBox: ~
stuart@stuart-VirtualBox:~$ test
stuart@stuart-VirtualBox:~$
[23] GNU Grep - tutorial
[24] GNU Core Utilities Manual
[25] GNU Find Utility Manual
[26] GNU diff Utility Manual
[30] Linux System Administrators Guide
[31] Advanced Linux Programming
[40] Unix Filesystem Details
[42] Unix IPC - pipes (programming)

Terminal
References
command line - Wh...
Preserve bash histo...
https://people.cs.clemson.edu/~jmarty/courses/Spring-2017/CPSC424/lectures/References.html
history and multiple sjells
[Top level Bash Page] .... [References]
The majority of the manuals and guides are online at the GNU web site or at The Linux Documentation Project's (tldp)web site.
stuart@stuart-VirtualBox: ~
141 vi .bashrc
142 vim
143 sudo apt install vim
144 ls
145 geddit ~/.bashrc
146 man gedlt
147 gedit .bashrc
148 r f
149 'r cc'
150 gedlt .bashrc
151 ls
152 bash
153 bash
154 bash
155 ls
156 gedlt .bashrc
157 ls
158 ls
159 gedlt .bashrc
160 clear
161 clear
162 ls
163 history
stuart@stuart-VirtualBox:~$
stuart@stuart-VirtualBox: ~
142 vim
143 sudo apt install vim
144 ls
145 geddlt ~/.bashrc
146 man gedlt
147 gedit .bashrc
148 r f
149 'r cc'
150 gedit .bashrc
151 ls
152 bash
153 bash
154 bash
155 ls
156 gedit .bashrc
157 ls
158 ls
159 gedit .bashrc
160 clear
161 clear
162 ls
163 test
164 history
stuart@stuart-VirtualBox:~$
[23] GNU Grep - tutorial
[24] GNU Core Utilities Manual
[25] GNU Find Utility Manual
[26] GNU diff Utility Manual
[30] Linux System Administrators Guide
[31] Advanced Linux Programming
[40] Unix Filesystem Details
[42] Unix IPC - pipes (programming)



In the above example I have previously run several commands, then run two different commands in two separate terminals. When history is called the first time in both terminals the listings are incorrect, but after calling it once or twice more they match.

Question 4:

Head -n50 test.txt | tail -n26 >mySmallerFile.txt

Question 5:

Lines 1-5 do not work because the -eq flag requires a number, “apple” which is the value of “myString2” is not a proper number so it causes an error and is considered NOT to work.

In both cases for lines 9-12 a is set to 273, an integer, which allows proper comparison with $a which is also an integer. Even though one possible statement has 273 surrounded in double quotes, this number is interpreted as an integer.

Setting the variable ‘a’ to 273.0 does not work, because the variable is compared to 273, an int, the variable must also be an int in order to properly compare the two.

Question 6:

/[0-9][0-9][0-9][\s-]+[0-9][0-9][0-9][\s-]+[0-9][0-9][0-9][0-9]/