CIT 371 Lab 2: Learning the Bash Shell

You may use the Web Console, SSH or PuTTY to complete this lab assignment. See the “Student VM Access and Help Requests” Document for information on accessing your VMs.

In this lab, you will explore some of the features in the Bash shell. Power on VM1 (CentOS6) from Coivcenter. Open a terminal window. Commands are shown in **courier bold font**. Answer any *questions asked in italics*.

1. Most Linux commands have a number of options available to help you control the characteristics of the program. We start with commands without using options.
   1. Enter **ls**. *What does this command do?* Enter **pwd**. *What does this command do?* 
      1. The ls command outputs a list of the directories. The pwd command is the print working directory, which shows the pathway and directory being used in.
   2. Type **cd ~**. *What does this command do? How does your prompt change?* Type **ls** again. *How does this differ from part a?*
      1. This command allows you to change directory to the home directory. My current prompt changed to, [Student@centOS6Template ~] $. This differs from part a because we are asking for the listing of the directories is in the home file
   3. Type each of these commands and *briefly explain what each does*. **whoami,** **who, arch, cal, uname, date, vmstat**. NOTE: you can learn more about commands using **man *command*** as in man uname. This brings up the “manual” page. Use this if a command’s response does not tell you enough about what the command does.
      1. Whoami- this command tells you the username in which you are working in
      2. Who- the who command gives the list of the users who are working on the machine you are working on.
      3. Arch- is used to print the computers architecture
      4. Cal- provides a calendar
      5. Uname- prints basic system information, in this case its Linux
      6. Date- provides the date and time
      7. Vmstat- will distribute monitoring commands such as the process, memory, swap, io, system, etc.
   4. Type **cd ~/FILES** followed by **more addresses.txt** and **less addresses.txt**. Type **more /etc/passwd** and **less/etc/passwd**. *How do the two instructions, more and less, differ?* Hint: there is a difference at the bottom of the window as you wait to scroll further down by pressing <space> or <enter>.
      1. The two instruction differ with the more and less by the more option providing the text of the document in the terminal, compared to the less which provides the whole doucnemnt and have to press q to exit out of to go back to the terminal for commands. When typing the more and less for the /etc/passwd we can see that the less command provides us with little text information with a % of what’s left compared to the more which will show the whole text and will state end.
   5. To locate a program, you can use which and whereis. Type **which arch**. *What is the response?* Type **which vmstat**. *What is the response?* Type **which useradd**. *What is the response?* Type **whereis useradd**. *Why did you get a response when using whereis but not which?*
      1. The response to which arch is, /bin/arch.
      2. The response for which vmstat is , /usr/bin/vmstat
      3. The response for which useradd is, /usr/bin/which: no useradd
      4. You get a response to whereis instead of which because you can’t ask which user to add there is only an option for where is the user add. You can locate where its at you cant locate which or what useradd is.
2. Most Linux commands permit options, which usually appear as –*char* as in –a.
   1. Type **ls**. Now type **ls –l**. *How do the two differ?*
      1. The ls command will show you the list of the directory you are in which is the FILES. While the ls -l command will give you a long listing of the directory which will provide time last modified or created, the size, the date, the permissions, and much more compared to just the name in the ls.
   2. Type **host**. As this *command* expects parameters of at least a hostname, you are shown its usage. Along with the hostname (e.g., www.nku.edu), you can also provide a number of options like –i (reverse IP lookup) or –t *type* to specify a query type. Type **host www.nku.edu**. *What is the response?* Type **host –v www.nku.edu**. *What does the –v option do?* Look at the response from the command, *what sections do you see?* *How much information did you receive (look for the number of bytes returned) and how long did it take?* The IP address given at the bottom of this command is the DNS server that sent you the information on www.nku.edu. Type **host –i** *IPaddr* (where *IPaddr* is the IP address of the DNS returned from the previous host –v command). *What is the name of the DNS server’s IP alias?* As noted above,–i performs a “reverse lookup”. *What does that mean?*
      1. The response to the host [www.nku.edu](http://www.nku.edu) is, “ is an alias for wwwserv4.hh.nku.edu, has address for 10.2.109.51. When typing the -v option this simply provides the version of the host also this enables the verbose output. The sections I can see when doing the command host -v [www.nku.edu](http://www.nku.edu) is the header, question, and authority sections. The information received was 102 bytes returned in 0 ms. The DNS server’s Ip alias is IP6.INT. The reverse lookup means we are able to look up an IP address from a host compared to a host looking up the IP address.
   3. Type **ps**. This command shows you the status of all active processes. There are a number of options available. For ps, most of these options do not start with a hyphen. Type each of these: **ps a, ps u** and **ps aux**. *How do the responses differ?*
      1. The ps a will show all the active process compared to the ps u, which will show just you and your active process. While the ps aux will show every active process on every system no matter if its you as the Student or root.
3. As noted earlier, commands all have man (manual) pages that tell you about how to use the command, including the available options. When viewing a man page, use the space bar to move down screen-by-screen, enter to move down one line, or the up and down arrows. Use man to answer the following questions.
   1. ps: *What does this man page tell you about ps? Provide a short (2-3 line) summary.*
      1. This provides a snapshot of the current process. The man page tells us what options we have to work with, with the ps command. We can also see descriptions and how we can use the man with said options such as the aux option.
   2. ls:  *what does –a do? what does –r do? what does –R do?*
      1. -a will allow ALL entries to be shown,” does not ignore entries”
      2. -r wil reverse the order while sorting
      3. -R will list the subdirectories recursively.
   3. who: *what does –b do? what does –H do?*
      1. -b is a boot, and will provide time of the last system boot
      2. -H is heading, and will print the line of the column heading.
   4. vmstat: *what does this instruction do? there are a number of synopses, what are these?* 
      1. This instruction allows for a report of virtual memory statistics.
      2. Vmstat -a -n -t -s -f -P -V -f, and many more.
4. The history list is a useful feature that allows you to easily recall a previous command. History is used for two reasons, to allow you to reuse a command without having to type it in again and so that you can recall a complex command that you may not precisely remember.
   1. Type **history**. *How many commands are in your list?*
      1. 1038
   2. To recall the most recent instruction, type **!!**. Type this. *What happens? Why?*
      1. This recalls the most recent instruction, there was no recent instruction so I did an ls command to see and make sure the !! worked. It didn’t work originally due to pulling up the history command there was no prior instructions.
   3. To recall a previous instruction, type **!*#*** where *#* is the number in the history list. Type **!1**. *What happened?* Type **!*n*** where *n* is the number of instruction prior to the first history command (this should correspond to the man instruction from 3d).  *What happened?*
      1. Bash !1: event not found
      2. Bash !n: event not found
      3. NOTE: I don’t know if this is what you are looking for but I have messed around trying to figure out the ! commands.
   4. To recall a previous instruction, you can also type **!*str*** where *str* is a string of the beginning of the command. Type **!m**. This should recall your last man command. Type **!p** and it should recall your last ps command (from 2c). Type **!ho**. *What command is executed?* *Why did you use !ho instead of !h?* 
      1. The !ho brought me back to the host -v option. We used !ho instead of !h because the !ho will bring you back to the host -v and the version compared to !h which will just provide the size and the display of the file.
   5. Type **history** again. How much longer is your list now than it was in step 4a?
      1. It is now 1048, which is 10 steps longer then in 4a.
5. History is only one form of shortcut available in Bash. Another is the use of the alias. An alias is a command that you write which stands for another command. You use aliases primarily to shorten a longer command, or to provide a replacement for a command you often mistype.
   1. To see what aliases are already defined, type **alias**. *How many aliases are there?*
      1. There is 6 alias.
   2. Define your own alias with the notation alias name=command. If the command on the right side contains one or more blank spaces, enclose the command in ‘’ marks. Try this one: **alias home=‘cd ~’**. Type **home** to make sure it works. *Explain the command and why we called it home.*
      1. This command will easily allow us to go back to our home directory. The “~” means home in linux and so we just made a shortcut to go home, naming it home.
   3. Enter an alias which will perform a long listing of the current directory showing all files in reverse alphabetical order. Use man to identify the proper options. Test it out to see if it works correctly. *What alias did you create?* 
      1. Alias reverse=’ls -l-r” . We created a reverse listing of the directory.
   4. Enter an alias which will output your current working directory, your user name, the architecture, and the date. You can specify multiple instructions in one command by separating each instruction with a semicolon (;). *What alias did you enter?*
      1. Alias basics=’pwd;whoami;arch;date’
   5. You can remove a defined alias by typing unalias *name* where *name* is the name of the alias. Type **alias ..=‘cd ..’** and type **..** to test it. Now type **unalias ..** Type **alias**.  *Is the***..** *alias listed?*
      1. No, the .. is no longer listed.

When done, remember to save your answers and submit them. Your lab submission will be this updated lab manual with your answers embedded in this document. Please also highlight your answers in yellow. If using a university computer, log out. If you are not on campus, disconnect from the VPN.