**Lab 2. UNIX commands.**

**Purpose and rationale**

The purpose of this lab is to quickly get students up to speed with basic usage of the UNIX development environment, as a preparation for all future lab activities.

**Lab environment**

All work should be done on a machine in the department's Linux cluster. Refer to our Lab1 directions. You should use **atoz, sp1, sp2** or **sp3, instead of athena**.  This is practice for a task later in the semester.

[**NOTE**: I accessed all four by logging into “athena”, typing “ssh atoz” or “ssh spX”, typing “yes”, and re-entering my password.]

**Part 1. Introductory UNIX lab2 (also known as *Give-Linux-some-time*):**

1. Read/browse the **man** pages for the Shell commands listed below in #6
2. To view the manual for the command “script”, type **man script**.
3. Use the space bar to scroll through the display from **man**.
4. Type **q** to quit each session.
5. **SEARCH**. Some of the commands below will show up as CSH\_BUILTINS or BASH\_BUILTINS
   1. In this mode, the needed information is somewhere in a big display.
   2. Example: When doing a **man history**, you get more information that you expect.
      1. Type **/history** to **search** for the word “history” and see occurrences of that word.
      2. Typing an **n** will take you to the next occurrence.
6. Check out the “man” pages for the following two columns of commands.

script

man (note the standard sections of

the manual i.e 1, 2, 3 ...)

who (also try w)

gcc

touch

top

mkdir

ls

ps

cd

file

cat

wc

grep

cp

diff

rm

history

jobs

make

ssh

head

tail

logout

vi

view (*This one is buried in the* ***vi***

*page…search for it.)*

exit

* more directions on next page

**Part 2. Prepare a script to show your work:**

Run the script command to make a script of your terminal session. At the prompt,

type: **script StudentName\_lab2.txt**

Practice the Shell commands using the below list.

* The Part 1 instruction to read/scan the various commands is **not** to be included in your script file.
* At the end of the practice session, please be sure to exit script session with **exit** command.
* If you need to leave the script before you are finished,

re-open the script and append to it bytyping: **script –a StudentName\_lab2.txt**

* You might run into errors while executing these examples. Look at the errors and see if they make sense. Correct the issues if possible and rerun the commands, and then just keep going.
* Follow the commands as listed below. The occasional extra command (like **ls** , **pwd,** or **cd**, for example) is just fine.
* If you incorrectly type a command, then backspace to fix it, it might look something like this:

**[bielr@sp1 lab2]> wc [K[K[Kgrep lab file1**

Do not try to edit your script file. Just leave it as is. I know what it is.

**history -c C**lear the previous history so your script is not a mile long.

**cd csc60** Move to your directory for this class.

**mkdir lab2** Make a directory named lab2.

**cd lab2** Move to directory lab2.

**pwd** Print current working directory (lab2).

We will be moving back and forth between csc60, lab2, and aaa.

**mkdir aaa** Make a new directory aaa

**cd aaa**  Change current directory to aaa

**pwd** Check that you moved from one directory to another.

**cd ..** Change to upper directory

**pwd** Print current working directory. You should be back in Lab2

**ls > file1** List directory content and redirect output to a file called "file1"

**cat file1** Display text content in file1

**less file1** Like *cat* but paginated

**q** To quit the **less** command

**file \*** Check file types of all files

**wc file1** Print newline, word, and byte counts for file1

**wc \*** Word count all files in directory

**grep lab file1** Find word *lab* in file1.

**cp file1 file2** Copy file1 to a new file2

**ls** Check that you have both files

**cd aaa** Move one directory below Lab2.

**cp ../file1 .** Copy file1 from directory above to here. *Note the* **space-dot** *at the end of*

*the command.*

**🡪 more commands on next page 🡪 🡪 🡪**

**ls** Check that you got file1 here.

**mv file1 file2** Rename file1 to file2

**ls**  Check to see that file 1 changed to file 2

**mv ../file1 .** Move file1 from directory above to here. *Note the* **space-dot** *at the end of*

*the command.*

**cd .**. Move up to Lab2

**ls** Check that you now have file2 here.

**cd aaa** Move back down to directory aaa

**ls**  Check that aaa still contains both file1 and file2.

**cmp file1 file2** Compare file1 with file2, show differences. Same file so no

differences.

**ls > aaalist** Create a different file

**cmp file1 aaalist** Now compare two files known to be different

**diff file1 aaalist** Like cmp except shows more info

**rm file1** Remove file1. Answer the delete prompt with: **y**

**ls** Verify its removal.

**ps -u** Show all user's running Process ID's

**ps -l** Show processes (lower case L) (including Process ID Parent Process ID)

**!!**  Repeat previous command

**history** A list of the commands you have done.

**Two** choices here to get from aaa to csc60:

**(1) cd ..** Move up a directory to lab2.

**cd ..** Move up a directory to csc60.

**or do one command instead of the two “cd ..” commands.**

**(2) cd ../..** Move up past lab2 to csc60.

**pwd** Print current working directory. You should be back in csc60

**cd lab1** Move down to the lab1 directory

*[****NOTE 2:*** *The commands* ***below*** *will require that you be on in the directory where* ***lab1.c*** *resides. If you path differs, you still need to move to the directory where your* ***lab1.c*** *file resides, and then try these commands.]*

**head lab1.c** *or* **head -20 lab1.c** List first 10 or 20 lines of code

**q** Type q to quite the application

**tail lab1.c** *or* **tail -20 lab1.c** List last 10 or 20 lines of code

**q** Type q to quite the application

**ls -al | less** Directory listing (too long) 'piped' to 'less' for viewing

**q** Type q to quite the application

**history**  History of commands given

**Quit the script session. *[Note 3:*** *The script ends when the forked shell exits: (a control-D to exit the* ***Bourne shell*** *(sh(1)), and exit; logout or control-d (if ignoreeof is not set) for the* ***C-shell****, csh(1)).To determine what shell you are in, type:* ***echo $SHELL***

**exit** Exit your login on sp1, sp2, sp3, or atoz.

**Deliverables**

Please upload your Lab 2 script file (**StudentName\_lab2.txt**) to Canvas.

**Note to folks with their own UNIX/Linux machines**:

I expect you to do the above assignment. I expect to see the “history” command. If you feel it invades your privacy, then you have three choices:

1. Log off and back in to start a fresh new session;
2. At the prompt, type: **history -c** which will clear the command history of your computer
3. Do your work on *athena* like everyone else.

…