3. Exercises  
1) look   
(1) Run command ‘look excl’ to see the output. What does this command do?

=> - it shows file/usr/dict/word , means there are no such ‘file and directory’

(2) Create a file named myfile with four lines as follows. Then run ‘look excl myfile’ to see output. Write   
down the output and figure out what this command do?  
 Exclamation  
 Exclamation  
 exclamation

trying this

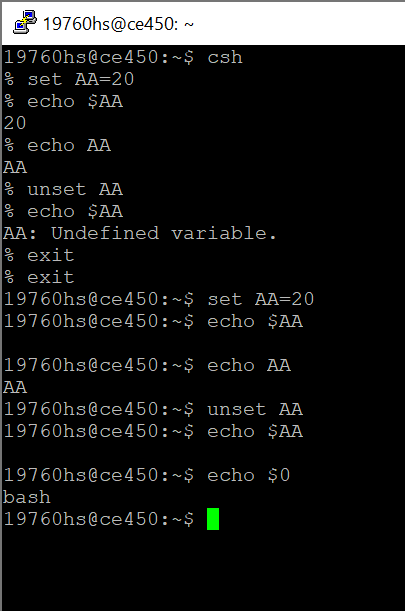
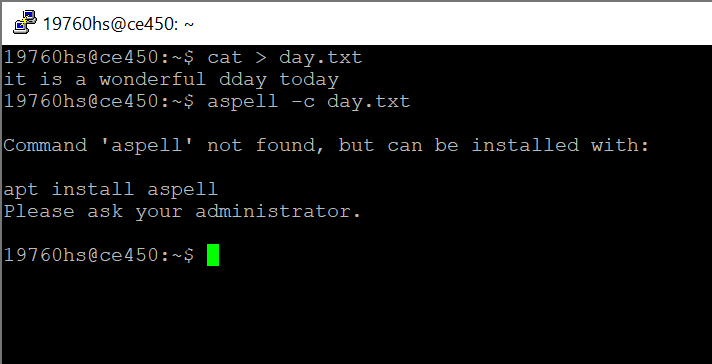
=>output – Exclaimation

=> this command prints the first string from the content of file ‘myfile’

(3) Type in the command ‘look air | grep e | wc -w’ to see the output. What does this command do?   
Explain from the output.

* Type in ‘look air’,
* Then type in grep e myfile – output is exclamation
* Then type in wc -w myfile – output is 5 myfile
* 2) spell

(1) Create a file with the following contents. Then use spell to list all of the misspelled words.  
 Write down your command and output.  
 It is a wonderfull dday today.

* day.txt file created using ‘ cat > day.txt’, written above given content.
* Used control d to save it
* Then enter aspell -c spelling. Output- command not found
* 

3) set and unset.

(1) Enter C shell. How do you do it? Write down the command.

=> type in command – ‘csh’

(2) Create a local variable AA by ‘set AA=20’.  
 This command instructs the corresponding shell to place in memory the variable named AA with a   
 current value of 20.

=> local variable created using command ‘set AA=20’

(3) Run ‘echo $AA’ to see the output. Write down the output.

=> output – 20

(4) Run ‘echo AA’ to see the output. Write down the output.

=> output – AA

(5) What is the difference between (3) and (4)?

=> the difference between no.3 and no.4 is no.3 gives the value of AA but no.4 just printed AA

(6) Run ‘unset AA’ then ‘echo $AA’. What will be output? So what does ‘unset’ command do to   
 variable AA?

=> command unset – clear the variable

=> echo $AA – undefined variable

(7) Change shell to bash. Write down the command you use.

=> To change the shell use command- cat /etc/shells.

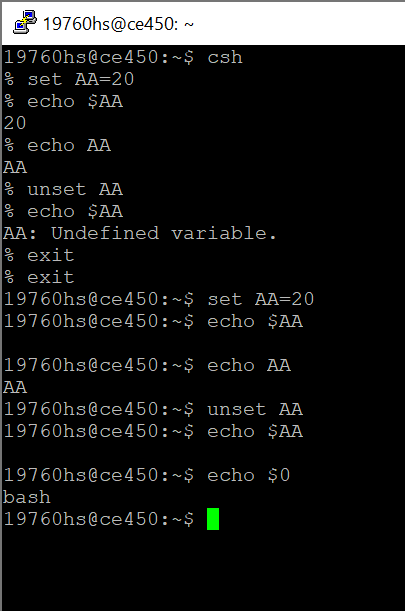
(8) Create a variable by ‘AA=200’. Then do the same procedure from (3) to (6).

=> echo $AA – gives nothing

=>echo AA – print only AA

=> type in unset AA to clear the variables

=> then type in echo $AA- output- nothing



4) setenv and unsetenv

(1) Go to C shell. Type the following commands to display all environment variables and their values   
under UNIX-like operating systems:  
env

Clearecho



(2) Run ‘echo $HOME’, ‘echo $OSTYPE’, ‘echo $HOST’, and ‘echo $USER’ to see the output. What   
each one is set to and representing

echo $HOME= /home/19760hs

echo $OSTYPE=undefined variable

echo $HOST = undefined variable

echo $USER = username- 19760hs

(3) Type in ‘setenv PS $HOME’. Then ‘cd $PS’ to go to home directory.

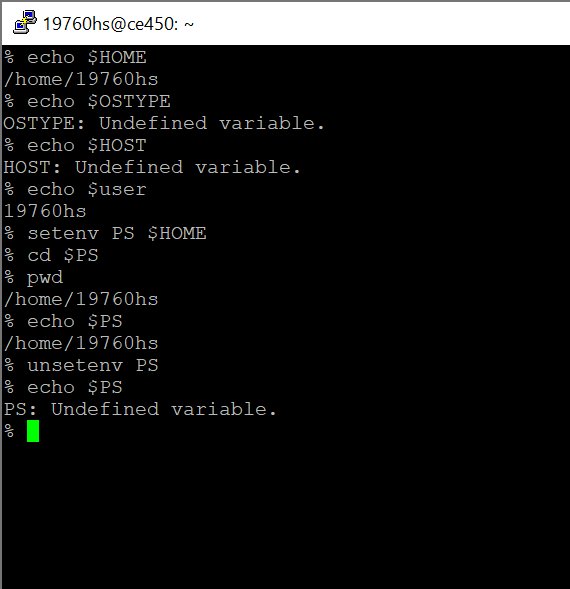
* Setenv PS $HOME : cd $PS
* Then type in ‘Pwd’ for home directory

(4) Type in ‘unsetenv PS ‘. Then ‘echo $PS‘ to see the output.

=> Unsetenv PS

=> Echo $PS

=> output shows PS is an undefined variable



5) export

(1) Change shell to bash. Create a variable named a by ‘a=200’.

=> typed in exec/bin/bash to change shell to bash

=> variable created using set a=200

(2) Enter ‘echo $a’ to see the output. What is that?

* Typed in ‘echo $a’
* Output = 200

(3) Then enter a subshell by ‘bash’. Run ‘echo $a’ again. What is the output? Why is the output like   
 that?

=> the local variable can’t access the local variable

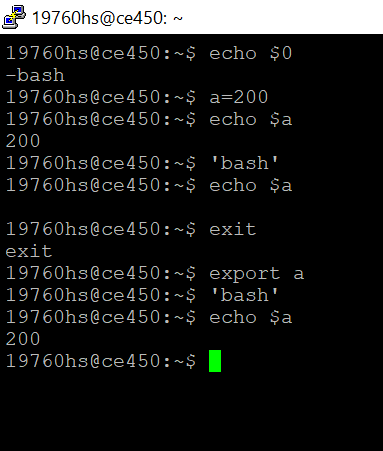
(5) Go back to the previous shell by ‘exit’ then run ‘export a’. Enter a subshell by ‘bash’and run   
‘echo $a’. So what is the effect of ‘export’ command?

=> typed in ‘exit’ , then run ‘export a’

=> after that entered ‘bash’ and run ‘echo $a’

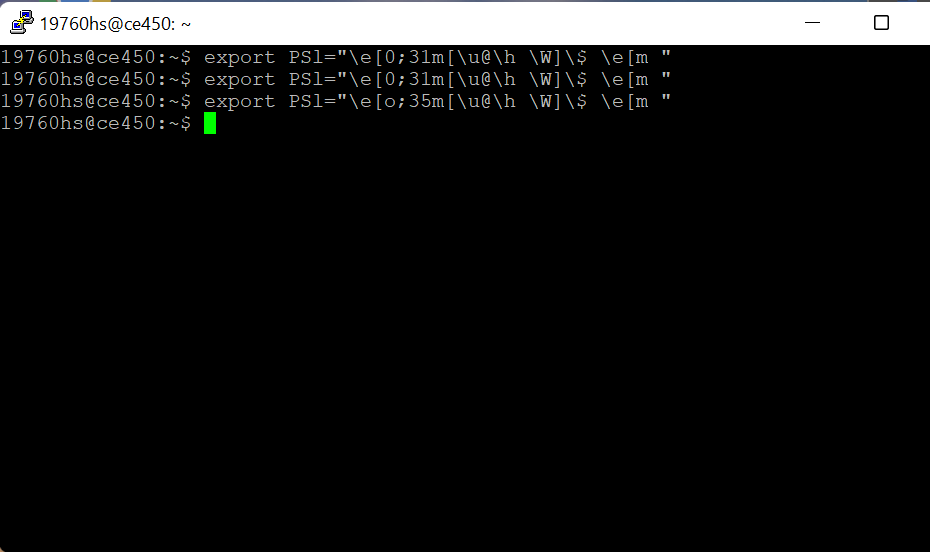
=> output – 200

=> export helps to get the value back from a shell by making it a global variables in each new shell



(6) Type in ‘export PS1="\e[0;31m[\u@\h \W]\$ \e[m " ’ and see the prompt color change.   
You can change the color code (e.g. 0;31 to 0;35 or 0;32) to change to different color.

=> show no color change in my putty

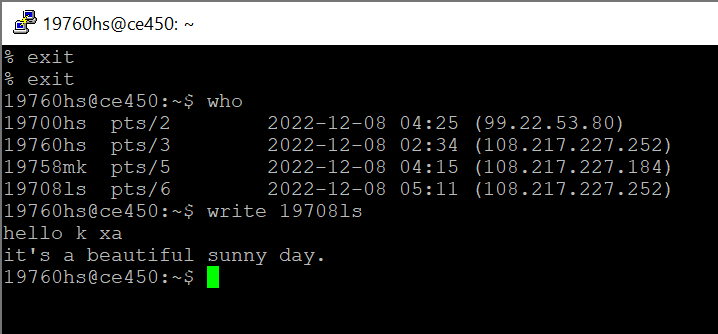


6) Write

(1) Use a command to know all of the users who log in to the same machine.  
 Write to one of your classmates who have been login to the same machine by ‘write <user name>’.

=> to know all of the user use command – who

(2) Type in the message that will show on his screen.   
 (3) You two can start conversing with and write message to each other on the screen. At the end of   
 Conversation, enter CTRL-D.



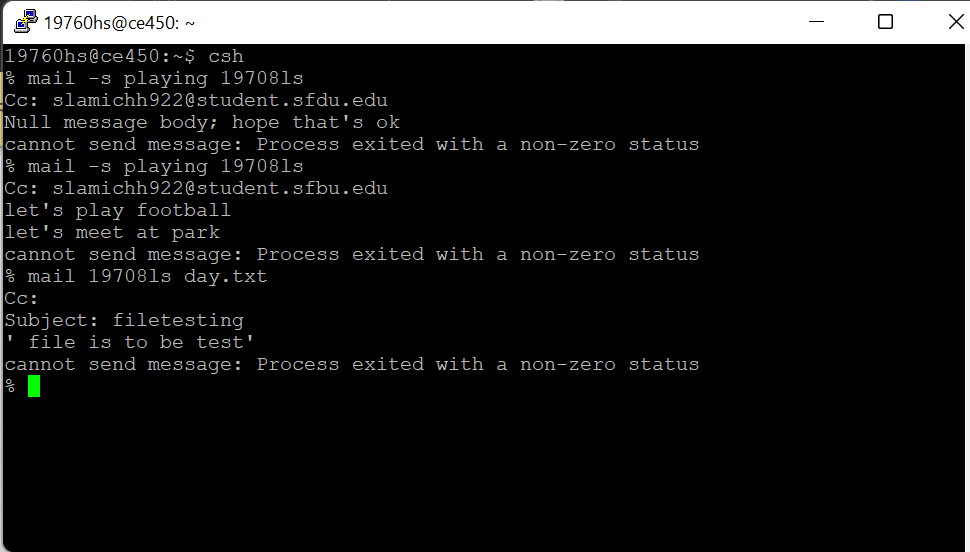
7) mail

(1) Send email to your classmate by ‘mail –s <subject> <user name>’ if he is login to the same   
 machine. Then type in the message. Enter CTR-D to send.

* => not working

Send to different machine by ‘mail m-s <subject> <email address>.  
 (2) send a text file by ‘mail <user name> < <text file>’.

=>not working



8) alias and unalias  
 (1) Create an alias ll to do the same thing of ‘ls –la’ by “alias ll=’ls –la’”.

=> typed in alias ll= ls -a

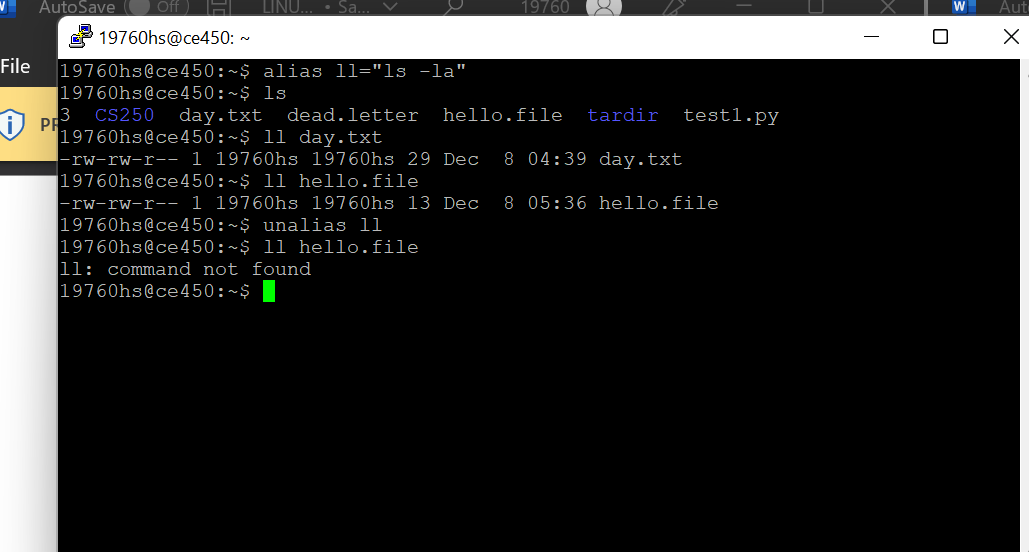
=> output – command not found

(2) Now type in ll. What is the output?

=> typed in ll – listed all the files and directory

(3) Enter ‘unalias ll’ then enter ‘ll’ what is the output?

=> command not found



9) tty

(1) Run ‘tty’ command, and note the device name of your terminal.

=> /dev/pts/3

(2) run command ‘echo hello > /dev/pts/<you should put the number you got   
 from part (1).   
 This command display hello onto your screen.  
   
 (3) Run the following two commands:  
 echo "I said hello" > hello.file  
 cp hello.file /dev/tty  
   
 What is the output?  
 What do these commands do?

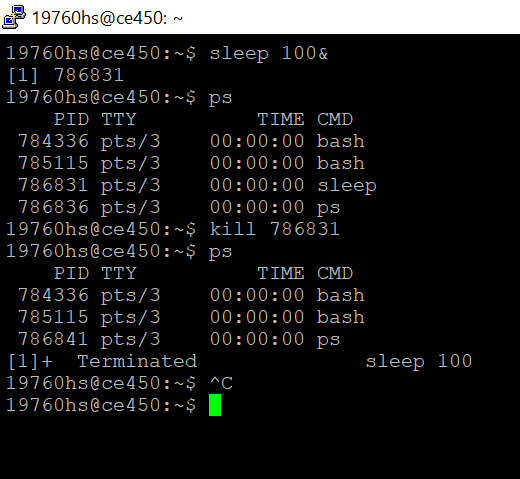
=> first command creates a file named hello.file and the second command copies the content of the hello.file.



10) ps

Each process is a running program.  
   
(1) Run the command ‘sleep 100&’ to put it into background to you can have prompt back.  
(2) Enter ‘ps’ to know its process ID. What is the PID?

(3) Kill it by ‘kill <the PID you found in step (2).

(4) Enter ‘ps’ to see if it has been terminated. 

11) tar

(1) Create a directory named tardir. Add three files f1, f2, f3 in tardir. Then three files were added which are mentioned above.

=> file created using command – ‘mkdir tadir’

=> files added using ‘touch’ command

(2) Now archive tardir by ‘tar –cvf tardir.tar tardir’. Write down the size of tardir.tar.

(3) Now archive and compress tardir by ‘tar –cvzf tardircompress.tar.gz tardir’. Write down the size   
 of this compressed tarball. Is compressed tarball smaller?  
 (3) Remove every files in tardir.  
 => removed file using command ‘rm’

(5) Uncompress and Extract the tar file by ‘tar –xzvf tardircompress.tar.gz –C tardir’. List the   
contents of tardir. Have all files been unzipped and extracted back?

