**LAB 0**

**Basic Exercises**

1. Once the virtual machine has finished booting, use the command **pwd** to print the current (or **p**resent) **w**orking **d**irectory.

**Note:** Using the command *pwd*, the current (present) working directory printed as **/home/jely2159**.

1. How many files does the *home directory* contain?  A simple way to find out is to use the **ls** command.

**Answer: none**

1. How many *hidden* files does the *home directory* contain?  With no arguments, the **ls** command doesn’t show hidden files.  Look at the man page for **ls** by running the command ‘**man ls**’.  You can navigate in the man page by using the up and down keys.  
     
   HINT: *Hidden* files in Unix/Linux have names starting with “.”.  For example, “**.bash\_history**” is a *hidden* file.

**Answer: four** (specifically: .bash\_history, .bash\_logout, .bash\_profile, .bashrc)

1. In what directory would you expect to find the **cp** command?

**Answer: /bin** (I found this by typing which cp)

1. Where is the command to make a directory (**mkdir)** located on the filesystem?  What command did you use to find **mkdir**?  Give an alternative to the command you initially used to find **mkdir**.

**Answer: /bin is where mkdir is located. I used “which mkdir” to find its location on the filesystem. An alternative I found to “which mkdir” is: “find / -name mkdir”. This command allows all directories under the root directory containing “mkdir” to be displayed. Using this command, I found that mkdir is located in /usr/bin.**

1. Use the **mkdir** command to create a new directory under the root user’s home directory (i.e. **/root/**).  Name it anything you’d like.  Use the **touch** command to create a file under that new directory.  What does the new file contain?

Note: I needed to switch from my user home directory /home/jely2159 to /root, so I used the su – command and confirmed I was in /root by using pwd. I created a new directory called newDir and used touch newDir/newFile to create a new file in newDir called newFile.

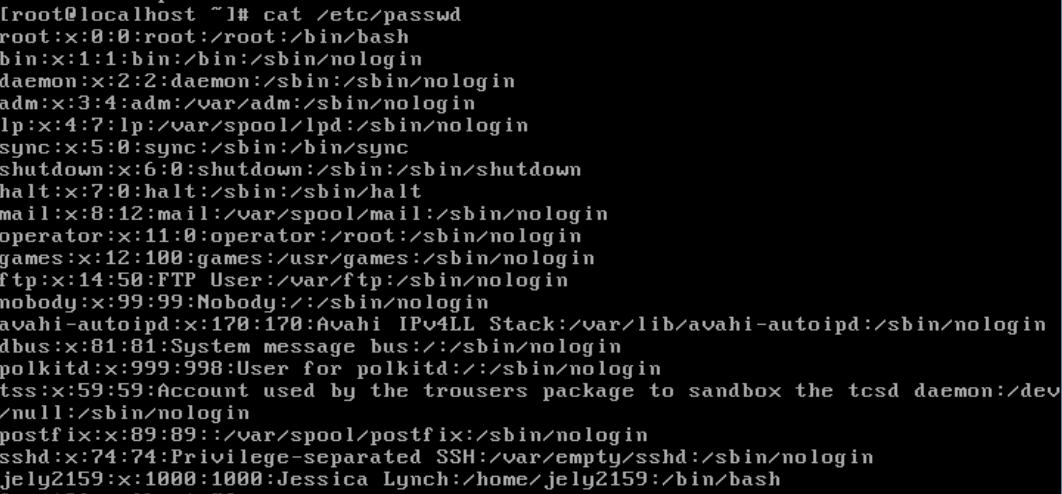
**Answer: I used both the less and cat commands to confirm that the new file newFile contains nothing and that it is indeed empty.**

1. By default, the **rm** command will not remove directories.  You can use the flag *-r* to tell the **rm** command to remove recursively; i.e., remove all files & directories under the target directory (and the target directory itself).  What happens when you run the command “**rm**” without -rf to remove the directory you created in #6?  What happens when you run the command “**rm -rf**” to remove the directory you created in #6?

**Answer: Using “rm” without –rf, resulted in the following error message: “rm: cannot remove ‘newDir’: Is a directory”. Using “rm” with –rf, removes the directory completely including of course the file I created within it.**

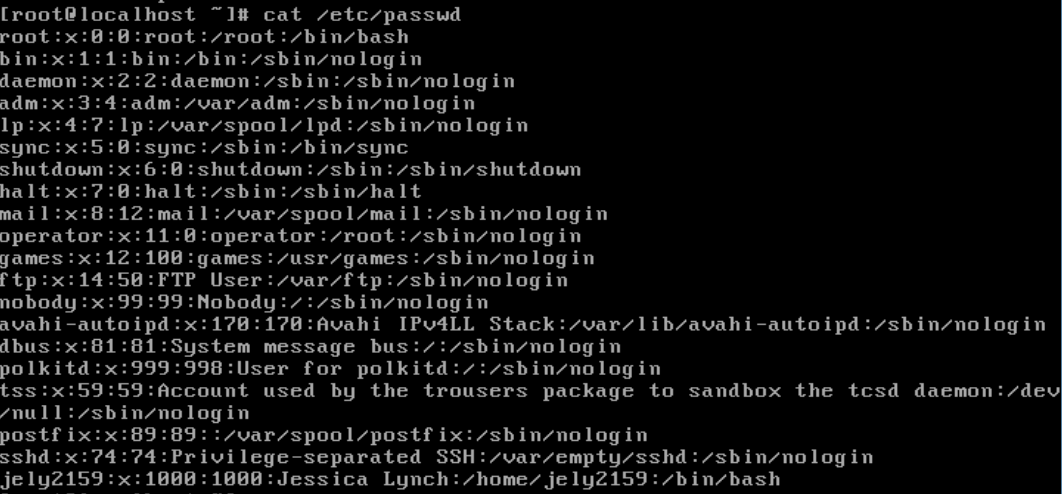
1. Print the contents of **/etc/passwd**, which contains the list of users on the system in a very specific format.  This format is:  
           username:password\_hash:user\_id\_number:group\_id\_number:full\_name:home\_directory:default\_shell

cat /etc/passwd printed out the following:



Write a *command pipeline* to print a list of just usernames here:

Note: The command pipeline I used to print the below list of just usernames is: “cat /etc/passwd | awk –F: ‘{print $1}’”. GNU awk or gawk is a pattern scanning and processing language. The –F: is a command parameter to specify the field delimiter which is a colon (:). The ‘{print $1}’ specifies to print the first field based on the field delimiter, which according to the format per line is the username. I found that an alternative to this would be: **cat /etc/passwd | cut –d: -f1**. ☺

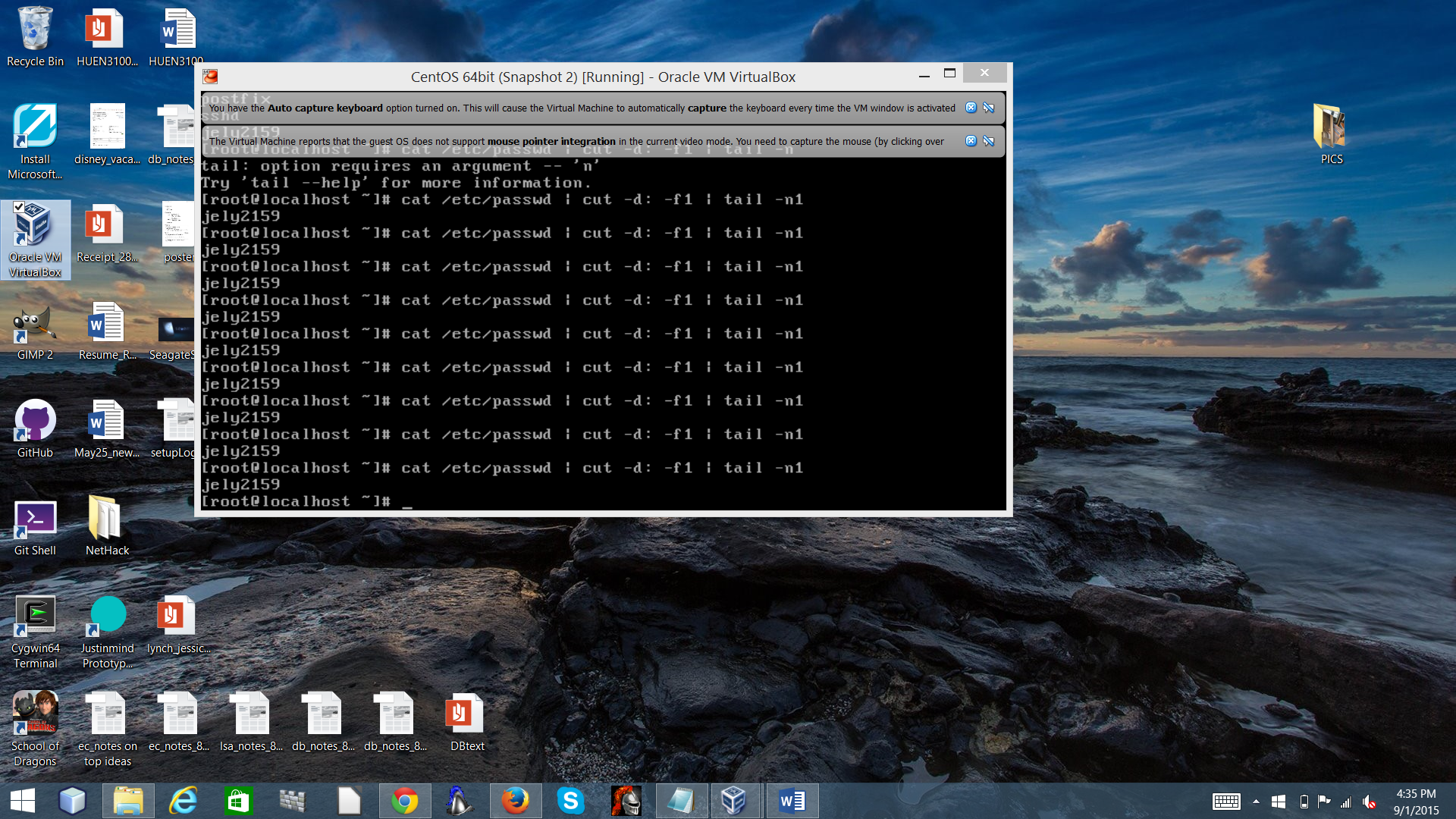


1. Write a *command* *pipeline* of the **cat**, **cut**, and **tail** commands to print only the username of the last user in **/etc/passwd** here:

The command pipeline combinng cat, cut and tail that I used is:

**cat /etc/passwd | cut –d: -f1 | tail –n1**

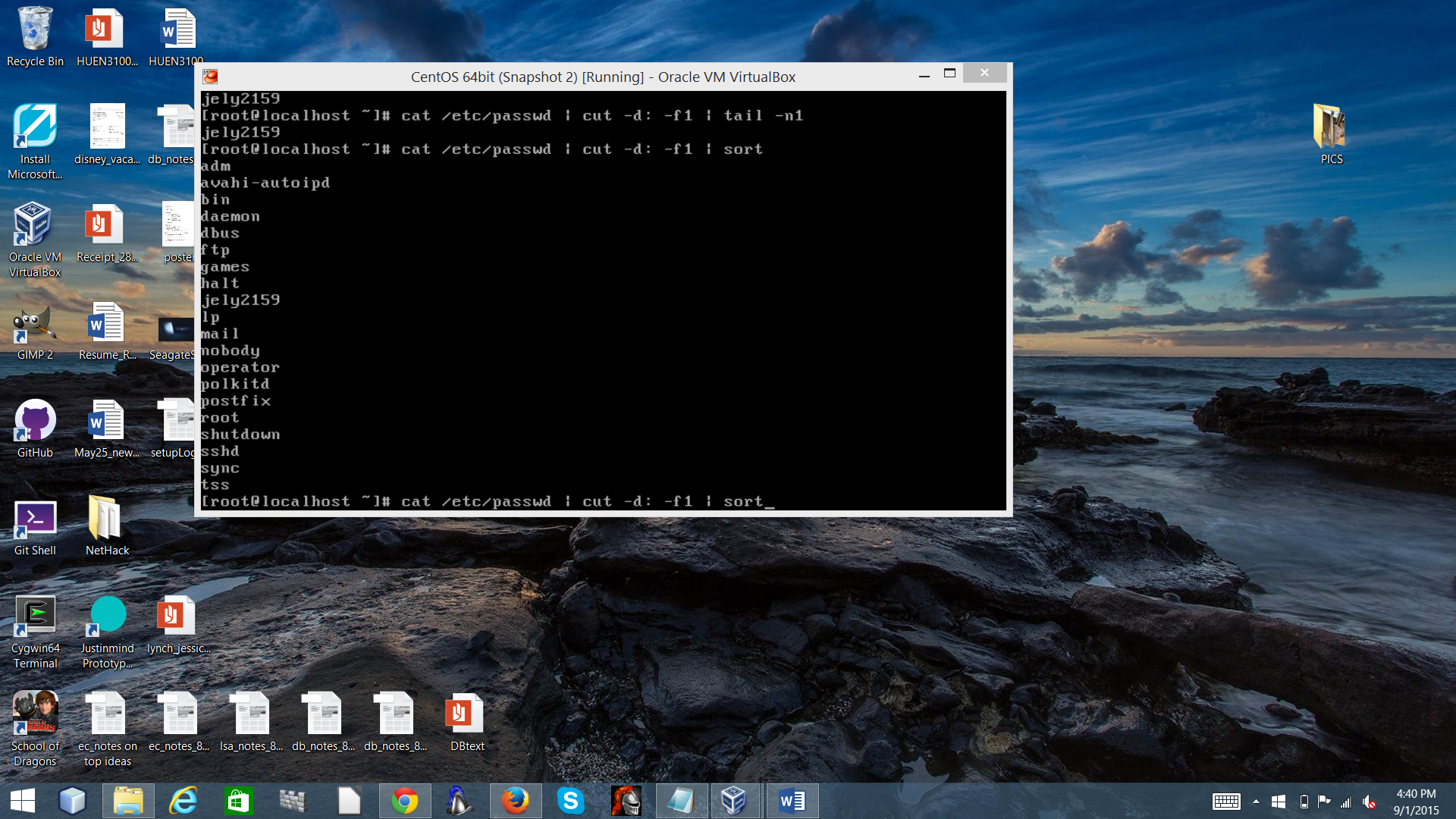
Output:



1. Combine the **cat**, **cut**, and **sort** commands to print only the usernames, sorted alphabetically, in descending order.  Write the *command pipeline* here:

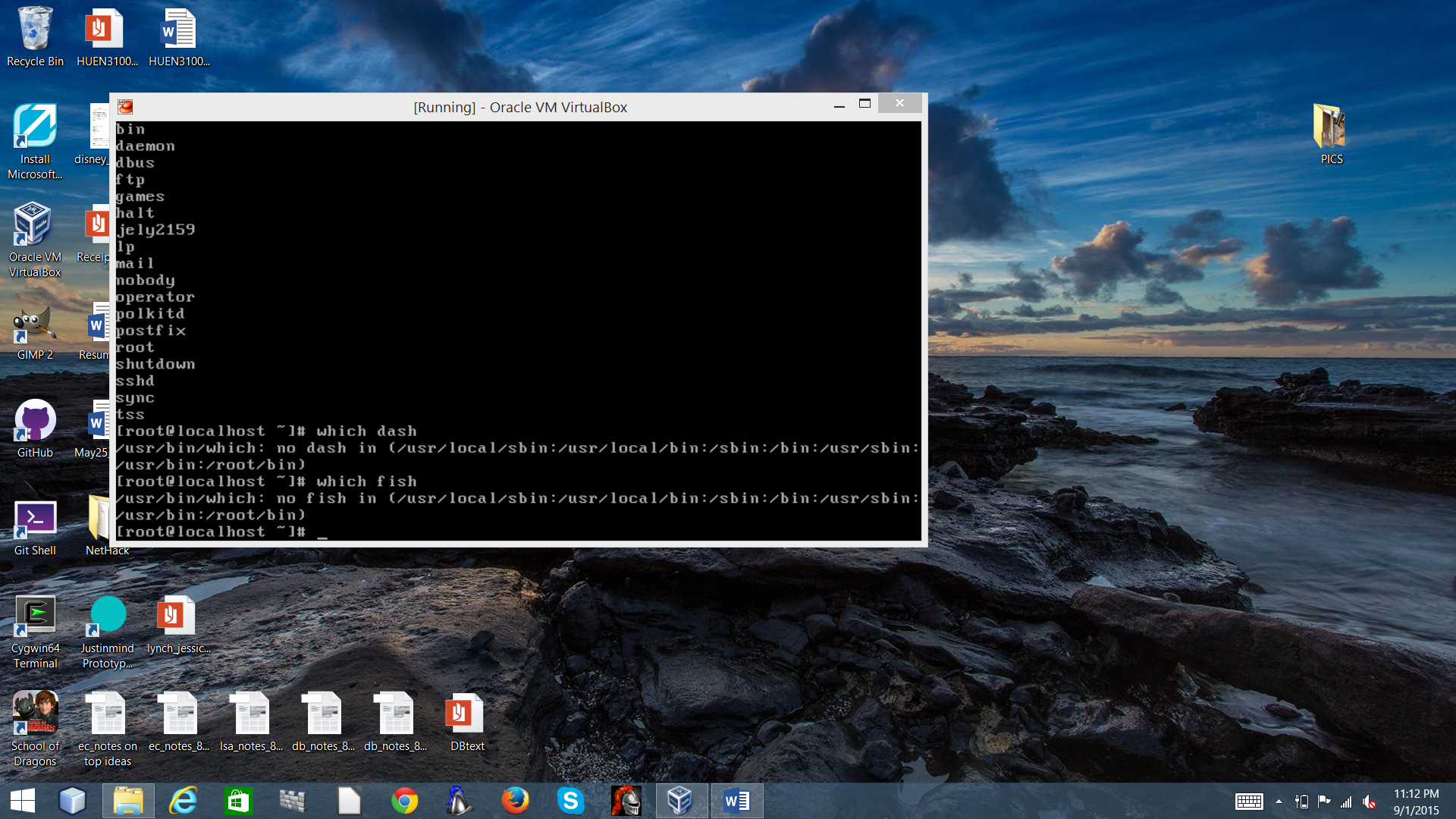
**cat /etc/passwd | cut –d: -f1 | sort**

output:



1. Is the Debian Almquist Shell (**dash**) available on this virtual machine?  Is the Fish shell (**fish**) available?  List two ways below to check the availability of a shell.

Running “which dash” and also “which fish” gives me the following output:



1. What is the current value of the $PATH environment variable?  How would you append the directory **/usr/local/bin**?

echo PATH

1. Issue this command and explain the result. “> time; date >> time; cat < time**”**
2. Take a [snapshot of the virtual machine](http://www.google.com/url?q=http%3A%2F%2Fwww.howtogeek.com%2F150258%2Fhow-to-save-time-by-using-snapshots-in-virtualbox%2F&sa=D&sntz=1&usg=AFQjCNFGOe5OD8egOjmsBTcxqtRH9BV-cA), then run the command “**rm / -rf**” on your virtual machine.  What happened?  Restart the virtual machine (you may have to click Machine, then Reset).  Does it boot?