CIS-449 Intro to Software Security

With Professor Dr. Anys Bacha

Assignment 1

Student: Demetrius Johnson

02 February 2023

Due: 08 February 2023 at 9am

Abstract

In this lab, we explore the use of various Linux commands when running a Bash shell; to do this, we use a virtual machine that is running Linux Ubuntu 20.04, using VirtualBox Hypervisor to run and manage the machine. After, we log into the machine over SSH and SFTP in order to transfer files and interface with the operating system of the virtual machine without needing to use the GUI of the virtual machine, which is much slower. Some of the primary commands we run to learn Bash syntax are **grep, cp, ls, mv, rm, touch,** and **cd.** We also work with flags/options of the functions to gain more control from the power of the functions, as well as some of the syntax of regular expressions such as ***, [],** or **^.** Lastly, we used some of the bash operators such as pipe **|,** and redirect **>** in order to redirect output of functions/commands to another function or file, respectively. Over the course of this lab, you will learn to navigate the Linux system, and how to connect to virtual machines using other interface methods.

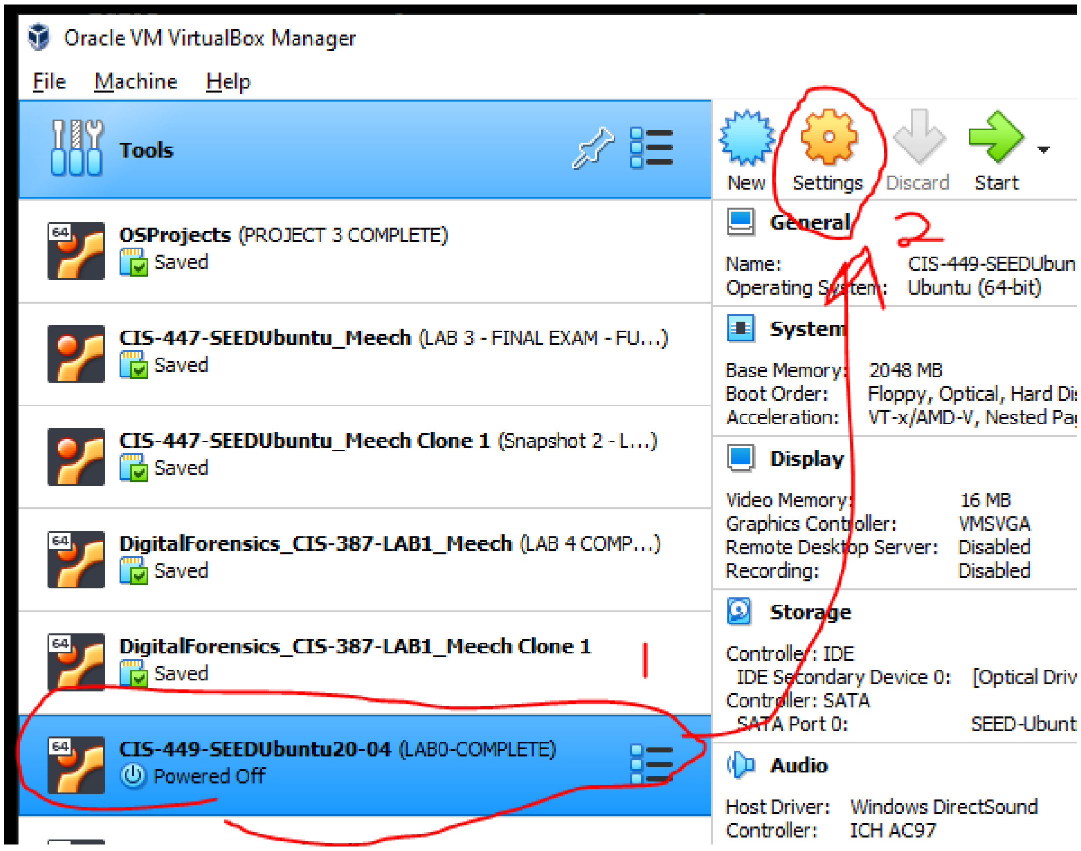
TASK 1: Enabling and connecting to virtual machine over SSH and SFTP

SSH via PuTTY

Background

I will write some notes here of my solution, and why the solution works, as a I discovered/researched according to my knowledge about computer networks:

1. FIRST OF ALL MAKE SURE THE VIRTUAL MACHINE IS POWERED OFF BEFORE TOGGLING THE NECESSARY SETTINGS!
   a) Select your machine to configure
   b) Then click "settings", as shown below:



OSProjects (PROJECT 3 COMPLETE) Saved
CIS-447-SEEDUbuntu_Meech (LAB 3 - FINAL EXAM - FU...) Saved
CIS-447-SEEDUbuntu_Meech Clone 1 (Snapshot 2 - L...) Saved
DigitalForensics_CIS-387-LAB1_Meech (LAB 4 COMP...) Saved
DigitalForensics_CIS-387-LAB1_Meech Clone 1 Saved
CIS-449-SEEDUbuntu20-04 (LAB0-COMPLETE) Powered Off

General
Name: CIS-449-SEEDUbun
Operating System: Ubuntu (64-bit)

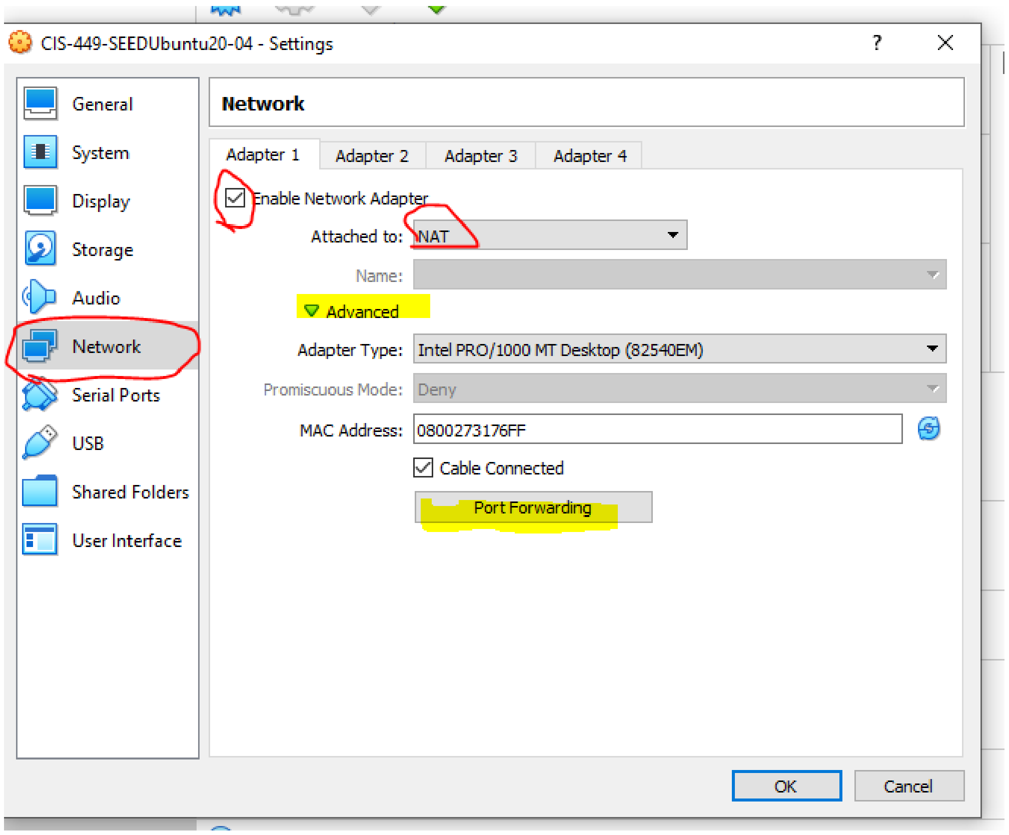
System
Base Memory: 2048 MB
Boot Order: Floppy, Optical, Hard Di
Acceleration: VT-x/AMD-V, Nested Pa

Display
Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage
Controller: IDE
IDE Secondary Device 0: [Optical Driv
Controller: SATA
SATA Port 0: SEED-Ubunt

Audio
Host Driver: Windows DirectSound
Controller: ICH AC97

2. Now go to the "Network" tab and make sure the following settings are configured as shown below (circled in red):
    a) Then, click "Advanced", then "Port Forwarding" as highlighted below:



CIS-449-SEEDUbuntu20-04 - Settings

Network

☑ Enable Network Adapter

Attached to: NAT

Name:

Advanced

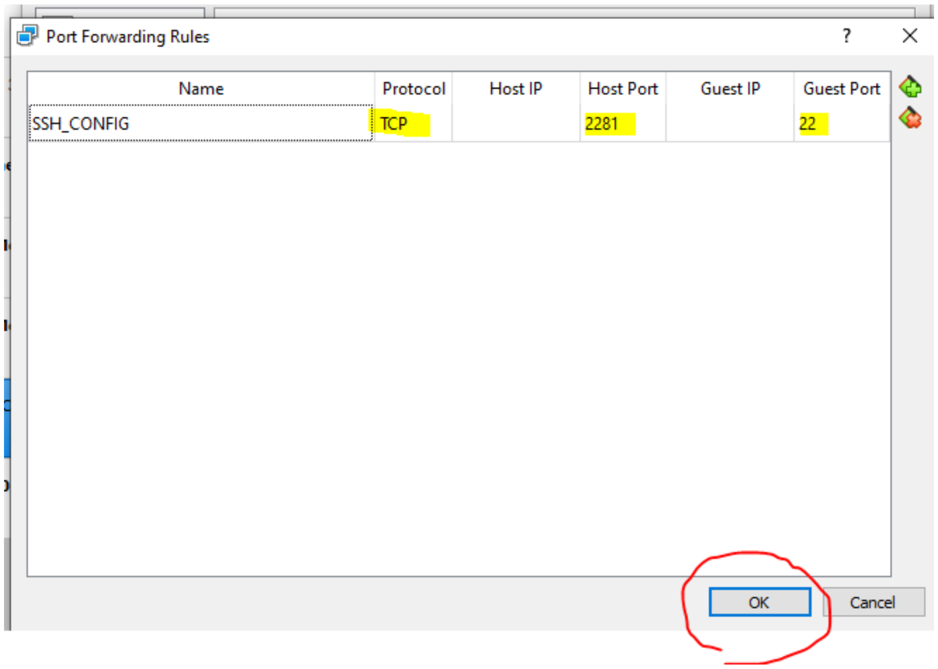
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Deny

MAC Address: 0800273176FF

☑ Cable Connected

3. Now, add a name (optional), make the protocol TCP (this can also work with UDP, but TCP will make connection reliable), set **Host Port to a value such as 2281**, and the **Guest Port to 22** (default ssh port number = 22, but it does not have to be the port used).



Name	Protocol	Host IP	Host Port	Guest IP	Guest Port
SSH_CONFIG	TCP		2281		22

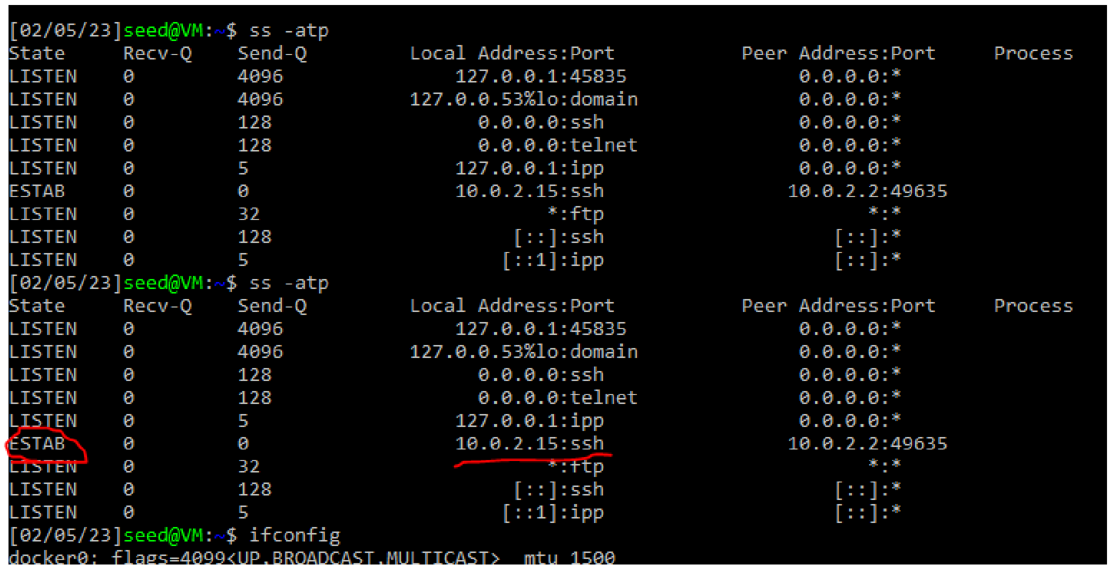
a) We leave the Host and Guest IP blank, and in doing so all IP traffic on the Host machine that is tagged to port 2281 will be forwarded to a default IP address and port number of the virtual machine → VirtualBox configuration sets IP address of virtual machines to 10.0.x.15, where x depends on which interface id + 2 the machine will use for network connectivity.

b) Now, according to [How to configure port forwarding in VirtualBox for NAT Networking | LaptrinhX](), Virtual Box uses NAT Networking to configure the networking/network adapter of the virtual machine.

c) The virtual machine is a on a separate private network than the local host machine.

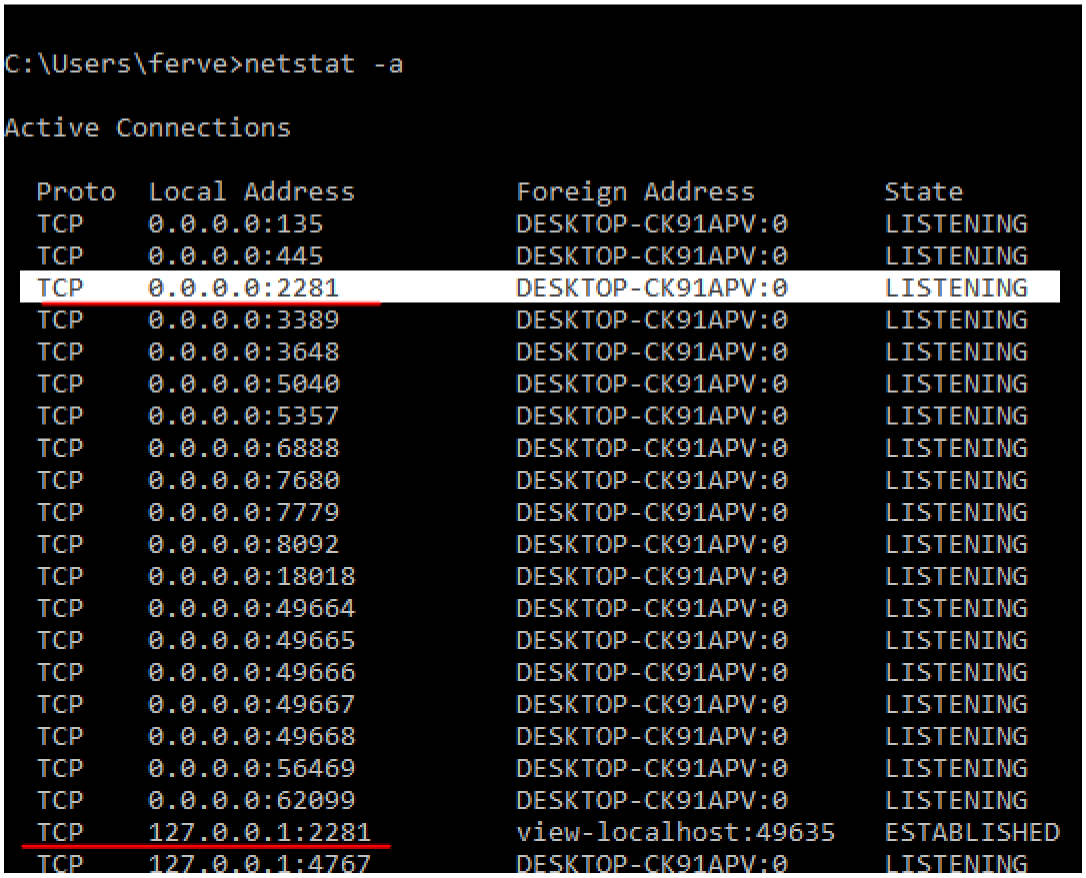
d) So, we must request to connect to the loopback address of the host machine (127.0.0.1) on port 2281. Even though every machine that runs IP has a loopback address of 127.0.0.1, and the host and virtual machine are technically the same hardware, VirtualBox keeps loopback traffic of the local host separate from the virtual machine loopback traffic.

e) VirtualBox will then map 127.0.0.1:2281 of local host traffic to be 10.0.x.15:23 and forward it to the virtual machine which is on the 10.0.2.1 network (thus acting as a router running NAT = network address translation). The default gateway of the virtual machine is 10.0.2.2, and thus when I ran a ss -atp to check TCP traffic on the virtual machine, you will see a connection is established on the virtual machine on local address 10.0.2.15, to peer address 10.0.2.2.



```
[02/05/23]seed@VM:~$ ss -atp
State     Recv-Q    Send-Q         Local Address:Port          Peer Address:Port     Process
LISTEN    0         4096               127.0.0.1:45835              0.0.0.0:*
LISTEN    0         4096           127.0.0.53%lo:domain             0.0.0.0:*
LISTEN    0         128                  0.0.0.0:ssh                0.0.0.0:*
LISTEN    0         128                  0.0.0.0:telnet             0.0.0.0:*
LISTEN    0         5                  127.0.0.1:ipp                0.0.0.0:*
ESTAB     0         0                  10.0.2.15:ssh               10.0.2.2:49635
LISTEN    0         32                         *:ftp                      *:*
LISTEN    0         128                     [::]:ssh                   [::]:*
LISTEN    0         5                      [::1]:ipp                   [::]:*
[02/05/23]seed@VM:~$ ss -atp
State     Recv-Q    Send-Q         Local Address:Port          Peer Address:Port     Process
LISTEN    0         4096               127.0.0.1:45835              0.0.0.0:*
LISTEN    0         4096           127.0.0.53%lo:domain             0.0.0.0:*
LISTEN    0         128                  0.0.0.0:ssh                0.0.0.0:*
LISTEN    0         128                  0.0.0.0:telnet             0.0.0.0:*
LISTEN    0         5                  127.0.0.1:ipp                0.0.0.0:*
ESTAB     0         0                  10.0.2.15:ssh               10.0.2.2:49635
LISTEN    0         32                         *:ftp                      *:*
LISTEN    0         128                     [::]:ssh                   [::]:*
LISTEN    0         5                      [::1]:ipp                   [::]:*
[02/05/23]seed@VM:~$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
```

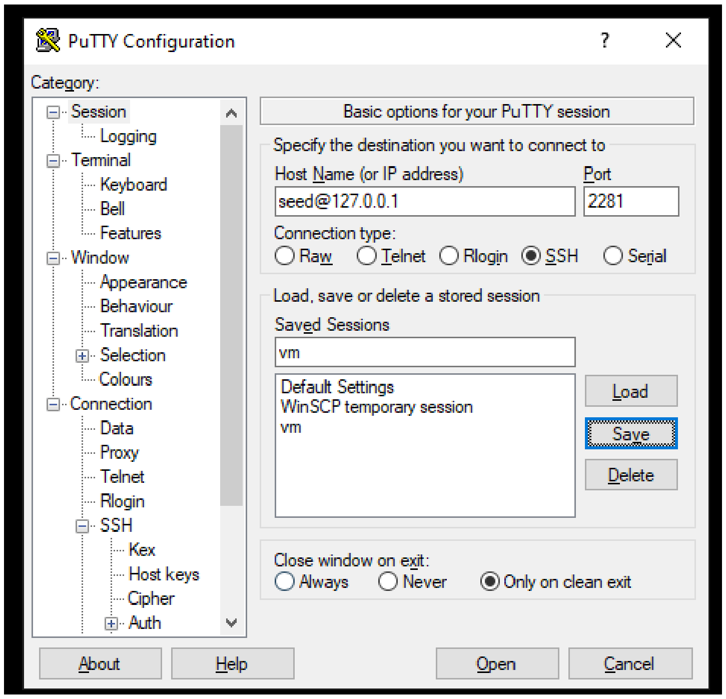
f) And when I ran netstat command on my local Windows machine, notice port 2281 on local machine (denoted by 0.0.0.0, and 127.0.0.1).



```
C:\Users\ferve>netstat -a

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            DESKTOP-CK91APV:0      LISTENING
  TCP    0.0.0.0:445            DESKTOP-CK91APV:0      LISTENING
  TCP    0.0.0.0:2281           DESKTOP-CK91APV:0      LISTENING
  TCP    0.0.0.0:3389           DESKTOP-CK91APV:0      LISTENING
  TCP    0.0.0.0:3648           DESKTOP-CK91APV:0      LISTENING
  TCP    0.0.0.0:5040           DESKTOP-CK91APV:0      LISTENING
  TCP    0.0.0.0:5357           DESKTOP-CK91APV:0      LISTENING
  TCP    0.0.0.0:6888           DESKTOP-CK91APV:0      LISTENING
  TCP    0.0.0.0:7680           DESKTOP-CK91APV:0      LISTENING
  TCP    0.0.0.0:7779           DESKTOP-CK91APV:0      LISTENING
  TCP    0.0.0.0:8092           DESKTOP-CK91APV:0      LISTENING
  TCP    0.0.0.0:18018          DESKTOP-CK91APV:0      LISTENING
  TCP    0.0.0.0:49664          DESKTOP-CK91APV:0      LISTENING
  TCP    0.0.0.0:49665          DESKTOP-CK91APV:0      LISTENING
  TCP    0.0.0.0:49666          DESKTOP-CK91APV:0      LISTENING
  TCP    0.0.0.0:49667          DESKTOP-CK91APV:0      LISTENING
  TCP    0.0.0.0:49668          DESKTOP-CK91APV:0      LISTENING
  TCP    0.0.0.0:56469          DESKTOP-CK91APV:0      LISTENING
  TCP    0.0.0.0:62099          DESKTOP-CK91APV:0      LISTENING
  TCP    127.0.0.1:2281         view-localhost:49635   ESTABLISHED
  TCP    127.0.0.1:4767         DESKTOP-CK91APV:0      LISTENING
```

How to connect via PuTTY

1. Open PuTTY
2. Enter username@127.0.0.1; username is seed thus do seed@127.0.0.1
3. Set port number to be as Host Port that you configured in previous step; in this case it is **2281** as following with my example.
4. Thus you are connecting the Host machine to itself over SSH on port 2281, which virtualbox will see that IP traffic and forward it to the virtual machine on port 22 (and also changing the destination IP to 10.0.x.15).



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): seed@127.0.0.1
Port: 2281

Connection type: Raw Telnet Rlogin [✓ SSH] Serial

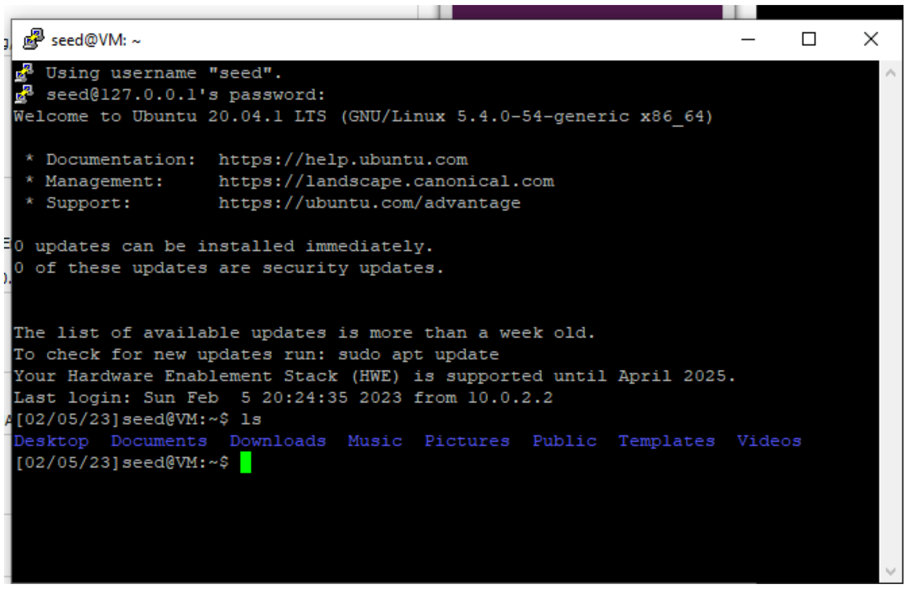
Load, save or delete a stored session

Saved Sessions: vm

Default Settings
WinSCP temporary session
vm

Close window on exit: Always Never [✓ Only on clean exit]

5. Now you just type in the password and you are connected:



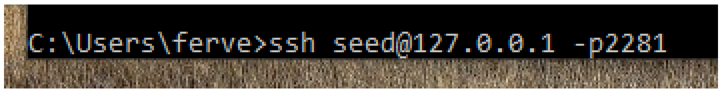
seed@VM: ~

```
Using username "seed".
seed@127.0.0.1's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 updates can be installed immediately.
0 of these updates are security updates.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sun Feb  5 20:24:35 2023 from 10.0.2.2
[02/05/23]seed@VM:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
[02/05/23]seed@VM:~$
```

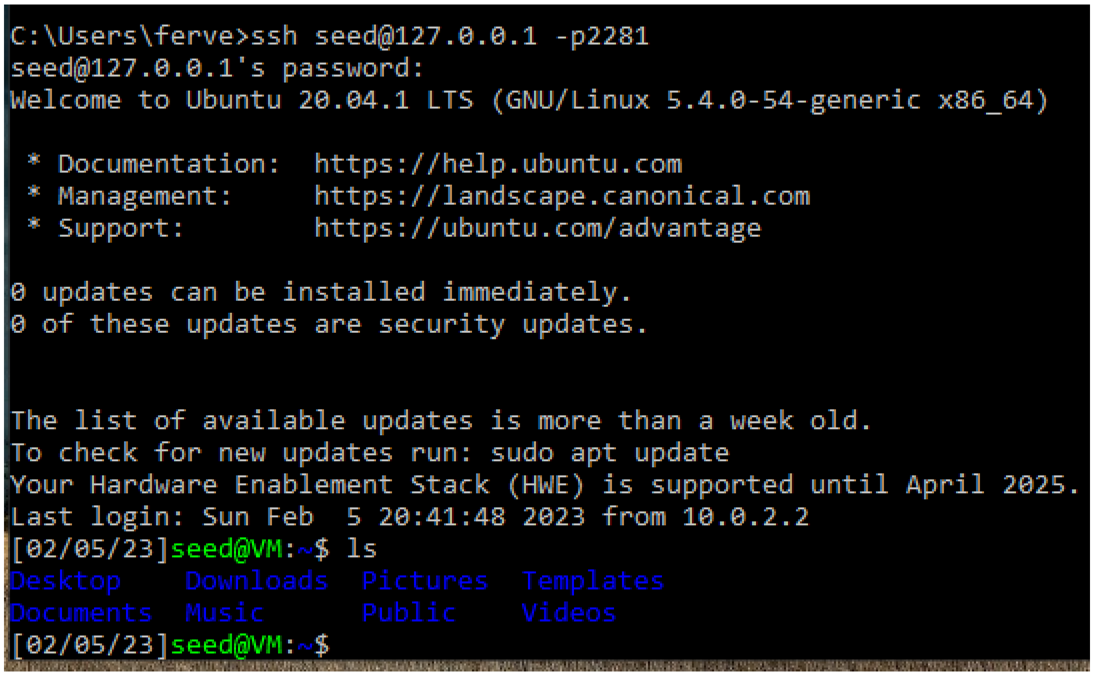
How to connect via Windows CMD

(ssh(1) - OpenBSD manual pages)

-p *port*	Port to connect to on the remote host. This can be specified on a per-host basis in the configuration file.



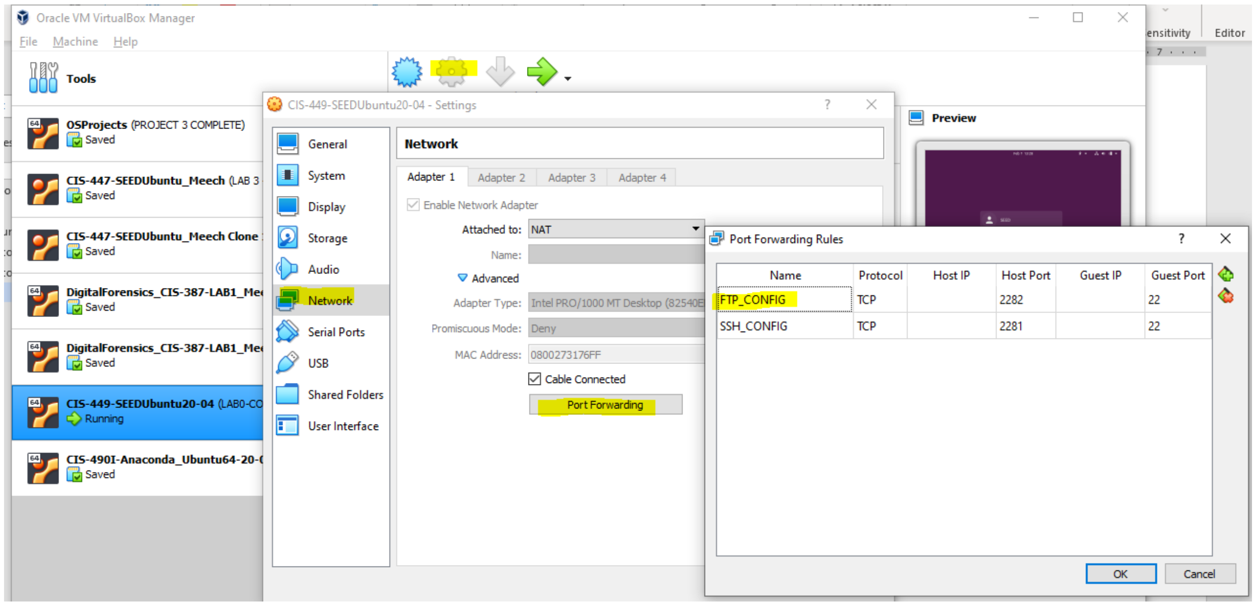
```
C:\Users\ferve>ssh seed@127.0.0.1 -p2281
```



```
C:\Users\ferve>ssh seed@127.0.0.1 -p2281
seed@127.0.0.1's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 updates can be installed immediately.
0 of these updates are security updates.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sun Feb  5 20:41:48 2023 from 10.0.2.2
[02/05/23]seed@VM:~$ ls
Desktop    Downloads  Pictures  Templates
Documents  Music      Public    Videos
[02/05/23]seed@VM:~$
```

SFTP via winSCP

Do the same steps as for SSH:



Oracle VM VirtualBox Manager
File Machine Help
Tools
OSProjects (PROJECT 3 COMPLETE) Saved
CIS-447-SEEDUbuntu_Meech (LAB 3 Saved
CIS-447-SEEDUbuntu_Meech Clone Saved
DigitalForensics_CIS-387-LAB1_Me Saved
DigitalForensics_CIS-387-LAB1_Me Saved
CIS-449-SEEDUbuntu20-04 (LAB0-CO Running
CIS-490I-Anaconda_Ubuntu64-20- Saved
CIS-449-SEEDUbuntu20-04 - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
Preview

Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

Enable Network Adapter

Attached to: NAT

Name:

Advanced

Adapter Type: Intel PRO/1000 MT Desktop (82540E

Promiscuous Mode: Deny

MAC Address: 0800273176FF

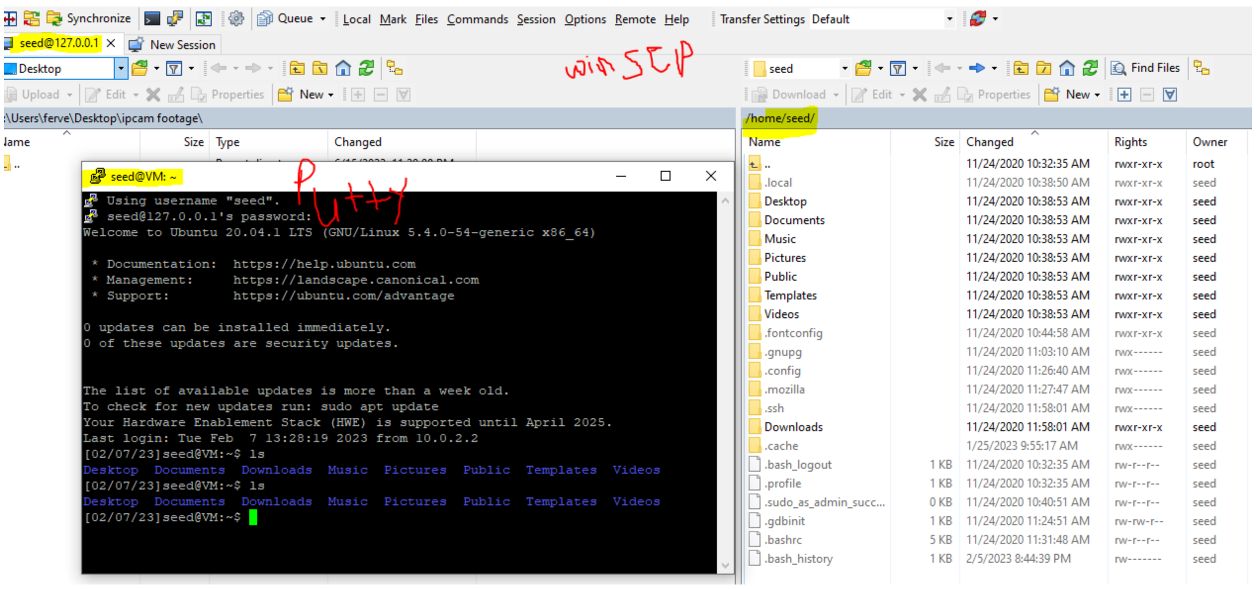
Cable Connected

Port Forwarding

Port Forwarding Rules

Name	Protocol	Host IP	Host Port	Guest IP	Guest Port
FTP_CONFIG	TCP		2282		22
SSH_CONFIG	TCP		2281		22

OK Cancel



Synchronize
Queue
Local Mark Files Commands Session Options Remote Help
Transfer Settings Default
seed@127.0.0.1
New Session
Desktop
Upload Edit Properties New
win SCP
seed
Find Files
Download Edit Properties New
Putty

:\Users\ferve\Desktop\ipcam footage\

/home/seed/

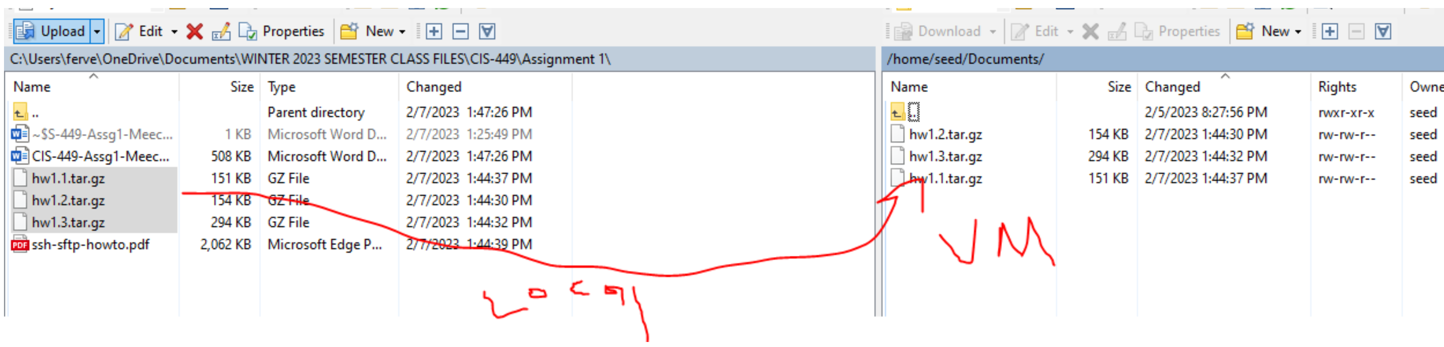
seed@VM: ~

```
Using username "seed".
seed@127.0.0.1's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 updates can be installed immediately.
0 of these updates are security updates.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Tue Feb  7 13:28:19 2023 from 10.0.2.2
[02/07/23]seed@VM:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
[02/07/23]seed@VM:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
[02/07/23]seed@VM:~$
```

Name	Size	Changed	Rights	Owner
..		11/24/2020 10:32:35 AM	rwxr-xr-x	root
.local		11/24/2020 10:38:50 AM	rwxr-xr-x	seed
Desktop		11/24/2020 10:38:53 AM	rwxr-xr-x	seed
Documents		11/24/2020 10:38:53 AM	rwxr-xr-x	seed
Music		11/24/2020 10:38:53 AM	rwxr-xr-x	seed
Pictures		11/24/2020 10:38:53 AM	rwxr-xr-x	seed
Public		11/24/2020 10:38:53 AM	rwxr-xr-x	seed
Templates		11/24/2020 10:38:53 AM	rwxr-xr-x	seed
Videos		11/24/2020 10:38:53 AM	rwxr-xr-x	seed
.fontconfig		11/24/2020 10:44:58 AM	rwxr-xr-x	seed
.gnupg		11/24/2020 11:03:10 AM	rwx------	seed
.config		11/24/2020 11:26:40 AM	rwx------	seed
.mozilla		11/24/2020 11:27:47 AM	rwx------	seed
.ssh		11/24/2020 11:58:01 AM	rwx------	seed
Downloads		11/24/2020 11:58:01 AM	rwxr-xr-x	seed
.cache		1/25/2023 9:55:17 AM	rwx------	seed
.bash_logout	1 KB	11/24/2020 10:32:35 AM	rw-r--r--	seed
.profile	1 KB	11/24/2020 10:32:35 AM	rw-r--r--	seed
.sudo_as_admin_succ...	0 KB	11/24/2020 10:40:51 AM	rw-r--r--	seed
.gdbinit	1 KB	11/24/2020 11:24:51 AM	rw-rw-r--	seed
.bashrc	5 KB	11/24/2020 11:31:48 AM	rw-r--r--	seed
.bash_history	1 KB	2/5/2023 8:44:39 PM	rw-------	seed

TASK 2: Prepare your home directory

I used WinSCP file transfer via SFTP client to transfer the tar ball files from my local machine to the VM, inside the home/seed/Documents directory:



Upload | Edit | Properties | New

C:\Users\ferve\OneDrive\Documents\WINTER 2023 SEMESTER CLASS FILES\CIS-449\Assignment 1\

Name	Size	Type	Changed
..		Parent directory	2/7/2023 1:47:26 PM
~$S-449-Assg1-Meec...	1 KB	Microsoft Word D...	2/7/2023 1:25:49 PM
CIS-449-Assg1-Meec...	508 KB	Microsoft Word D...	2/7/2023 1:47:26 PM
hw1.1.tar.gz	151 KB	GZ File	2/7/2023 1:44:37 PM
hw1.2.tar.gz	154 KB	GZ File	2/7/2023 1:44:30 PM
hw1.3.tar.gz	294 KB	GZ File	2/7/2023 1:44:32 PM
ssh-sftp-howto.pdf	2,062 KB	Microsoft Edge P...	2/7/2023 1:44:39 PM

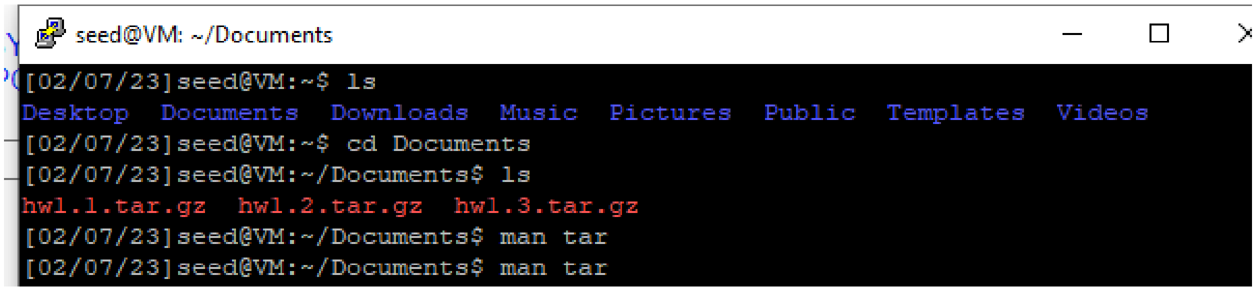
Download | Edit | Properties | New

/home/seed/Documents/

Name	Size	Changed	Rights	Owne
..		2/5/2023 8:27:56 PM	rwxr-xr-x	seed
hw1.2.tar.gz	154 KB	2/7/2023 1:44:30 PM	rw-rw-r--	seed
hw1.3.tar.gz	294 KB	2/7/2023 1:44:32 PM	rw-rw-r--	seed
hw1.1.tar.gz	151 KB	2/7/2023 1:44:37 PM	rw-rw-r--	seed

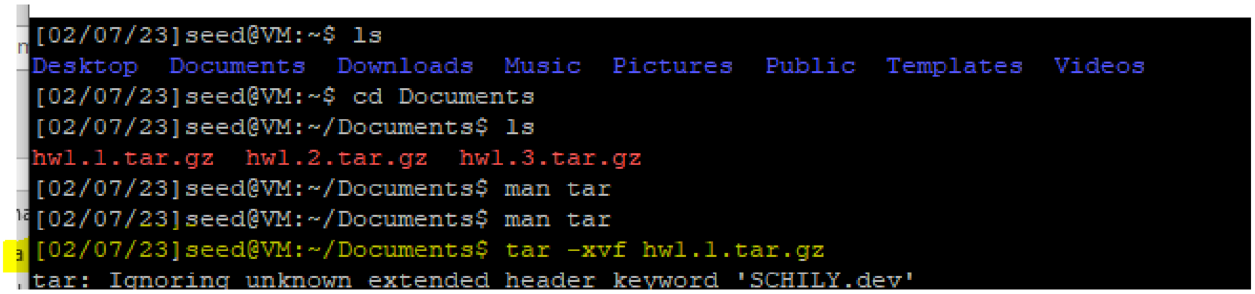
Now I navigate to the home/seed/Documents directory of the VM in PuTTY so that I can run "tar -xvf" command to decompress the tar ball files:

-Going to the Documents Directory and displaying using ls:



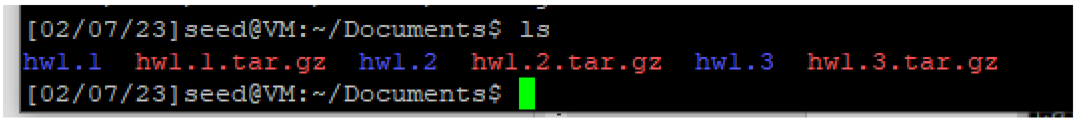
```
seed@VM: ~/Documents
[02/07/23]seed@VM:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
[02/07/23]seed@VM:~$ cd Documents
[02/07/23]seed@VM:~/Documents$ ls
hwl.1.tar.gz  hwl.2.tar.gz  hwl.3.tar.gz
[02/07/23]seed@VM:~/Documents$ man tar
[02/07/23]seed@VM:~/Documents$ man tar
```

-Now here is the output after running tar -xvf <tar file> →-x means extract, -v means verbose (show messages during extraction), and -f means archive file expected:



```
[02/07/23]seed@VM:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
[02/07/23]seed@VM:~$ cd Documents
[02/07/23]seed@VM:~/Documents$ ls
hwl.1.tar.gz  hwl.2.tar.gz  hwl.3.tar.gz
[02/07/23]seed@VM:~/Documents$ man tar
[02/07/23]seed@VM:~/Documents$ man tar
[02/07/23]seed@VM:~/Documents$ tar -xvf hwl.1.tar.gz
tar: Ignoring unknown extended header keyword 'SCHILY.dev'
```

-Now showing I have decompressed all tar ball files and they are in the Documents Directory:

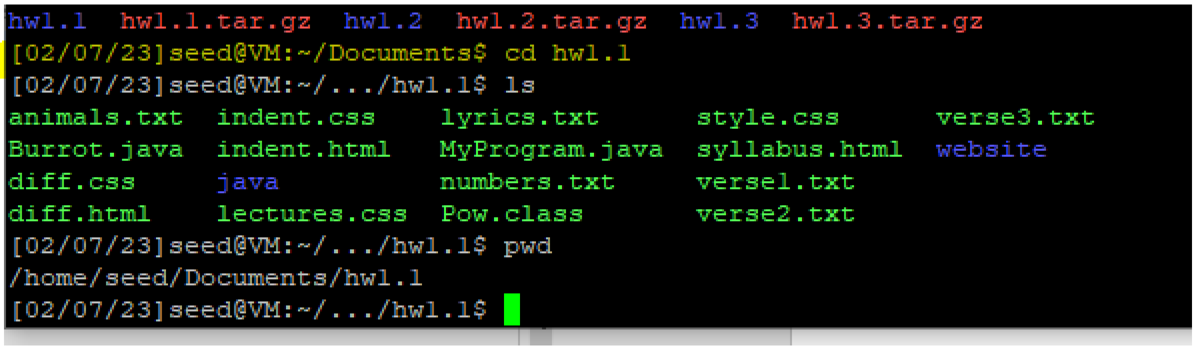


```
[02/07/23]seed@VM:~/Documents$ ls
hwl.1  hwl.1.tar.gz  hwl.2  hwl.2.tar.gz  hwl.3  hwl.3.tar.gz
[02/07/23]seed@VM:~/Documents$
```

TASK 3: Linux Bash Shell commands 1

For each of the numbered items below, determine a single bash shell statement that will perform the operation(s) requested. Each of your solutions must be a single one-line shell statement and should not use Linux's multi-statement joining operators such as |, &&, ||, and ; unless told otherwise. You can use the following references:

The numbered statements must be done from within the "/home/seed/Documents/hw1.1" directory. As such, you must use the "cd" command to change into the appropriate directory:

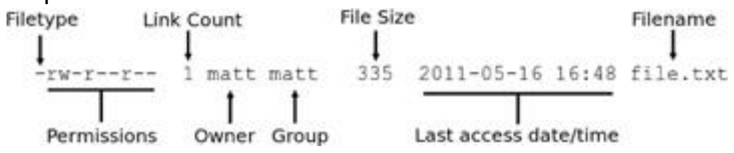


```
hw1.1  hw1.1.tar.gz  hw1.2  hw1.2.tar.gz  hw1.3  hw1.3.tar.gz
[02/07/23]seed@VM:~/Documents$ cd hw1.1
[02/07/23]seed@VM:~/.../hw1.1$ ls
animals.txt  indent.css    lyrics.txt      style.css      verse3.txt
Burrot.java  indent.html   MyProgram.java  syllabus.html  website
diff.css     java          numbers.txt     verse1.txt
diff.html    lectures.css  Pow.class       verse2.txt
[02/07/23]seed@VM:~/.../hw1.1$ pwd
/home/seed/Documents/hw1.1
[02/07/23]seed@VM:~/.../hw1.1$
```

1. List the contents of the "hw1.1" directory using the "ls" command. You must include the full command in your report and include a screenshot capturing the first page of the output. The output must show:

A long listing of all files and directories within hw1

- -l option



-

Any hidden files/directories

- -a option

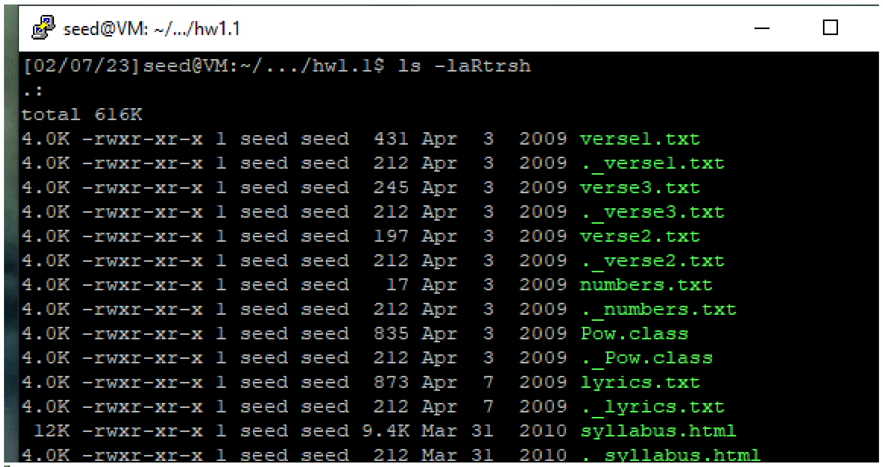
All files and directories within hw1 listed in reverse chronological order (oldest file/directory at the top and newest file/directory at the bottom).

- -R option for recursively listing all files and directories **in current directory and subdirectories**
- -t option: sort by modification time, newest first
- -r option: reverse order while sorting

Readable sizes of all files and directories

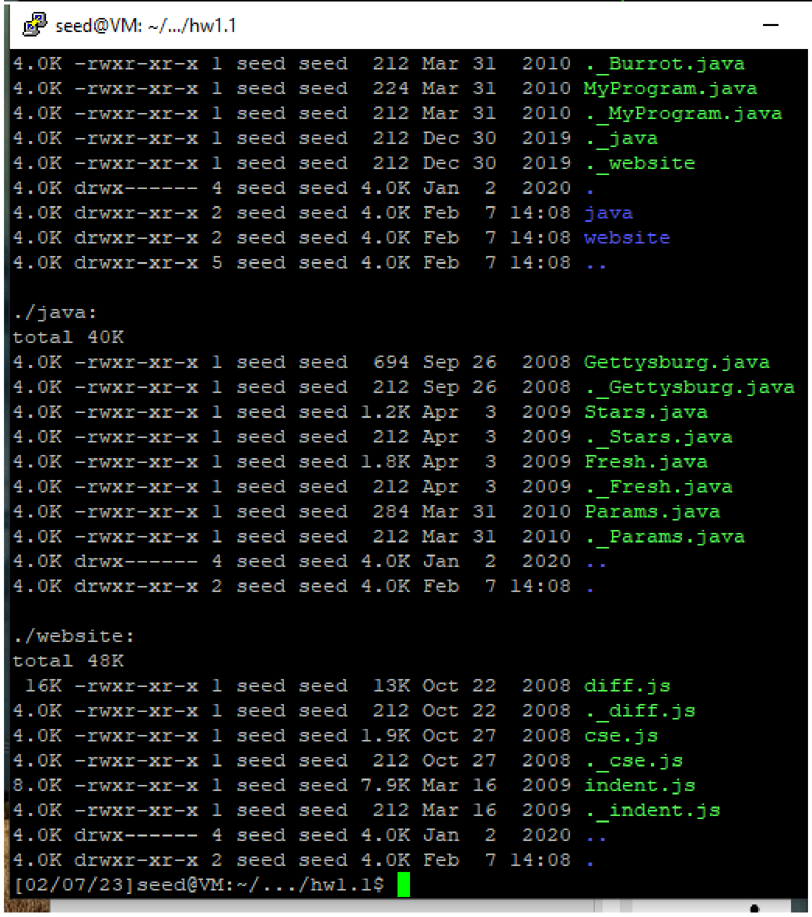
- -s (list size of files **in *blocks*, UNLESS -h option is included**)
- -h option: with -l and -s, print **sizes** like 1K 234M 2G etc.

Here is the output without the **less** function:

- 

seed@VM: ~/.../hw1.1

```
[02/07/23]seed@VM:~/.../hw1.1$ ls -laRtrsh
.:
total 616K
4.0K -rwxr-xr-x 1 seed seed  431 Apr  3  2009 verse1.txt
4.0K -rwxr-xr-x 1 seed seed  212 Apr  3  2009 ._verse1.txt
4.0K -rwxr-xr-x 1 seed seed  245 Apr  3  2009 verse3.txt
4.0K -rwxr-xr-x 1 seed seed  212 Apr  3  2009 ._verse3.txt
4.0K -rwxr-xr-x 1 seed seed  197 Apr  3  2009 verse2.txt
4.0K -rwxr-xr-x 1 seed seed  212 Apr  3  2009 ._verse2.txt
4.0K -rwxr-xr-x 1 seed seed   17 Apr  3  2009 numbers.txt
4.0K -rwxr-xr-x 1 seed seed  212 Apr  3  2009 ._numbers.txt
4.0K -rwxr-xr-x 1 seed seed  835 Apr  3  2009 Pow.class
4.0K -rwxr-xr-x 1 seed seed  212 Apr  3  2009 ._Pow.class
4.0K -rwxr-xr-x 1 seed seed  873 Apr  7  2009 lyrics.txt
4.0K -rwxr-xr-x 1 seed seed  212 Apr  7  2009 ._lyrics.txt
 12K -rwxr-xr-x 1 seed seed 9.4K Mar 31  2010 syllabus.html
4.0K -rwxr-xr-x 1 seed seed  212 Mar 31  2010 ._syllabus.html
```

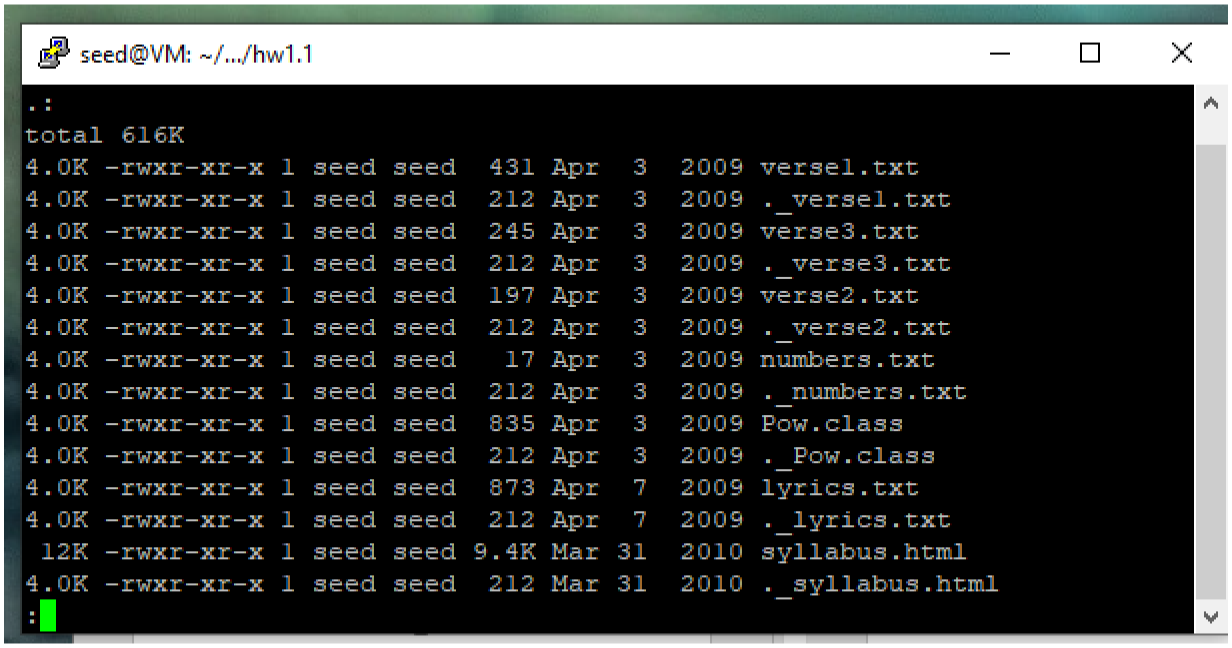
- 

seed@VM: ~/.../hw1.1

```
4.0K -rwxr-xr-x 1 seed seed  212 Mar 31  2010 ._Burrot.java
4.0K -rwxr-xr-x 1 seed seed  224 Mar 31  2010 MyProgram.java
4.0K -rwxr-xr-x 1 seed seed  212 Mar 31  2010 ._MyProgram.java
4.0K -rwxr-xr-x 1 seed seed  212 Dec 30  2019 ._java
4.0K -rwxr-xr-x 1 seed seed  212 Dec 30  2019 ._website
4.0K drwx------ 4 seed seed 4.0K Jan  2  2020 .
4.0K drwxr-xr-x 2 seed seed 4.0K Feb  7 14:08 java
4.0K drwxr-xr-x 2 seed seed 4.0K Feb  7 14:08 website
4.0K drwxr-xr-x 5 seed seed 4.0K Feb  7 14:08 ..

./java:
total 40K
4.0K -rwxr-xr-x 1 seed seed  694 Sep 26  2008 Gettysburg.java
4.0K -rwxr-xr-x 1 seed seed  212 Sep 26  2008 ._Gettysburg.java
4.0K -rwxr-xr-x 1 seed seed 1.2K Apr  3  2009 Stars.java
4.0K -rwxr-xr-x 1 seed seed  212 Apr  3  2009 ._Stars.java
4.0K -rwxr-xr-x 1 seed seed 1.8K Apr  3  2009 Fresh.java
4.0K -rwxr-xr-x 1 seed seed  212 Apr  3  2009 ._Fresh.java
4.0K -rwxr-xr-x 1 seed seed  284 Mar 31  2010 Params.java
4.0K -rwxr-xr-x 1 seed seed  212 Mar 31  2010 ._Params.java
4.0K drwx------ 4 seed seed 4.0K Jan  2  2020 ..
4.0K drwxr-xr-x 2 seed seed 4.0K Feb  7 14:08 .

./website:
total 48K
 16K -rwxr-xr-x 1 seed seed  13K Oct 22  2008 diff.js
4.0K -rwxr-xr-x 1 seed seed  212 Oct 22  2008 ._diff.js
4.0K -rwxr-xr-x 1 seed seed 1.9K Oct 27  2008 cse.js
4.0K -rwxr-xr-x 1 seed seed  212 Oct 27  2008 ._cse.js
8.0K -rwxr-xr-x 1 seed seed 7.9K Mar 16  2009 indent.js
4.0K -rwxr-xr-x 1 seed seed  212 Mar 16  2009 ._indent.js
4.0K drwx------ 4 seed seed 4.0K Jan  2  2020 ..
4.0K drwxr-xr-x 2 seed seed 4.0K Feb  7 14:08 .
[02/07/23]seed@VM:~/.../hw1.1$
```

The output must be piped to “less” to show the first page.

- Thus we need to do the following command and pipe it to **less** function:
  - ls -laRtrsh | less
  - Here is the output when we pipe to **less:**


seed@VM: ~/.../hw1.1

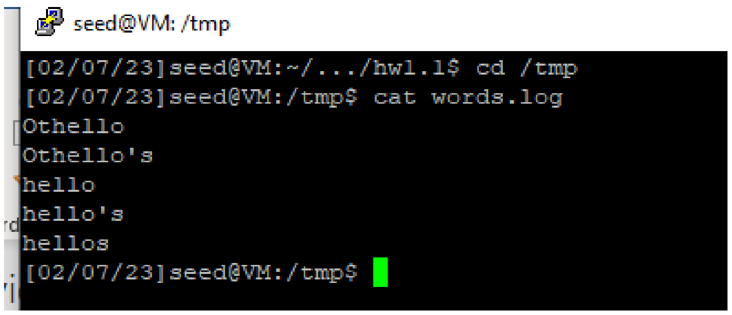
```
.:
total 616K
4.0K -rwxr-xr-x 1 seed seed  431 Apr  3  2009 verse1.txt
4.0K -rwxr-xr-x 1 seed seed  212 Apr  3  2009 ._verse1.txt
4.0K -rwxr-xr-x 1 seed seed  245 Apr  3  2009 verse3.txt
4.0K -rwxr-xr-x 1 seed seed  212 Apr  3  2009 ._verse3.txt
4.0K -rwxr-xr-x 1 seed seed  197 Apr  3  2009 verse2.txt
4.0K -rwxr-xr-x 1 seed seed  212 Apr  3  2009 ._verse2.txt
4.0K -rwxr-xr-x 1 seed seed   17 Apr  3  2009 numbers.txt
4.0K -rwxr-xr-x 1 seed seed  212 Apr  3  2009 ._numbers.txt
4.0K -rwxr-xr-x 1 seed seed  835 Apr  3  2009 Pow.class
4.0K -rwxr-xr-x 1 seed seed  212 Apr  3  2009 ._Pow.class
4.0K -rwxr-xr-x 1 seed seed  873 Apr  7  2009 lyrics.txt
4.0K -rwxr-xr-x 1 seed seed  212 Apr  7  2009 ._lyrics.txt
 12K -rwxr-xr-x 1 seed seed 9.4K Mar 31  2010 syllabus.html
4.0K -rwxr-xr-x 1 seed seed  212 Mar 31  2010 ._syllabus.html
:
```

  -
  - Note that page is relative to window size; **less** function will go to next page, up and down arrows moves to next line; press ‘q’ to quit (as I learned when using the function).

2. Run the following command "cat /usr/share/dict/words | grep -i hello > /tmp/words.log"

Write a paragraph explaining what the command does:

- The **cat** command is reading from a file on the path ***/usr/share/dict/***words, where **words** is the file.
- Then, instead of the standard output, the output from **cat** is redirected (via the pipe operator **|**)**,** to the **grep** function to be used as an input.
- Then, **grep** function (global regular expression search function) with -i parameter specifies to search for a value and ignore the case (upper or lower, i.e. e == E).
- Thus, **grep -i hello** searches for string "hello" (from **words** file) and outputs matches regardless of case (i.e. not case-sensitive; HELlo == hello).
- Finally, redirect operator **>** sends the output of **grep** to a file called **words.log** from the directory ***/temp***

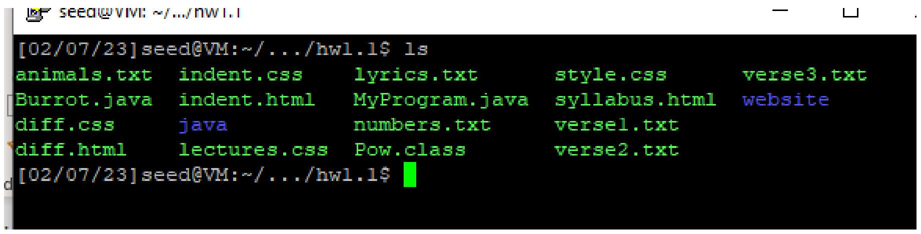

seed@VM: /tmp

```
[02/07/23]seed@VM:~/.../hw1.1$ cd /tmp
[02/07/23]seed@VM:/tmp$ cat words.log
Othello
Othello's
hello
hello's
hellos
[02/07/23]seed@VM:/tmp$ 
```

○

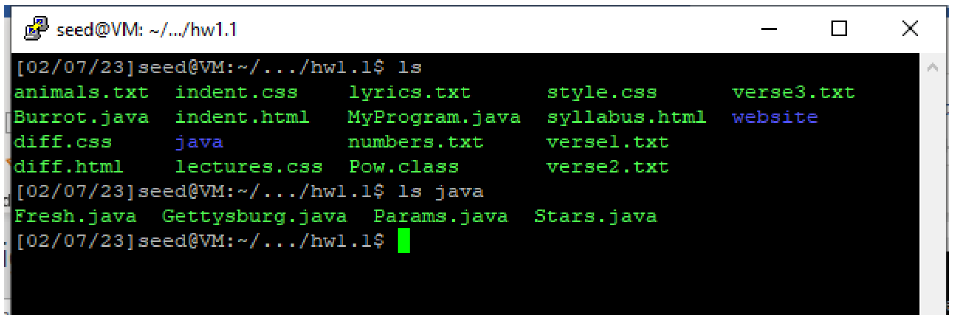
3. Copy the file numbers.txt from the current directory to the java subdirectory. Include the command in your report.

Showing contents of hw1.1 (java directory and numbers.txt directories are present)


seed@VM: ~/.../hw1.1

```
[02/07/23]seed@VM:~/.../hw1.1$ ls
animals.txt  indent.css    lyrics.txt      style.css      verse3.txt
Burrot.java  indent.html   MyProgram.java  syllabus.html  website
diff.css     java          numbers.txt     verse1.txt
diff.html    lectures.css  Pow.class       verse2.txt
[02/07/23]seed@VM:~/.../hw1.1$ 
```

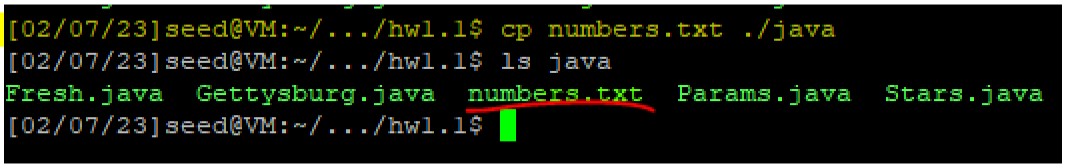
Before running command: showing contents of java directory:


seed@VM: ~/.../hw1.1

```
[02/07/23]seed@VM:~/.../hw1.1$ ls
animals.txt  indent.css    lyrics.txt      style.css      verse3.txt
Burrot.java  indent.html   MyProgram.java  syllabus.html  website
diff.css     java          numbers.txt     verse1.txt
diff.html    lectures.css  Pow.class       verse2.txt
[02/07/23]seed@VM:~/.../hw1.1$ ls java
Fresh.java  Gettysburg.java  Params.java  Stars.java
[02/07/23]seed@VM:~/.../hw1.1$ 
```

Now run this command to copy **numbers.txt** to the **java** subdirectory:

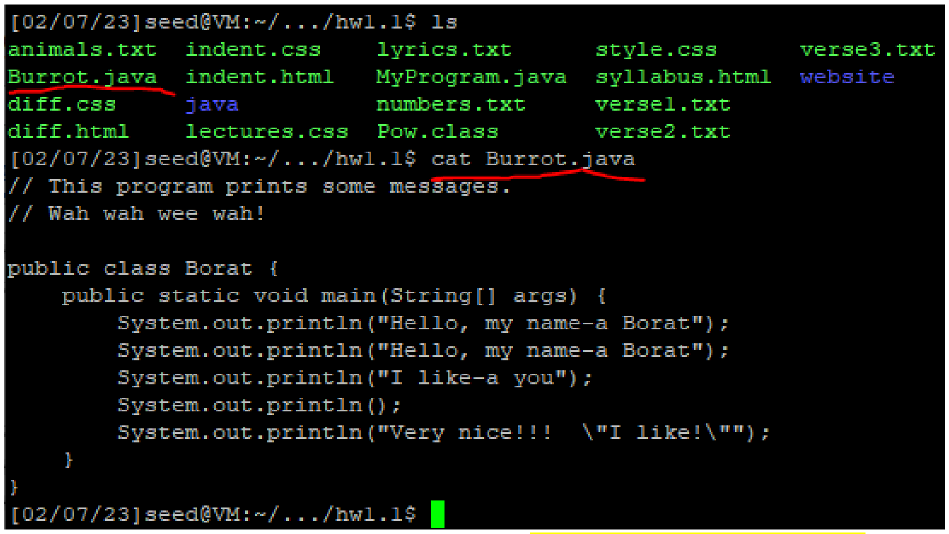
cp numbers.txt ./java



```
[02/07/23]seed@VM:~/.../hwl.1$ cp numbers.txt ./java
[02/07/23]seed@VM:~/.../hwl.1$ ls java
Fresh.java  Gettysburg.java  numbers.txt  Params.java  Stars.java
[02/07/23]seed@VM:~/.../hwl.1$
```

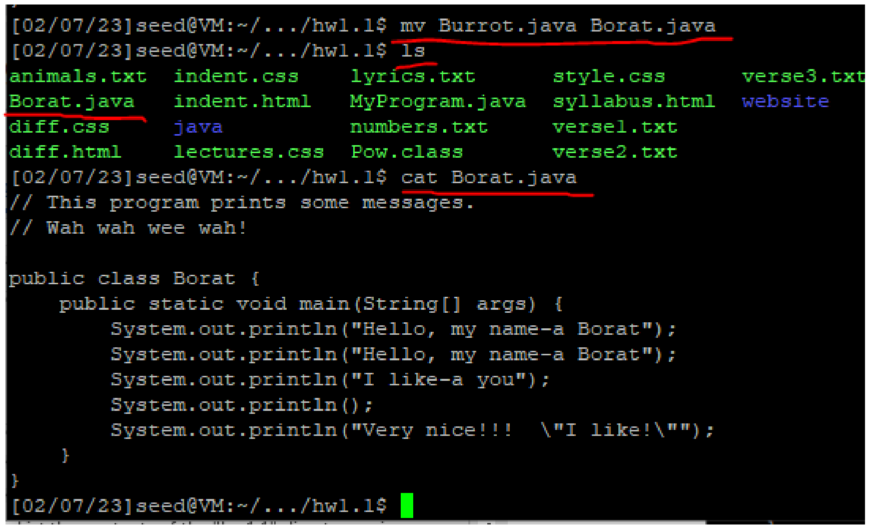
4. Rename the file Burrot.java to Borat.java (renaming is done using the same command as moving). Include the command in your report.

- Use mv command
  - mv - move (rename) files
  - Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.
- Showing output before changing Burrot.java:



```
[02/07/23]seed@VM:~/.../hwl.1$ ls
animals.txt  indent.css    lyrics.txt      style.css      verse3.txt
Burrot.java  indent.html   MyProgram.java  syllabus.html  website
diff.css     java          numbers.txt     versel.txt
diff.html    lectures.css  Pow.class       verse2.txt
[02/07/23]seed@VM:~/.../hwl.1$ cat Burrot.java
// This program prints some messages.
// Wah wah wee wah!

public class Borat {
    public static void main(String[] args) {
        System.out.println("Hello, my name-a Borat");
        System.out.println("Hello, my name-a Borat");
        System.out.println("I like-a you");
        System.out.println();
        System.out.println("Very nice!!!  \"I like!\"");
    }
}
[02/07/23]seed@VM:~/.../hwl.1$
```

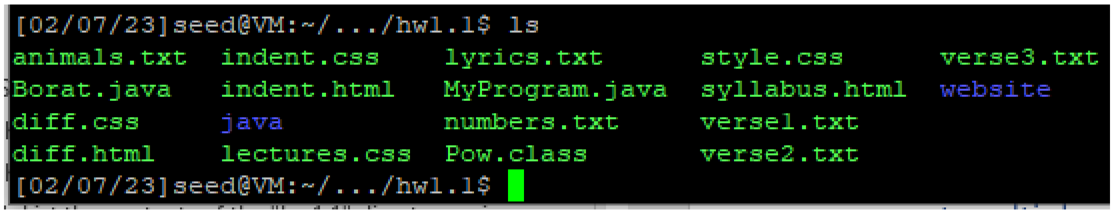
- Now run this command to change the name: **mv Burrot.java Borat.java**
- Notice file contents are the same as before, showing that the file is the same and it only has a different name:



```
[02/07/23]seed@VM:~/.../hwl.1$ mv Burrot.java Borat.java
[02/07/23]seed@VM:~/.../hwl.1$ ls
animals.txt  indent.css    lyrics.txt      style.css      verse3.txt
Borat.java   indent.html   MyProgram.java  syllabus.html  website
diff.css     java          numbers.txt     versel.txt
diff.html    lectures.css  Pow.class       verse2.txt
[02/07/23]seed@VM:~/.../hwl.1$ cat Borat.java
// This program prints some messages.
// Wah wah wee wah!

public class Borat {
    public static void main(String[] args) {
        System.out.println("Hello, my name-a Borat");
        System.out.println("Hello, my name-a Borat");
        System.out.println("I like-a you");
        System.out.println();
        System.out.println("Very nice!!!  \"I like!\"");
    }
}
[02/07/23]seed@VM:~/.../hwl.1$
```

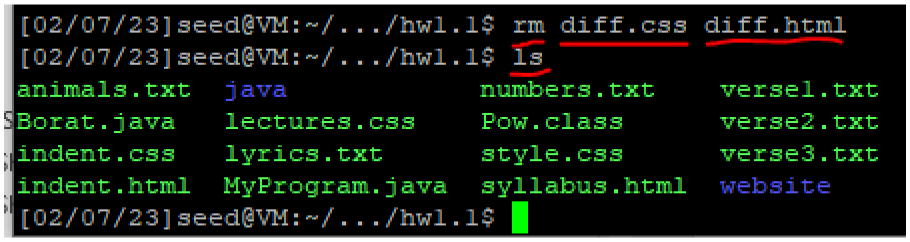
5. Delete the files diff.html and diff.css. This must be done with a single command and not multiple commands. Include the command in your report.

Before deletion of files:



```
[02/07/23]seed@VM:~/.../hw1.1$ ls
animals.txt  indent.css    lyrics.txt       style.css      verse3.txt
Borat.java   indent.html   MyProgram.java   syllabus.html  website
diff.css     java          numbers.txt      verse1.txt
diff.html    lectures.css  Pow.class        verse2.txt
[02/07/23]seed@VM:~/.../hw1.1$
```

After running command to delete files: **rm diff.css diff.html**

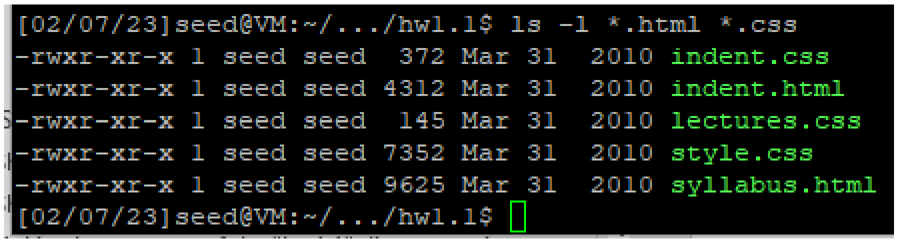


```
[02/07/23]seed@VM:~/.../hw1.1$ rm diff.css diff.html
[02/07/23]seed@VM:~/.../hw1.1$ ls
animals.txt  java            numbers.txt    verse1.txt
Borat.java   lectures.css    Pow.class      verse2.txt
indent.css   lyrics.txt      style.css      verse3.txt
indent.html  MyProgram.java  syllabus.html  website
[02/07/23]seed@VM:~/.../hw1.1$
```

6. List all web page files (files whose names end with the extension .html or .css) in the current directory. Note that the ls command can accept parameter(s) for what files you want it to list. You must include the full command in your report.

You can use a * (asterisk) as a "wild-card" character to specify a group of files. For example, *foo means all files whose names end with foo , and foo* means all files whose names begin with foo . You can use a wildcard in the middle of a file name, such as foo*bar for all files that start with foo and end with bar .

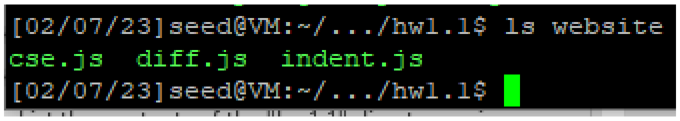
Run this command: **ls -l *.html *.css**



```
[02/07/23]seed@VM:~/.../hw1.1$ ls -l *.html *.css
-rwxr-xr-x 1 seed seed  372 Mar 31  2010 indent.css
-rwxr-xr-x 1 seed seed 4312 Mar 31  2010 indent.html
-rwxr-xr-x 1 seed seed  145 Mar 31  2010 lectures.css
-rwxr-xr-x 1 seed seed 7352 Mar 31  2010 style.css
-rwxr-xr-x 1 seed seed 9625 Mar 31  2010 syllabus.html
[02/07/23]seed@VM:~/.../hw1.1$
```

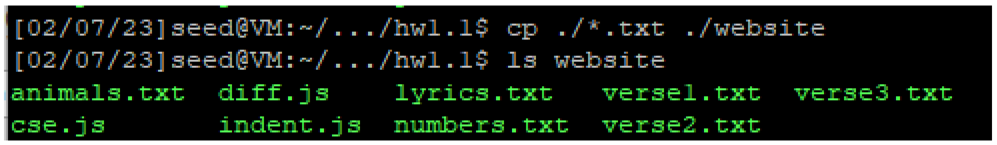
7. Copy all the text files (files whose names end with .txt) from the current folder to the website subdirectory. Include the command in your report.

The **website** directory before running the copy command:



```
[02/07/23]seed@VM:~/.../hw1.1$ ls website
cse.js  diff.js  indent.js
[02/07/23]seed@VM:~/.../hw1.1$
```

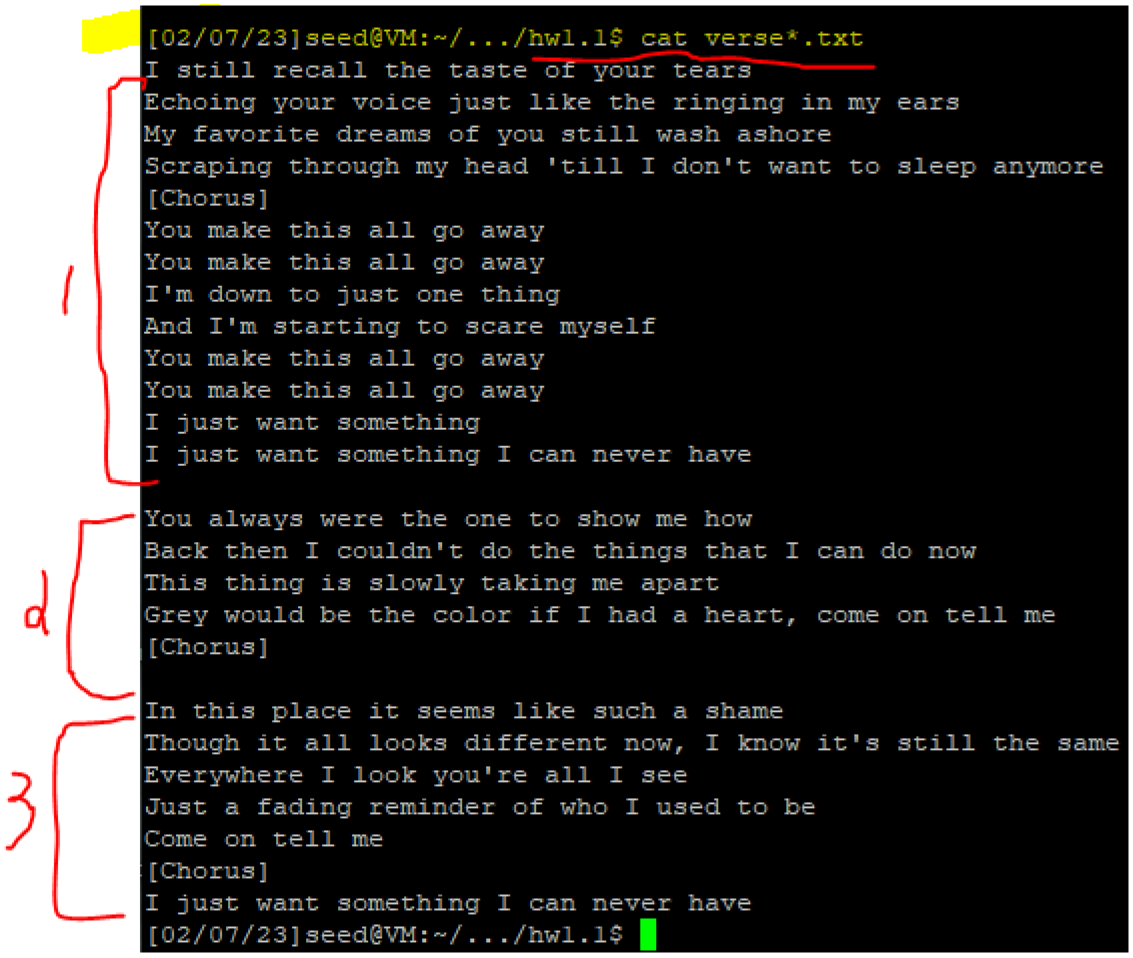
Now run **cp ./*.txt ./website** (note ./ means check current directory, *.txt means check for any instances where file ends in .txt



```
[02/07/23]seed@VM:~/.../hw1.1$ cp ./*.txt ./website
[02/07/23]seed@VM:~/.../hw1.1$ ls website
animals.txt  diff.js    lyrics.txt   verse1.txt  verse3.txt
cse.js       indent.js  numbers.txt  verse2.txt
```

8. Display the contents of all files whose names begin with verse and end with the extension .txt, such as verse1.txt and verse2.txt . (Write a single command that displays all their contents concatenated.) You must include the full command in your report.

Run the command: **cat verse*txt** (this means print to standard output the contents of any file that begin with **verse** and end with **txt**):

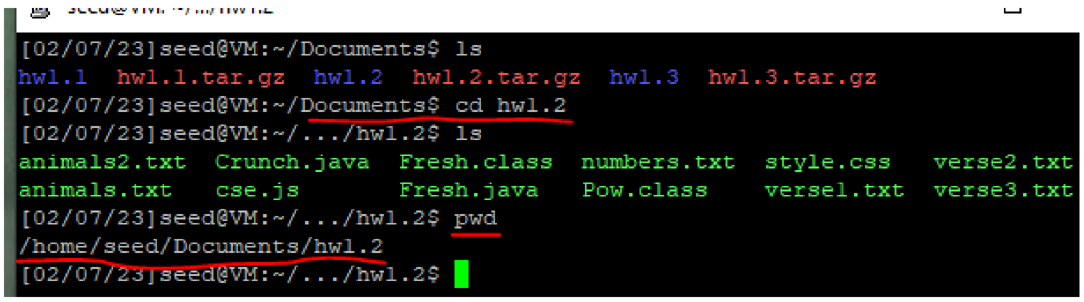


```
[02/07/23]seed@VM:~/.../hw1.1$ cat verse*.txt
I still recall the taste of your tears
Echoing your voice just like the ringing in my ears
My favorite dreams of you still wash ashore
Scraping through my head 'till I don't want to sleep anymore
[Chorus]
You make this all go away
You make this all go away
I'm down to just one thing
And I'm starting to scare myself
You make this all go away
You make this all go away
I just want something
I just want something I can never have

You always were the one to show me how
Back then I couldn't do the things that I can do now
This thing is slowly taking me apart
Grey would be the color if I had a heart, come on tell me
[Chorus]

In this place it seems like such a shame
Though it all looks different now, I know it's still the same
Everywhere I look you're all I see
Just a fading reminder of who I used to be
Come on tell me
[Chorus]
I just want something I can never have
[02/07/23]seed@VM:~/.../hw1.1$
```

TASK 4: Linux Bash Shell commands 2

For each item below, determine a single bash shell statement that will perform the operation(s) requested. Each solution must be a one-line shell statement, but you may use input/output redirection operators such as >, <, and |.

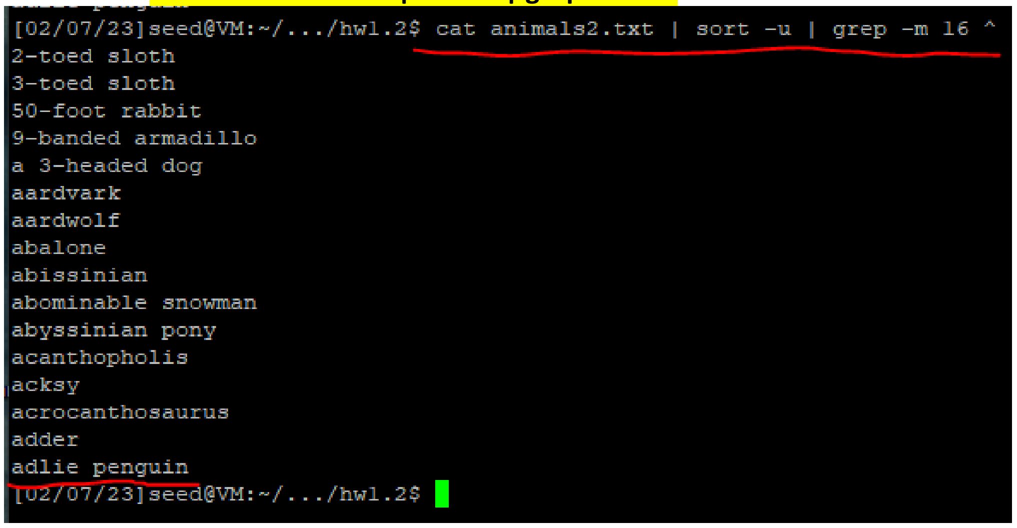
The number statements must be done from within the "/home/seed/Documents/hw1.2" directory. As such, you must use the "cd" command to change into the appropriate directory:



```
[02/07/23]seed@VM:~/Documents$ ls
hw1.1  hw1.1.tar.gz  hw1.2  hw1.2.tar.gz  hw1.3  hw1.3.tar.gz
[02/07/23]seed@VM:~/Documents$ cd hw1.2
[02/07/23]seed@VM:~/.../hw1.2$ ls
animals2.txt  Crunch.java  Fresh.class  numbers.txt  style.css   verse2.txt
animals.txt   cse.js       Fresh.java   Pow.class    verse1.txt  verse3.txt
[02/07/23]seed@VM:~/.../hw1.2$ pwd
/home/seed/Documents/hw1.2
[02/07/23]seed@VM:~/.../hw1.2$
```

1. The file animals2.txt contains an alphabetized list of animal names. It includes many duplicates. Output the first 16 distinct animals from the file, one per line. (The last one should be adlie penguin .) You must include the full command in your report and include a screenshot capturing the output.

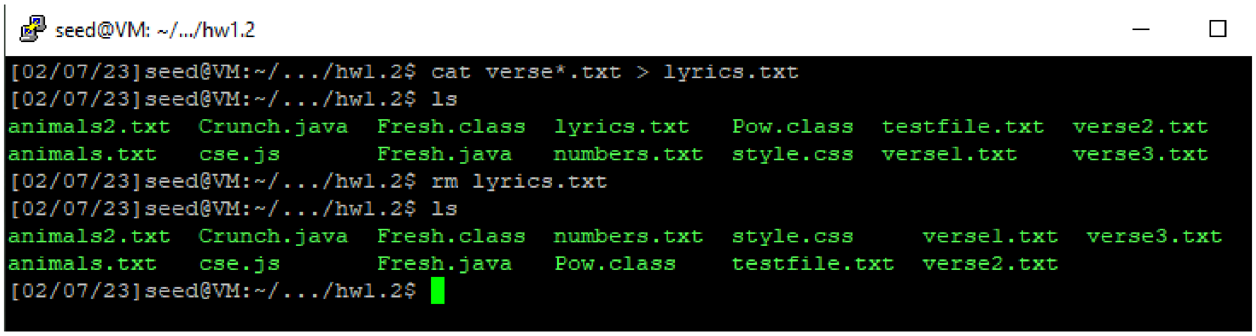
- First, we need to get all of the contents of the animals2.txt file using **cat** command
- Then, we need to redirect the output from **cat** to **sort** function
  - Then we need to use options for **sort:**
    - -u option: deletes all duplicates of sorted lines
- Then, we redirect **sort** output to **grep** and use options:
  - -m <number> option: only output the first m matches
  - ^ denotes that grep will search for all lines starting with any value
- Now if we combine everything together we do:
  - **cat animals2.txt | sort -u | grep -m 16 ^**



```
[02/07/23]seed@VM:~/.../hw1.2$ cat animals2.txt | sort -u | grep -m 16 ^
2-toed sloth
3-toed sloth
50-foot rabbit
9-banded armadillo
a 3-headed dog
aardvark
aardwolf
abalone
abissinian
abominable snowman
abyssinian pony
acanthopholis
acksy
acrocanthosaurus
adder
adlie penguin
[02/07/23]seed@VM:~/.../hw1.2$
```

2. Combine the contents of files verse1.txt, verse2.txt, and verse3.txt into a new file lyrics.txt. Include the full command in your report.

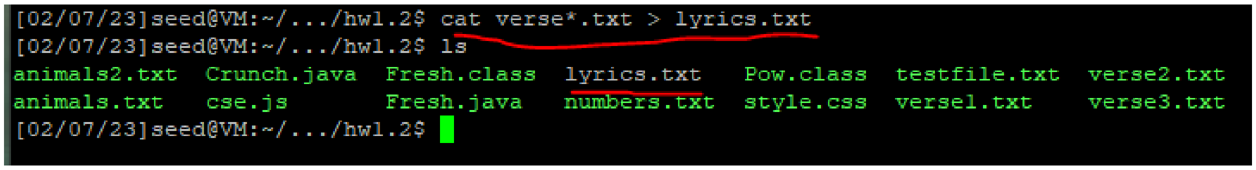
The hw1.2 directory before running the command:



seed@VM: ~/.../hw1.2

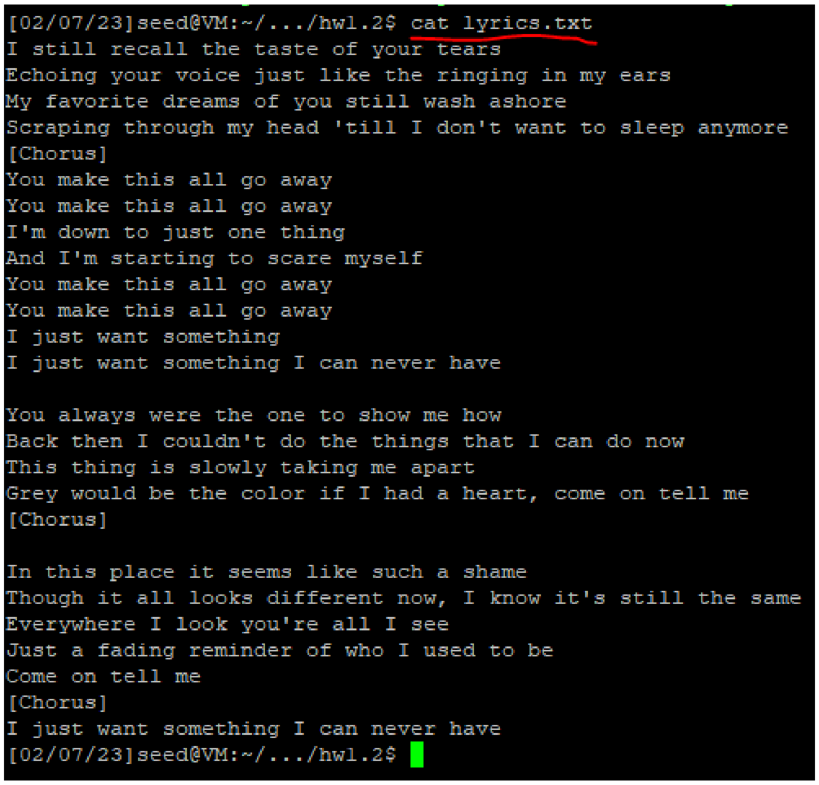
```
[02/07/23]seed@VM:~/.../hw1.2$ cat verse*.txt > lyrics.txt
[02/07/23]seed@VM:~/.../hw1.2$ ls
animals2.txt  Crunch.java  Fresh.class  lyrics.txt   Pow.class  testfile.txt  verse2.txt
animals.txt   cse.js       Fresh.java   numbers.txt  style.css  verse1.txt    verse3.txt
[02/07/23]seed@VM:~/.../hw1.2$ rm lyrics.txt
[02/07/23]seed@VM:~/.../hw1.2$ ls
animals2.txt  Crunch.java  Fresh.class  numbers.txt  style.css     verse1.txt  verse3.txt
animals.txt   cse.js       Fresh.java   Pow.class    testfile.txt  verse2.txt
[02/07/23]seed@VM:~/.../hw1.2$
```

Run this command: **cat verse*.txt > lyrics.txt** (cat will output all contents of files that start with **verse** and end with **.txt**, the then > will redirect contents to a file called lyrics.txt, which was not yet created, thus it will be created); also note lyrics.txt is white because it does not have execute permissions:



```
[02/07/23]seed@VM:~/.../hw1.2$ cat verse*.txt > lyrics.txt
[02/07/23]seed@VM:~/.../hw1.2$ ls
animals2.txt  Crunch.java  Fresh.class  lyrics.txt   Pow.class  testfile.txt  verse2.txt
animals.txt   cse.js       Fresh.java   numbers.txt  style.css  verse1.txt    verse3.txt
[02/07/23]seed@VM:~/.../hw1.2$
```

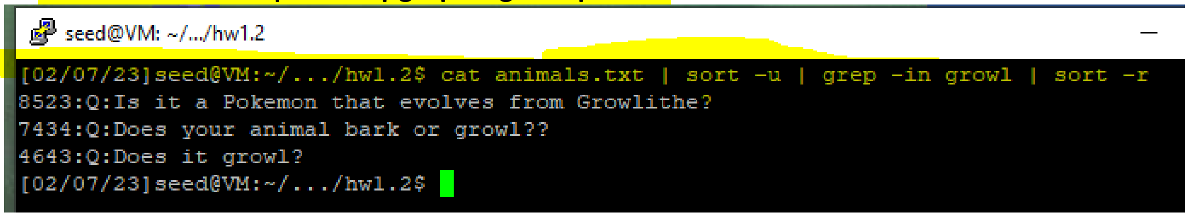
Now here are the contents of lyrics.txt:



```
[02/07/23]seed@VM:~/.../hw1.2$ cat lyrics.txt
I still recall the taste of your tears
Echoing your voice just like the ringing in my ears
My favorite dreams of you still wash ashore
Scraping through my head 'till I don't want to sleep anymore
[Chorus]
You make this all go away
You make this all go away
I'm down to just one thing
And I'm starting to scare myself
You make this all go away
You make this all go away
I just want something
I just want something I can never have

You always were the one to show me how
Back then I couldn't do the things that I can do now
This thing is slowly taking me apart
Grey would be the color if I had a heart, come on tell me
[Chorus]

In this place it seems like such a shame
Though it all looks different now, I know it's still the same
Everywhere I look you're all I see
Just a fading reminder of who I used to be
Come on tell me
[Chorus]
I just want something I can never have
[02/07/23]seed@VM:~/.../hw1.2$
```

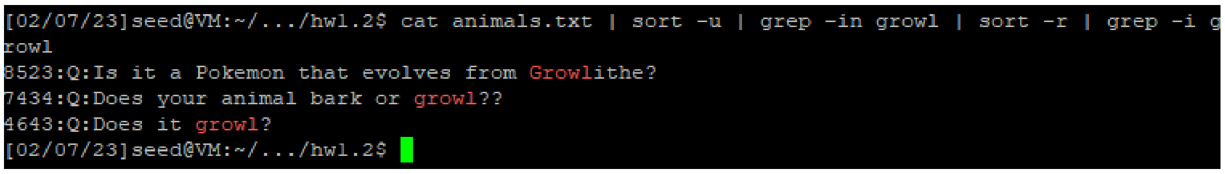
3. Display all lines from animals.txt that contain the word "growl" ignoring case, in reverse-ABC-sorted order and with no duplicates. Output the lines themselves only. Include the full command in your report.

- First, we need to get all of the contents of the **animals.txt** file using **cat** command
- Then, we need to redirect the output from **cat** to **sort** function
    - Then we need to use options for **sort:**
        - -u option: deletes all duplicates of sorted lines
- Then, we redirect **sort** output to **grep** and use options:
    - -i option: ignores case (i.e., e == E)
    - -n option: output line numbers
    - and then we need to search for string "**growl**"
- Finally, we need to redirect output from **grep** to another instance of **sort** function with -r option in order to reverse the sort.
- Now if we combine everything together we do:
- **cat animals.txt | sort -u | grep -in growl | sort -r**


seed@VM: ~/.../hw1.2

```
[02/07/23]seed@VM:~/.../hw1.2$ cat animals.txt | sort -u | grep -in growl | sort -r
8523:Q:Is it a Pokemon that evolves from Growlithe?
7434:Q:Does your animal bark or growl??
4643:Q:Does it growl?
[02/07/23]seed@VM:~/.../hw1.2$
```

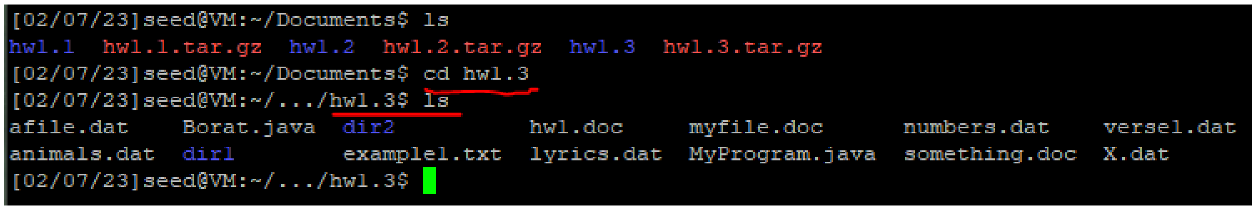
I redirected it one more time to grep command just for sake of color:



```
[02/07/23]seed@VM:~/.../hw1.2$ cat animals.txt | sort -u | grep -in growl | sort -r | grep -i g
rowl
8523:Q:Is it a Pokemon that evolves from Growlithe?
7434:Q:Does your animal bark or growl??
4643:Q:Does it growl?
[02/07/23]seed@VM:~/.../hw1.2$
```

TASK 5: Linux Bash Shell commands 3

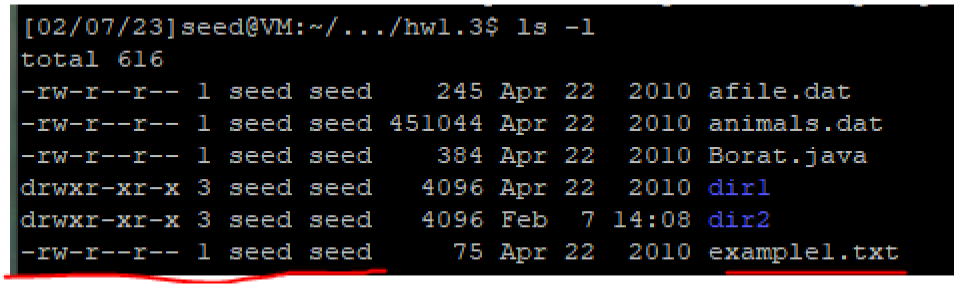
The number statements must be done from within the "/home/seed/Documents/hw1.3" directory. As such, you must use the "cd" command to change into the appropriate directory:



```
[02/07/23]seed@VM:~/Documents$ ls
hw1.1  hw1.1.tar.gz  hw1.2  hw1.2.tar.gz  hw1.3  hw1.3.tar.gz
[02/07/23]seed@VM:~/Documents$ cd hw1.3
[02/07/23]seed@VM:~/.../hw1.3$ ls
afile.dat    Borat.java  dir2          hw1.doc     myfile.doc      numbers.dat    versel.dat
animals.dat  dir1        example1.txt  lyrics.dat  MyProgram.java  something.doc  X.dat
[02/07/23]seed@VM:~/.../hw1.3$
```

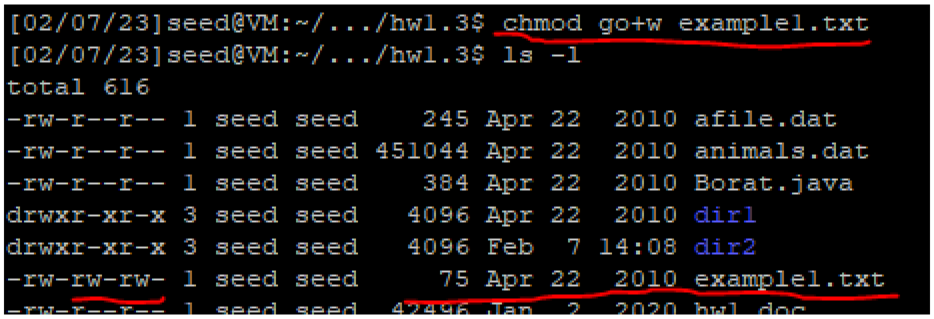
1. Set the file example1.txt in the current directory so that its group and other can write to the file. (You don't need to change any other permissions that might currently be set on the file.). Include the full command in your report.

Here are the original permissions of the file:



```
[02/07/23]seed@VM:~/.../hw1.3$ ls -l
total 616
-rw-r--r-- 1 seed seed    245 Apr 22  2010 afile.dat
-rw-r--r-- 1 seed seed 451044 Apr 22  2010 animals.dat
-rw-r--r-- 1 seed seed    384 Apr 22  2010 Borat.java
drwxr-xr-x 3 seed seed   4096 Apr 22  2010 dir1
drwxr-xr-x 3 seed seed   4096 Feb  7 14:08 dir2
-rw-r--r-- 1 seed seed     75 Apr 22  2010 example1.txt
```

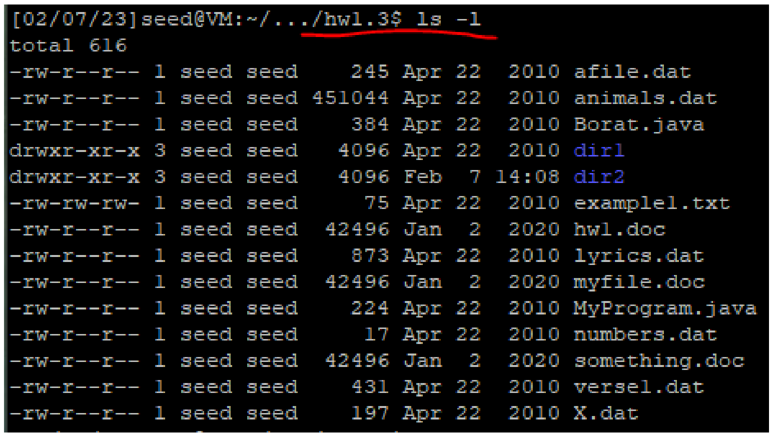
Now run the command: **chmod go+w example1.txt** (where **go** refers to letters g and o for *groups* and *others*, and +w means "add write permissions")



```
[02/07/23]seed@VM:~/.../hw1.3$ chmod go+w example1.txt
[02/07/23]seed@VM:~/.../hw1.3$ ls -l
total 616
-rw-r--r-- 1 seed seed    245 Apr 22  2010 afile.dat
-rw-r--r-- 1 seed seed 451044 Apr 22  2010 animals.dat
-rw-r--r-- 1 seed seed    384 Apr 22  2010 Borat.java
drwxr-xr-x 3 seed seed   4096 Apr 22  2010 dir1
drwxr-xr-x 3 seed seed   4096 Feb  7 14:08 dir2
-rw-rw-rw- 1 seed seed     75 Apr 22  2010 example1.txt
-rw-r--r-- 1 seed seed  42496 Jan  2  2020 hw1.doc
```

2. Set all files with extensions .dat and .doc to be read/writable (but not executable) by their owner, and to have no access from others. (Your command should grant these exact permissions and revoke any others.) Do this using the standard letter code arguments for granting and removing permissions. Include the full command in your report.

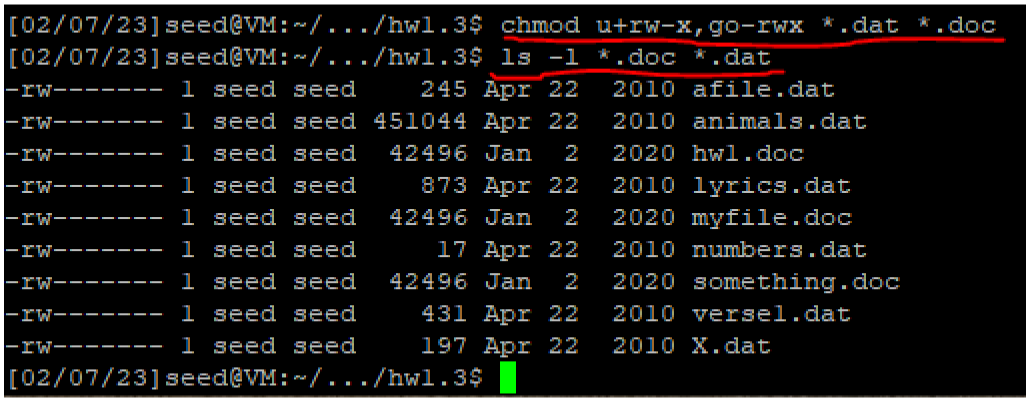
Before changing permissions:



```
[02/07/23]seed@VM:~/.../hw1.3$ ls -l
total 616
-rw-r--r-- 1 seed seed    245 Apr 22  2010 afile.dat
-rw-r--r-- 1 seed seed 451044 Apr 22  2010 animals.dat
-rw-r--r-- 1 seed seed    384 Apr 22  2010 Borat.java
drwxr-xr-x 3 seed seed   4096 Apr 22  2010 dir1
drwxr-xr-x 3 seed seed   4096 Feb  7 14:08 dir2
-rw-rw-rw- 1 seed seed     75 Apr 22  2010 example1.txt
-rw-r--r-- 1 seed seed  42496 Jan  2  2020 hw1.doc
-rw-r--r-- 1 seed seed    873 Apr 22  2010 lyrics.dat
-rw-r--r-- 1 seed seed  42496 Jan  2  2020 myfile.doc
-rw-r--r-- 1 seed seed    224 Apr 22  2010 MyProgram.java
-rw-r--r-- 1 seed seed     17 Apr 22  2010 numbers.dat
-rw-r--r-- 1 seed seed  42496 Jan  2  2020 something.doc
-rw-r--r-- 1 seed seed    431 Apr 22  2010 versel.dat
-rw-r--r-- 1 seed seed    197 Apr 22  2010 X.dat
```

Run the command: **chmod u+rw-x,go-rwx *.dat *.doc** (this adds rw but no x for user/owner, then no rwx for g and o, and do this for all files ending with .dat and .doc)

Now I will output contents of hw1.3 directory of only .dat and .doc files, showing the permissions:



```
[02/07/23]seed@VM:~/.../hw1.3$ chmod u+rw-x,go-rwx *.dat *.doc
[02/07/23]seed@VM:~/.../hw1.3$ ls -l *.doc *.dat
-rw------- 1 seed seed    245 Apr 22  2010 afile.dat
-rw------- 1 seed seed 451044 Apr 22  2010 animals.dat
-rw------- 1 seed seed  42496 Jan  2  2020 hw1.doc
-rw------- 1 seed seed    873 Apr 22  2010 lyrics.dat
-rw------- 1 seed seed  42496 Jan  2  2020 myfile.doc
-rw------- 1 seed seed     17 Apr 22  2010 numbers.dat
-rw------- 1 seed seed  42496 Jan  2  2020 something.doc
-rw------- 1 seed seed    431 Apr 22  2010 versel.dat
-rw------- 1 seed seed    197 Apr 22  2010 X.dat
[02/07/23]seed@VM:~/.../hw1.3$
```

3. The file /etc/passwd stores a list of all users' names and user account names on the system, along with a bit of other information such as what shell program they use. (The default shell for most users, and the one we have been learning about in this course is the Bash shell, stored in the file /bin/bash) **How many users exist on this Linux system that use the Bash shell by default?**
(Hint: To figure this out, you will need to search for lines in the passwd file that mention bash.)

You may assume that no line of /etc/passwd contains the phrase "/bash" other than to specify the Bash shell). Include the full command in your report.

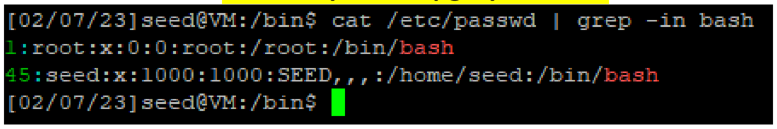
The following information I got from https://www.ibm.com/docs/bg/aix/7.2?topic=passwords-using-etcpasswd-file to help me use for referencing what each field means:

Traditionally, the /etc/passwd file is used to keep track of every registered user that has access to a system.

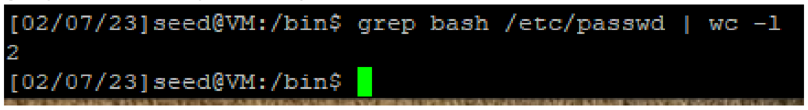
The /etc/passwd file is a colon-separated file that contains the following information:

- *User name*
- *Encrypted password*
- *User ID number (UID)*
- *User's group ID number (GID)*
- *Full name of the user (GECOS)*
- *User home directory*
- *Login shell*

Example: root:x:0:0:root:/root:/bin/bash

- First, **cat** the passwd file
- Then, pipe output of **cat** as input to **grep** using -in options to ignore case and to output line numbers (just for visual convenience) and search for value "bash"
- Thus the full command is: **cat /etc/passwd | grep -in bash**
  - 

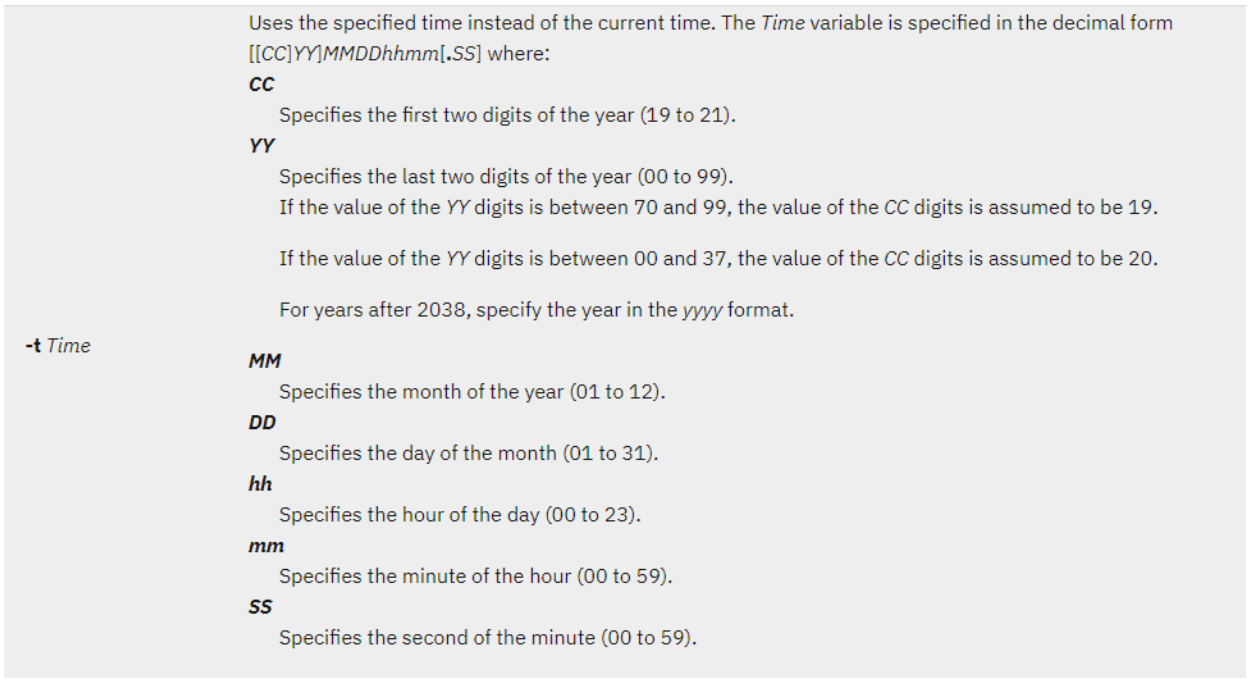
```
[02/07/23]seed@VM:/bin$ cat /etc/passwd | grep -in bash
1:root:x:0:0:root:/root:/bin/bash
45:seed:x:1000:1000:SEED,,,:/home/seed:/bin/bash
[02/07/23]seed@VM:/bin$
```

- Thus only 2 users use the bash shell by default: **root**, and **seed**.
- The following command will output number of lines returned by grep from using wc -l:
  - **grep bash /etc/passwd | wc -l**
  - 

```
[02/07/23]seed@VM:/bin$ grep bash /etc/passwd | wc -l
2
[02/07/23]seed@VM:/bin$
```

4. Create a file foo.txt using the "touch" command. Change the file's last-modified date to be January 4 of this year at 8:56am. Include the full command in your report.

I will use this reference from https://www.ibm.com/docs/en/aix/7.1?topic=t-touch-command in order to format the time for the touch command:



-t *Time*

Uses the specified time instead of the current time. The *Time* variable is specified in the decimal form [[*CC*]*YY*]*MMDDhhmm*[**.***SS*] where:

CC
Specifies the first two digits of the year (19 to 21).

YY
Specifies the last two digits of the year (00 to 99).
If the value of the *YY* digits is between 70 and 99, the value of the *CC* digits is assumed to be 19.

If the value of the *YY* digits is between 00 and 37, the value of the *CC* digits is assumed to be 20.

For years after 2038, specify the year in the *yyyy* format.

MM
Specifies the month of the year (01 to 12).

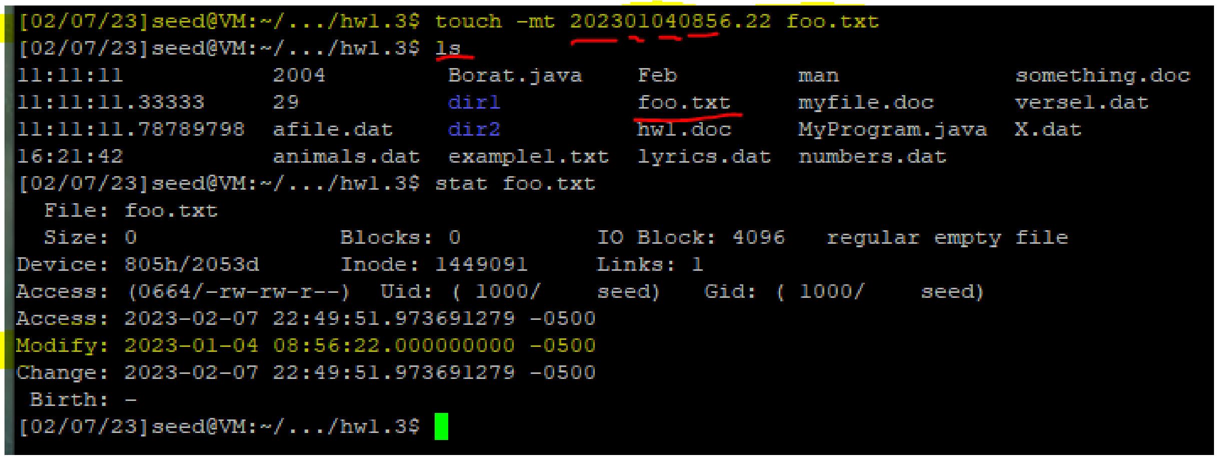
DD
Specifies the day of the month (01 to 31).

hh
Specifies the hour of the day (00 to 23).

mm
Specifies the minute of the hour (00 to 59).

SS
Specifies the second of the minute (00 to 59).

- I will use command **touch** command and use options -m (modify only the date modified time) and -t to set the date based on the -m flag, and using the format as shown above when -t is used:
- **touch -mt 202301040856.22 foo.txt**
- Notice in the screenshot below when I run **stat** command, date Modified is the only thing changed on foo.txt to the jan 4 2023 8:56am time:



```
[02/07/23]seed@VM:~/.../hw1.3$ touch -mt 202301040856.22 foo.txt
[02/07/23]seed@VM:~/.../hw1.3$ ls
11:11:11            2004          Borat.java    Feb         man            something.doc
11:11:11.33333      29            dir1          foo.txt     myfile.doc     versel.dat
11:11:11.78789798   afile.dat     dir2          hw1.doc     MyProgram.java X.dat
16:21:42            animals.dat   example1.txt  lyrics.dat  numbers.dat
[02/07/23]seed@VM:~/.../hw1.3$ stat foo.txt
  File: foo.txt
  Size: 0               Blocks: 0          IO Block: 4096   regular empty file
Device: 805h/2053d      Inode: 1449091     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/    seed)   Gid: ( 1000/    seed)
Access: 2023-02-07 22:49:51.973691279 -0500
Modify: 2023-01-04 08:56:22.000000000 -0500
Change: 2023-02-07 22:49:51.973691279 -0500
 Birth: -
[02/07/23]seed@VM:~/.../hw1.3$
```

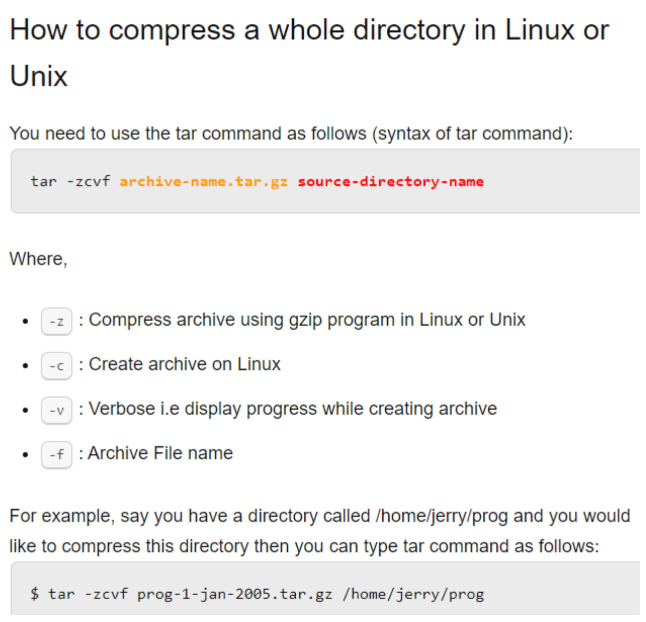
Conclusion

Overall, I learned how to greatly express the power of Linux command line interface using the Bash shell. I especially enjoy the power of the pipe and redirect operators as I see how they can be combined into very elaborate and useful statements that give me some desired output. I think one of the most powerful commands to utilize and understand is undoubtedly the **grep** command. It is very extensive in its options and functionality for a reason, since all file systems rely on regular expressions that humans require in order to use and manage a file system (particularly searching and modifying the file system). Thus, if you can understand how to use **grep, cat,** and the **redirect >** and **pipe |** operators, you will likely be able to understand most of all of the other commands/functions, and be able to harness (through combining) the power of these fundamental commands and operators as combined also with other commands.

From README.TXT FILE

I use this source to help me compress https://www.cyberciti.biz/faq/how-do-i-compress-a-whole-linux-or-unix-directory/ :

1) To compress the assg1_complete.tar.gz file:



How to compress a whole directory in Linux or Unix

You need to use the tar command as follows (syntax of tar command):

```
tar -zcvf archive-name.tar.gz source-directory-name
```

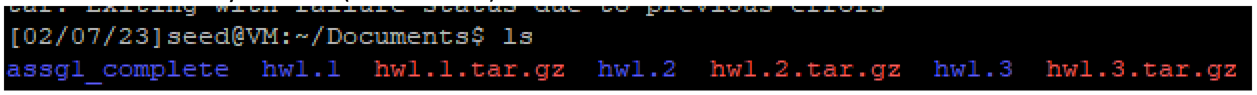
Where,

- `-z` : Compress archive using gzip program in Linux or Unix
- `-c` : Create archive on Linux
- `-v` : Verbose i.e display progress while creating archive
- `-f` : Archive File name

For example, say you have a directory called /home/jerry/prog and you would like to compress this directory then you can type tar command as follows:

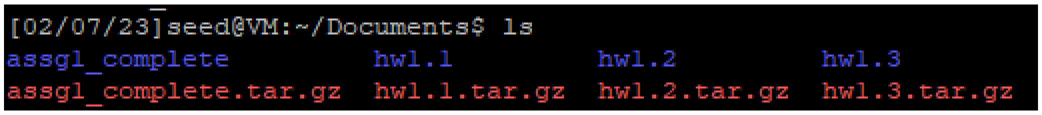
```
$ tar -zcvf prog-1-jan-2005.tar.gz /home/jerry/prog
```

- Here, I will explain how I compressed (and how you can decompress) the compressed file assg1_complete.tar.gz that I uploaded to CANVAS. It contains the modified versions of hw1.1, hw2.2, and hw1.3 that resulted from me working in them upon completion of this lab.
- To compress, I ran the following command:
    - **tar -zcvf assg1_complete.tar.gz ./hw1.1 ./hw1.2 ./hw1.3**
- This is the directory I was in (Documents)



```
[02/07/23]seed@VM:~/Documents$ ls
assg1_complete  hw1.1  hw1.1.tar.gz  hw1.2  hw1.2.tar.gz  hw1.3  hw1.3.tar.gz
```

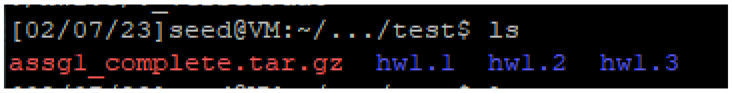
- After running the above command, this is the result:

- 

```
[02/07/23]seed@VM:~/Documents$ ls
assg1_complete         hw1.1         hw1.2         hw1.3
assg1_complete.tar.gz  hw1.1.tar.gz  hw1.2.tar.gz  hw1.3.tar.gz
```

- You see the .tar.gz file is present.

2) To decompress (extract) the file contents of my assg1_complete.tar.gz file:

- Run the command: **tar -xvf assg1_complete.tar.gz (note, x flag is for "extract")**
- This will extract contents of the tar file into the current directory of where the tar file is stored.
- In my case, I created a directory called **test** , and made a copy of the tar file into that directory, then navigated to that directory and ran the command.
- Then **test** directory has these contents after extraction of tar file:
- 

```
[02/07/23]seed@VM:~/.../test$ ls
assg1_complete.tar.gz  hw1.1  hw1.2  hw1.3
```

- As you can see, my tar file contains the 3 directories that I worked in as required for the assignment.