# Unit 2 Lab - Essential Tools Files, Redirects, and Permissions

**Required Materials**

Putty or other connection tool

Lab Server

Root or sudo command access

**EXERCISES (Warmup to quickly run through your system and familiarize yourself)**

1. cd ~
2. ls
3. mkdir evaluation
4. mkdir evaluation/test/round6  
   This fails, can you find out why?
5. mkdir -p evaluation/test/round6  
   This works, think about why?
6. cd evaluation
7. pwd  
   What is the path you are in?
8. touch testfile1
9. ls  
   What did this do?
10. touch testfile{2..10}
11. ls  
    What did this do differently than earlier?
12. touch .hfile .hfile2 .hfile3
13. ls  
    Can you see your newest files? Why or why not? (man ls)  
    What was the command to let you see those hidden files?
14. ls –l   #what do you know about this long listing? Think about 10 things this can show you. Did it show you all the files or are some missing?

**LAB**

This lab is designed to help you get familiar with the basics of the systems you will be working on. Some of you will find that you know the basic material but the techniques here allow you to put it together in a more complex fashion.

It is recommended that you type these commands and do not copy and paste them. Word sometimes likes to format characters and they don't always play nice with Linux.

1. Gathering system information
    a. hostname
    b. cat /etc/*release
       What do you recognize about this output? What version of RHEL (CENTOS) are we on?
    c. uname
    d. uname -a
    e. uname –r

       man uname to see what those options mean if you don't recognize the values

2. Check the amount of RAM
    a. cat /proc/meminfo
    b. free
    c. free –m
       What do each of these commands show you? How are they useful?

3. Check the number of processors and processor info
    a. cat /proc/cpuinfo
       What type of processors do you have? How many are there? (counting starts at 0)
    b. cat /proc/cpuinfo | grep proc | wc –l
       Does this command accurately count the processors?

4. Check Storage usage and mounted filesystems
    a. df
       But df is barely readable, so find the option that makes it more readable `man df`
    b. df -h
    c. df -h | grep –i var
       What does this show, or search for? Can you invert this search? (hint `man grep` look for invert or google "inverting grep's output")
    d. df –h | grep –i sd
       This one is a little harder, what does this one show? Not just the line, what are we checking for? (hint if you need it, google "what is /dev/sda in linux")
    e. mount

Mount by itself gives a huge amount of information. But, let’s say someone is asking you to verify that the mount is there for /home on a system. Can you check that quickly with one command?
mount | grep –i home
#This works, but there is a slight note to add here. Just because something isn’t individually mounted doesn’t mean it doesn’t exist. It just means it’s not part of it’s own mounted filesystem.
    mount | grep –i /home/xgqa6cha
    will produce no output
    df –h /home/xgqa6cha
    will show you that my home filesystem falls under /home.

f. cd ~; pwd; df -h .
   This command moves you to your home directory, prints out that directory, and then shows you what partition your home directory is on.
g. du –sh .
   will show you space usage of just your directory
   try `du –h .` as well to see how that ouput differs
   read `man du` to learn more about your options.

5. Check the system uptime
   a. uptime
   b. man uptime
      Read the man for uptime and figure out what those 3 numbers represent. Referencing this server, do you think it is under high load? Why or why not?

6. Check who has recently logged into the server and who is currently in
   a. last
      Last is a command that outputs backwards. (Top of the output is most recent). So it is less than useful without using the more command.
   b. last | more
      Were you the last person to log in? Who else has logged in today?
   c. w
   d. who
   e. whoami
      how many other users are on this system? What does the pts/0 mean on google?

7. Check who you are and what is going on in your environment
   a. printenv
      This scrolls by way too fast, how would you search for your home?

b. printenv | grep –i home
c. whoami
d. id
e. echo $SHELL

8. Check running processes and services
   a. ps –aux | more
   b. ps –ef | more
   c. ps –ef | wc –l

9. Check memory usage and what is using the memory
   a. Run each of these individually for understanding before we look at part b.
      free –m
      free –m | egrep “Mem|Swap”
      free –m| egrep “Mem|Swap” | awk ‘{print $1, $2, $3}’

   b. free -t | egrep "Mem|Swap" | awk '{print $1 " Used Space = " ($3 / $2) * 100"%"}'
      Taking this apart a bit:
      You’re just using free and searching for the lines that are for memory and swap
      You then print out the values $1 = Mem or Swap
      You then take $3 used divided by $2 total and multiply by 100 to get the percentage

10. Have you ever written a basic check script or touched on conditional statements or loops? (Use ctrl + c to break out of these)
    a. while true; do free -m; sleep 3; done
       Watch this output for a few and then break with ctrl + c
       Try to edit this to wait for 5 seconds
       Try to add a check for uptime and date each loop with a blank line between each and 10 second wait:
       while true; do date; uptime; free -m; echo “ “; sleep 10; done
    b. Since we can wrap anything inside of our while statements, let’s try adding something from earlier:
       While true; do free -t | egrep "Mem|Swap" | awk '{print $1 " Used Space = " ($3 / $2) * 100"%"}'; sleep 3; done

    c. seq 1 10
       What did this do?
       Can you man seq to modify that to count from 2 to 20 by 2’s?
    d. Let’s make a counting for loop from that sequence
       for i in `seq 1 20`; do echo "I am counting i and am on $i times through the loop"; done

Can you tell me what is the difference or significance of the $ in the command above? What does that denote to the system?