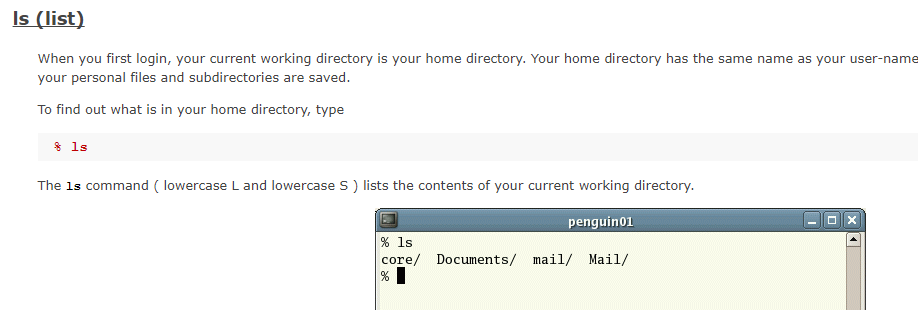
# Linux Lab 1 Key



# Basic File Commands

1) Open a terminal

2) Change to the root directory.

[john@localhost ~]$ cd /

3) Do an "ls" command while you are in the root of the file system tree to see what's there. Note that there is a directory at the root (/) called "root". This is the home directory for the root user, not the root of the tree.

[john@localhost /]$ ls

bin boot dev etc home lib lib64 lost+found media misc mnt net opt proc root sbin selinux srv sys tmp usr

4) Change directory to home, then do "ls" to see what's there. You probably have only your user home directory, named the same as your user name. Change directory into your home directory, the one named after your user.

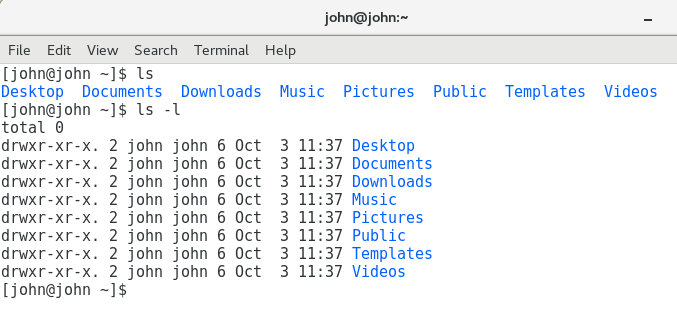
[john@localhost /]$ cd home  
[john@localhost home]$ ls  
john  
[john@localhost home]$ cd john  
[john@localhost ~]$

5) Use "pwd" (print working directory) to see the path to the directory you are in. Hopefully it is /home/<your user name>.

[john@localhost ~]$ pwd  
/home/john

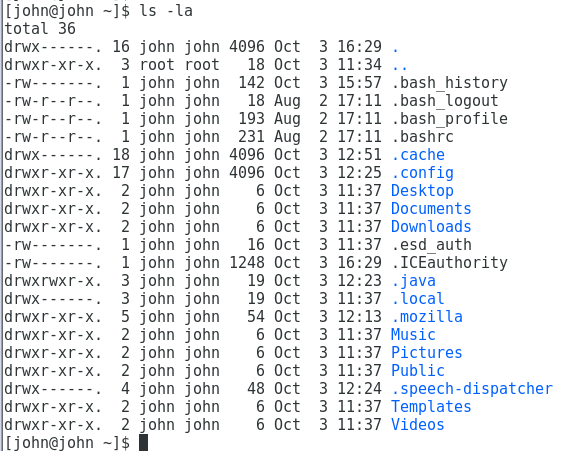
Note: /home/john is the full, or absolute, path to the home directory for the user john. In the commands for step 4, we were able to enter just home,

6) In your home directory, do "ls", then "ls -l", and compare them. What extra information do you see?

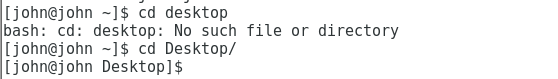
You see this information in the “-l” long format:

* file permissions,
* number of links,
* owner name,
* owner group,
* file size,
* time of last modification, and
* file/directory name

7) In your home directory, do "ls -la". This shows a lot of hidden directories that contain configuration information for your profile and programs. (Lots of forensics information about what the user has done can be gleaned from these directories.) How are the hidden directories/files named differently from those that are not hidden?

  
File names of hidden files start with “.”.

8) Change directory to your desktop. Note that cd desktop does not work, but cd Desktop does. Linux is case sensitive, Windows is not.



9) Create two new directories on your desktop using mkdir. You should see the new directories appear on your desktop.

[john@john Desktop]$

[john@john Desktop]$ mkdir dir1 dir2

[john@john Desktop]$

10) Create two new files inside the first directory you made. Create the first file using the "touch" command and the second by using the "echo" command and redirection.

[john@john Desktop]$ cd dir1

[john@john dir1]$ touch file1

[john@john dir1]$ echo "stuff" > file2

[john@john dir1]$ ll

total 4

-rw-rw-r--. 1 john john 0 Oct 12 13:28 file1

-rw-rw-r--. 1 john john 6 Oct 12 13:29 file2

[john@john dir1]$

11) Copy the files to the second directory.

Any of the next three lines will work—you only have to use one

[john@john dir1]$ cp \* ../dir2

[john@john dir1]$ cp \* ~/Desktop/dir2/

[john@john dir1]$ cp /home/john/Desktop/dir1/\* /home/john/Desktop/dir2/

12) Delete the first directory. Remember that you'll have to empty it first.

Oops, did dir2 by mistake

[john@john dir1]$ cd ../dir2

[john@john dir2]$ ls

file1 file2

[john@john dir2]$ rm \*

[john@john dir2]$ cd ..

[john@john Desktop]$ rmdir dir2

Might as well do dir1 as well…more than one way to do it.

[john@john Desktop]$ ls -l

total 0

drwxrwxr-x. 2 john john 32 Oct 12 13:29 dir1

[john@john Desktop]$ rm dir1

rm: cannot remove ‘dir1’: Is a directory

[john@john Desktop]$ rm -rf dir1

[john@john Desktop]$ ls -l

total 0

13) Open Firefox, and download an image (picture). By default, it will be put in your /home/<username>/Downloads directory.

14) Change directory to ~/Downloads. What does that do, and what does the ~ do?

[john@john /]$ cd ~/Downloads/

[john@john Downloads]$ pwd

/home/john/Downloads

15) List the contents of your Downloads directory. Hopefully, your picture is there.

[john@john Downloads]$ ls -l

total 996

-rw-rw-r--. 1 john john 1019563 Oct 10 09:16 suit.jpg

16) Run "file <name of your picture>". Does the information seem correct?

[john@john Downloads]$ file suit.jpg

suit.jpg: JPEG image data, EXIF standard

[john@john Downloads]$

17) Get a list of processes running on your machine by using "ps aux". To see just one page at a time, pipe the output from "ps aux" into "more", using this: "ps aux | more". Try it again using "ps aux | less". (Less is an improved version of more.)

[john@john Downloads]$ ps aux | less

<then this comes up>

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

root 1 0.1 0.1 193700 6840 ? Ss 12:19 0:06 /usr/lib/systemd/systemd --switched-root --system --deserialize 21

root 2 0.0 0.0 0 0 ? S 12:19 0:00 [kthreadd]

root 3 0.0 0.0 0 0 ? S 12:19 0:00 [ksoftirqd/0]

root 5 0.0 0.0 0 0 ? S< 12:19 0:00 [kworker/0:0H]

root 7 0.0 0.0 0 0 ? S 12:19 0:00 [migration/0]

root 8 0.0 0.0 0 0 ? S 12:19 0:00 [rcu\_bh]

root 9 0.0 0.0 0 0 ? S 12:19 0:01 [rcu\_sched]

root 10 0.0 0.0 0 0 ? S 12:19 0:00 [watchdog/0]

:

<many more lines>

18) Our terminal is a bash shell, so we should have a bash process in there somewhere. You can filter it from the noise by piping the output from "ps aux" into the great Linux search program "grep." Try " ps aux | grep bash ".

[john@john Downloads]$ ps aux | grep bash

root 741 0.0 0.0 115256 936 ? S 12:19 0:00 /bin/bash /usr/sbin/ksmtuned

john 1901 0.0 0.0 51332 580 ? Ss 12:20 0:00 /usr/bin/ssh-agent /bin/sh -c exec -l /bin/bash -c "env GNOME\_SHELL\_SESSION\_MODE=classic gnome-session --session gnome-classic"

john 2452 0.0 0.0 116296 3124 pts/0 Ss 12:20 0:00 bash

john 4610 0.0 0.0 112664 972 pts/0 R+ 13:44 0:00 grep --color=auto bash

[john@john Downloads]$

Process 2452 run by the john user is the bash terminal that is open. The one below it, 4610, is me searching with grep.

19) Note: The ss application has replaced netstat, although you will see netstat referenced in many books and blogs. If you want to practice with netstat, you will have to install it by typing:  
sudo apt install net-tools in Ubuntu  
sudo yum install net-tools in CentOS  
(this is optional, ss works fine.)

Use ss -l (-l is a lower-case L, for listening) to get a list of listening network ports. Unix uses network sockets or streams for internal communications so this output will be very noisy. You can filter out the internal connection noise by just looking for TCP and UDP connections, which are connections with other computers.

john@svgs-f20-1:~$ ss -l

Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port Process

nl UNCONN 0 0 rtnl:evolution-calen/2420 \*

nl UNCONN 0 0 rtnl:systemd-resolve/653 \*

nl UNCONN 0 0 rtnl:NetworkManager/737 \*

nl UNCONN 0 0 rtnl:xdg-desktop-por/2289 \*

nl UNCONN 0 0 rtnl:goa-daemon/2119 \*

(many line omitted.)

Use ss –help to see the options that ss allows. What would the command ss -nat do? Once you have made a guess, execute ss -nat to see what it does.

john@svgs-f20-1:~/Desktop$ ss --help

Usage: ss [ OPTIONS ]

ss [ OPTIONS ] [ FILTER ]

-h, --help this message

-V, --version output version information

-n, --numeric don't resolve service names

-r, --resolve resolve host names

-a, --all display all sockets

-l, --listening display listening sockets

-o, --options show timer information

-e, --extended show detailed socket information

-m, --memory show socket memory usage

-p, --processes show process using socket

-i, --info show internal TCP information

--tipcinfo show internal tipc socket information

-s, --summary show socket usage summary

--tos show tos and priority information

-b, --bpf show bpf filter socket information

-E, --events continually display sockets as they are destroyed

-Z, --context display process SELinux security contexts

-z, --contexts display process and socket SELinux security contexts

-N, --net switch to the specified network namespace name

-4, --ipv4 display only IP version 4 sockets

-6, --ipv6 display only IP version 6 sockets

-0, --packet display PACKET sockets

-t, --tcp display only TCP sockets

-S, --sctp display only SCTP sockets

-u, --udp display only UDP sockets

-d, --dccp display only DCCP sockets  
(some lines omitted)

john@svgs-f20-1:~/Desktop$ ss -nat

State Recv-Q Send-Q Local Address:Port Peer Address:Port Process

LISTEN 0 4096 127.0.0.53%lo:53 0.0.0.0:\*

LISTEN 0 5 127.0.0.1:631 0.0.0.0:\*

LISTEN 0 5 [::1]:631 [::]:\*

You should find that ss -nat lists all connections (-a) but limits the display to TCP connections (-t). The -n causes ss to show port numbers as numbers and not by names (53 vs. dns, for example.)

Now execute an ss command that will show all UDP ports that are open.

john@svgs-f20-1:~$ ss -nau

State Recv-Q Send-Q Local Address:Port Peer Address:Port Process

UNCONN 0 0 127.0.0.53%lo:53 0.0.0.0:\*

ESTAB 0 0 192.168.77.129%ens33:68 192.168.77.254:67

UNCONN 0 0 0.0.0.0:53677 0.0.0.0:\*

UNCONN 0 0 0.0.0.0:631 0.0.0.0:\*

UNCONN 0 0 0.0.0.0:5353 0.0.0.0:\*

UNCONN 0 0 [::]:57867 [::]:\*

UNCONN 0 0 [::]:5353 [::]:\*

Note: We will discuss TCP, UDP, and port numbers in great detail when we reach the networking modules.

20) Execute the following commands:

[john@localhost ~]$ ls

centfile.txt file other pwgen~ Templates

Desktop JohnPublicKey Pictures pwlist Unsaved Document 1~

Documents kalifile.txt Public ssltest.py Videos

Downloads Music pwgen stuff-file

[john@localhost ~]$ mkdir testdir

[john@localhost ~]$ cd testdir/

[john@localhost testdir]$ touch empty file

[john@localhost testdir]$ ls -l empty

-rw-rw-r--. 1 john john 0 Aug 25 13:30 empty

[john@localhost testdir]$ ls -l file

-rw-rw-r--. 1 john john 0 Aug 25 13:30 file

[john@localhost testdir]$ cd ..

[john@localhost ~]$ rmdir testdir

rmdir: failed to remove `testdir': Directory not empty

[john@localhost ~]$

What did the command, "touch empty file", do?

It made two separate files, “empty” and “file”.

Why did rmdir fail? What would you do to correct it?

The directory was not empty.

Come up with two ways to delete the testdir in one line. (Hint: One way I can

think of involves a ";". Another involves rm. Any help methods, including

Google, are fair game.) Test your answers to be sure they work.

rm -rf testdir  
rm testdir/\*; rmdir testdir

21) Why did this fail?

[john@localhost ~]$ echo 'lots of stuff here' > Myfile

[john@localhost ~]$

[john@localhost ~]$

[john@localhost ~]$ cat MyFile

cat: MyFile: No such file or directory

[john@localhost ~]$

Linux is case sensitive, so Myfile is different from MyFile.