Craig Celestin
Linux for Noobs Challenges
Due 07/17/2021 at midnight

NOTES IN PROGRESS
Intro to Linux
Linux = operating sys like Windows, MacOS - began in the 50s when OS ran batch processes, batch program w/o user interaction, read data from file/punch card then output to other file/printer. In 60s, interactive OS became popular allowing multiple users from different terminal to operate host at the same time (time-sharing OS), which challenged batch OS (dev of Unix)
Unix = free original release, then implemented TCP/IP protocol stack (early workstation OS).
90s -Unix = mainstream OS in server market, became commercialized. Linux became a competitor OS, simple, acquired attractive features. Linux = kernel of OS, basis of other running programs, implement multi-tasking, hardware management. Free apps w/ installation running on top of Kernel, required: CLI (user interaction, write shell scripts)
C compiler derived from GNU project of Free Software Foundation (began in 1984 with aim of developing free UNIX-like OS) Linux = GNU/Linux sys, 92-03 (TCP/IP Network, GUI sys attracted industry)

Linux Desktop Envi - no graphical interface (was xFree86), now there is a GUI in current distribution (Xorg). Desktop envi achieved via X Window System (X) = toolkit + architecture agreement, Xorg = framework of X implementation. Xorg = server providing GUI services like Apache. Achieving a complete desktop envi requires X client

Linux Terminal - Operate Linux sys via Shell. Linux sys provides a terminal emulator (the Terminal), Terminal and console are different (terminal - /dev/tty device on Linux where multi-user function is done. Linux provides 6 virtual consoles (CLI ‘terminals’ allowing user login)
In GUI, differences in user experience are caused by shell differences, UNIX/Linux kernel. Shell is a ‘provided user interface’ software, accepts user input commands. Shell - hides details of underlying OS (GNOME/KDE - 'virtual shell’/’graphical shell’), scripting language controls system. Shell = script interpreter that controls system boot, X11 boot, other utilities
Popular shells = Bash (sh), zsh, ksh, csh/Default shell = Bash, default desktop envi = GNOME/Unity

Input, Out, Err - Commands are most important (three streams)
touch - create file, cd /etc/ change to etc directory, pwd - print current directory
Tab - auto completes a command, $tail - next input has no reaction, $find / - continuous output
Ctrl+C (quits running command)
Shortcuts: ctrl+d (keyboard input ends/exits terminal), ctrl+s (pause current program, press key to resume), ctrl+z (put current program into back), ctrl+a (move cursor to beginning of input line, home equivalent), ctrl+e (move cursor to end of input line, end equivalent), ctrl+k (remove content from place cursor is to end of line), alt+backspace (delete word forward), shift+pgup(scroll up terminal), shift + pgdn (scroll down terminal)

Wildcards - special expression (* and ? matches strings), search folder w/ find command, use it instead / Wildcard input is handled in terminal by shell - search for possible matches as path/file name (match exists, replaced w/ path extension) - shell implementation of path extension function
$ls *.txt - list files ending in .txt
$touch love_{1..10}_linuxt.xt (create multiple files at once)
Common shell wildcards -

`*`	Match any zero or more characters
`?`	Match any single character
`[list]`	Match any single character of the list
`[!list]`	Match characters other than any single character of list
`[c1-c2]`	Match any single character in the range c1 to c2 such as [0-9], [a-z]
`{string1,string2,...}`	Match string1 or string2 (or more) with a string

`{c1..c2}`	Match all characters or numbers in the range c1 to c2 such as {1..10}

Section	Description
1	General order
2	System call
3	Library functions, covering the C standard library
4	Special files (usually devices in /dev) and drivers
5	File format and convention
6	Games and screensavers
7	Others
8	System management commands and daemons

man command sections: frmt (man 1 ls)
Format - name, synopsis, description, examples,
Specific parameters of a function: ls --help
Lab 0 - banner (graphic character output) - sudo apt-get update, sudo ap-get install sysvbanner
(2) banner labex or printerbanner -w 50 A
(-w *specifies print width*)

Linux User/Group + File Permissions

(1) Due to Linux user management, authority management, different users can't easily modify the files of others

$ who am i/$ who mom likes - indicates user name of user opening pseudo terminal
pts=pseudo terminal (relative to /dev/tty devices)
pts/#-corresponds to terminal number

Parameters of `who`:

Parameter	Description
`-a`	Print all information
`-d`	Print dead process
`-m`	Same as `am i`, `mom likes`
`-q`	Get number of users logged in and their user names
`-u`	Get the list of users logged in
`-r`	Get the current run level

Root account has supreme right of system like create/add users - root permission = system privilege like SYSTEM privilege, higher than admin privileges
Root =super admin user account, operate all system subjects / add, delete, modify, search for any file
Sudo can get root privileges, sudo command needs 2 reqs: password of user logged in, user must be in sudo user group
Su, su-, sudo
su<a> - switch to user a, command execution requires user password, run cm command at privilege level w/ sudo <cmd> require current user to belong to sudo group, enter current user password
Su - <user> (switch users) - $sudo adduser jack, enter custom pass
ls /home - shows home directory created for new user, log into new account - (su -l jack)
Each user has a home (user group) - collection of users who share resources, permission, private resources - similar to family structure
How to view user group name - $groups labex
You can view the /etc/sudoers.d/labex file. We have created this file in the /etc/sudoers.d directory to give sudo permissions to user label
$ cat /etc/group | sort (cat - read contents of specified file and print out, | sort - text sorted and output by dict sort (dictionary sort) (view contents of /etc/group file)
$ cat /etc/group | grep -E "labex" - filter out results from initial search for just labex
Each user group has a record - group_name:password:GID:user_list
Invite new user to join sudo user group for root privileges: su -l jack -> sudo ls
(after warning) user usermod to add user to user group /use the root user to add other users to user group
exit -> sudo groups jack -> sudo usermod -G sudo jack -> sudo groups jack
Delete a user: $ sudo deluser jack --remove-home

File Permissions
ls -l (list files in long format)
file type - everything in Linxus is a file, file permissions: read permission (r) : cat <file name> , read contents of a file
Write permission (w): read file contents, execute permission (x): refers to binary program file/script file (exe file on Windows), a directory w/ read and execute permissions can open and view contained files - only dir w/ write permissions can create other files in them
# of links - number of files linked to inode, file size - node size to rep file size (ls, -lh parameter view file sizes)
ls -A (display all files except . (current dir), .. (parent dir)/ ls -Al (use -A and -l parameters..list all files in long format)
view full dir properties: ls -dl <directory>/ ls -AsSh (show all file sizes) / ‘s’ (used for file size display), ‘S’ (sort file by file size)

Login as jack: sudo adduser jack -> su -l jack -> touch iphone2 -> cd /home/jack -> ls iphone2 -> exist -> sudo groups jack -> sudo usermod -G sudo jack -> sudo groups jack -> su -l jack (password) -> sudo chown labex iphone 2 (change file owner of ‘labex’) -> ll iphone 2 (file owner modified for labex)

Modify File Permissions
Document you don’t want other users to read, write, execute - 2 methods
Method 1: Binary # representation (file has 3 groups of permissions - owner, user group, others) corresponding to ‘rwx’ triplet. File ‘iphonex’ permissions changed to ‘only for owner to use’
echo “echo \”hello labex”\”” > iphone2 -> chmod 700 iphone2 (allow other users to read iphone2 file)
same results: chmod go-rw iphone2 (u,g,o represent user (file owner), group, others)
+, - (add/remove corresponding permissions)

Addition
useradd v adduser (useradd - only creates user, need to use passwd to set up password for newuser and then command adduser - creates user, directory, password)

```
labex:~/ $ sudo adduser labextest                                    [2:52:15]
Adding user `labextest' ...
Adding new group `labextest' (1000) ...
Adding new user `labextest' (1000) with group `labextest' ...
Creating home directory `/home/labextest' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for labextest
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
labex:~/ $ su -l labextest                                           [2:52:54]
Password:
labextest@91fabe4d1ec1:~$ touch /opt/forlabex
touch: cannot touch '/opt/forlabex': Permission denied
labextest@91fabe4d1ec1:~$ exit
logout
labex:~/ $ sudo groups labextest                                     [2:54:06]
labextest : labextest
labex:~/ $ sudo usermod -G sudo labextest                            [2:54:14]
labex:~/ $ sudo groups labextest                                     [2:54:27]
labextest : labextest sudo
labex:~/ $ su -l jack                                                [2:54:36]
No passwd entry for user 'jack'
labex:~/ $ su -l labextest                                           [2:54:41]
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

labextest@91fabe4d1ec1:~$ sudo touch /opt/forlabex
[sudo] password for labextest:
labextest@91fabe4d1ec1:~$ sudo chown labex /opt/forlabex
labextest@91fabe4d1ec1:~$ ll /opt/forlabex
-rw-r--r-- 1 labex root 0 Jul 17 02:55 /opt/forlabex
labextest@91fabe4d1ec1:~$
```

File and Directory (Linux basic concepts - directory structure, abs path, relative path: cut, copy, rename, edit, file operations)
Directory Structure - Linux
Different implementation mechanism between Windows and Linux
Windows use disk symbol (C drive, D drive) for file management, another files can also be stored in any directory, after some time disk files are messy
UNIX - directory bases, Linux as well - tree like directory structure to build entire system
All stored on disk - Linux disk is mounted on a directory, every directory can use a file system on the network. Use NFS (Network File System) server to load specific directory

FHS Std - Normal operation of sys is based on directory structure, FHS defines use of every system area, minimal required files set, directories, provides exception handling and contradiction handling
2 Specifications - 1.Various directories below / contain specific files like setting files (/etc), executable files (/bin,/sbin) 2. Explicit definitions of subdirectories of 2 directories /usr, /var like /var/log to place system log files and /usr/share for placing shared data
$tree - view entire directory structure, command not found: sudo apt-get update -> sudo apt-get install tree

Directory Path
Path = route of where you go, use cd (switch directories), . (current directory), .. (upper directory/parent directory), ls-a (view hidden files), - (last used directory), ~ (current user home directory), pwd (get current path)
Absolute path - contains root directory, subdirectories (file/directory is contained)
/usr/local/bin ('bin' directory in local directory in the usr directory of root (/) directory)
Relative path - path relative to present working directory (pwd)
If in /var/log, desire change to /var/log/kernel - use relative path concept to change directory to kernel
Change directory to /usr/local/bin by using relative path concept: cd ~, cd ../../usr/local/bin
Using abs. path / relative path go to another directory (tab - auto fills path, avoid input errors, tab twice to show all results)

Basic Linux File Operations
Create new file - touch, change timestamp of existing file, w/o parameters, file name -> can create an empty file
Used (cd ~) to switch back to user's home/labex directory
~ cd -> touch test
Create new directory: mkdir mydir (create empty directory titled 'mydir')
Using mkdir w/ -p parameter (create parent directory if it does not exist)
Creating multi-level directories (mkdir -p father/son/grandson)

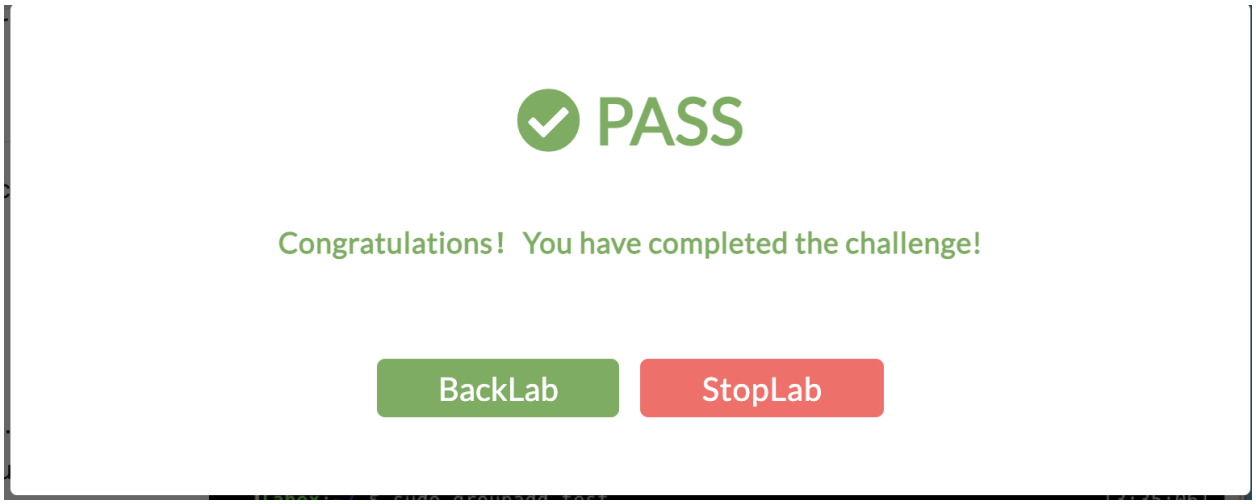
Copy
Copy Files - use cp to replicate file to specified directory/ copy prev file: cp test father/son/grandson
Copy Directories - cp w/ -r, -R parameter, (recursive copy) cp -r father family - avoids error that occurs when directly using cp to copy a directory
(file deletion) rm test
(Force deletion of files w/ read-only permissions) rm -f test
Delete Directories - rm -r family
Move Files - mv (move file to another place) (move 'file1' to directory named 'Documents') - mkdir Documents -> mv file1 Documents
-Rename Files- mv file1 myfile
Rename Multiple Files: touch file{1..5}.txt -> rename 's/\.txt/\.c/' *.txt -> rename 'y/a-z/A-Z/'*.c

3.5 - View Files
Commands used to print file content to std output (terminal), cat - prints contents in forward order, tac - prints contents in reverse order
Execute shell command, system open tree std files: std input file (stdin), std output file (stdout), standard error output file (stderr). Process gains input from std input file, output data to std output file. Any error - process sends error msg to std error file
View file passwd copied from /etc: cd /etc/ -> cat passwd (shows contents of passwd) and cat -n passwd (shows line numbers and contents in passwd file) (nl - add line number, print)

More and less - more (simpler vs less), rolls in one direction. Less - similar to more but allows backward movement in file + forward movement, does not have to read entire input file prior to starting. Less uses termcap, runs on a variety of terminals + limited support for hardcopy terminals. Head and tail allows for viewing of file -

---

Add New User and Group Challenge
labex:~/ $ sudo adduser jack
labex:~/ $ usermod -d home/jack jack
labex:~/ $ usermod --shell /bin/zsh jack
labex:~/ $ sudo usermod -G labex jack
labex:~/ $ sudo groupadd dev
labex:~/ $ sudo usermod -aG dev jack
labex:~/ $ sudo adduser bob
labex:~/ $ sudo usermod --shell/bin/bash bob
labex:~/ $ sudo usermod -G labex bob
labex:~/ $ sudo groupadd test
labex:~/ $ sudo usermod -aG test bob



Find a File Challenge

Backup System Log Challenge
Analyze Historical Commands Challenge

Data Extraction Challenge
Send regex

Download it as pdf, go to repo (forked) and upload pdf file