Linux mastery

Wednesday, April 13, 2022

12:58 PM

Section 1 : Welcome to the course

Section 2: Setting up your Linux machine

Section 3: Mastering the linux terminal

Section 3 - Duration : 2 hr 36 mins

1.How to open and close the terminal

* Keyboard shortcut : ctrl+alt+T - open
* Keyboard shortcut : ctrl+T - close

2.Our first commands

* echo - printing words
* Cal - calendar
* Up arrow - last command
* History - Shows history
* Exit - Exit terminal

3.Terminals, commands and shells

* Commands are basic text which gets interpreted by shell.
* Terminal is the mediator in between command and shell.
* The terminal is nothing more than just a window into the shell
* Most common type is bash shell
* Shell is a program to interpret command.
* Commands are case-sensitive.

4.Understanding command structure

* Structure: <CommandName> <options> <inputs>.
* Shortform options use '-'.
* Longform options use '--'.

5.Using Linux Manuals Part 1

* Structure

|  |  |
| --- | --- |
| Section | contents |
| 1 | User commands |
| 2 | System calls |
| 3 | C Library functions |
| 4 | Devices and special files |
| 5 | File formats and Conventions |
| 6 | Games |
| 7 | Miscellaneous |
| 8 | System administration |

* System administrator is used by only administrator.

6.Using Linux Manual Part 2

* Command for manual : 'man' .
* '-k' option is used for manual searching of commands.
* Man <Section number> <Command>.
* If no section number is provided, it will search in section 1.
* 'q' to exit the manual.
* Synopsis shows how to use the commands.
* For ex:- $ man -k which

7.Using Linux Manual Part 3

* If the instructions is not present in manual we use 'help' command to find instructions.

8.Command Input and Output

* Standard data stream: Standard input, Standard Output, Standard error
* Standard error : where error and log messages go.
* Standard I/P : defaultly connected to keyboard.
* Standard O/P : defaultly connected to Terminal.
* Standard error : defaultly connected to Terminal.

9.Redirection Part 1

* Cat- print content of file.
* cat > output.txt - Type content and appends in the file output.txt.
* ctrl+c - to get back to command line exit the cat command.
* ">" - overwrites the file.
* ">>" - appends in the present file.

10.Piping

* A pipe is a form of redirection that is used to send the output of one command or program or process to another for further processing.
* " | " pipe in linux.
* Example : date | cut --delimiter " " --fields 1.

11.Tee command

* tee command reads standard input (stdin) and writes it to both standard output (stdout) and one or more files.
* It basically breaks the output of a program so that it can be both displayed and saved in a file.
* It does both the tasks simultaneously, copies the result into the specified files or variables and also display the result.
* Example : date | tee fulldate.txt | cut --delimiter = " " --field = 1
* the above command will display the output and save the file in home directory as fulldate.txt.
* Example : date | tee fulldate.txt | cut --delimiter = " " --field = 1 > today.txt
* the above command will create and save both fulldate.txt and today.txt file.

12.Xargs command

* It is used to build and execute commands.
* $ command -options arguments
* Example: date | xargs echo (or) date | xargs echo "hello"
* some commands only accept command line arguments in that case xargs can be used.

13.Aliases

* Aliases are custom made command name used for a user defined function.
* .bash\_aliases - Create this file to create custom commands
* Example - alias <command name>=' <commands>…<commands>'

Section 4 - Duration 5 hr

1.The structure of linux file system

* It follows a tree structure.
* "/"- Root directory.
* Root user - Adminstrator or a super user who has maximum control over the system.
* "/root" - Home directory for the root user.

2.Navigating the file system Part 1

* "pwd" - print working directory.
* "ls" - list the contents of the directory.

3.Navigating the file system Part 2

* "cd" - change directory.
* "cd .." - change to previous directory.

4.Navigating the file system Part 3

* "Tab key" - It is used for autocompletion of commands.

5.File extensions in linux

* Changing file extension name doesn't affect the actual file extension.
* In linux file extension are considered only from header of file.
* "file <filename>" - To know the file type.

6.Wildcards

* Used to build regular expression.
* "\*" - Every expression included.
* Example: ls D\* - prints list of contents in directory starting with letter D.
* Example: ls [0-9].txt - prints list of contents in directory with 0-9 as filename.
* Example: ls [0-9]?.txt - prints list of contents in directory with 0-9 as filename with extra letter or number .

7.Creating files and folders Part-1

* Touch <filename> - to create file.
* mkdir - to make directory.
* mkdir -p - to create sub-directories in one time.
* To make file name with space use " ".
* Filename's are case-sensitive.

8.Creating files and folders Part 2

* { } - brace expansion.
* To create multiple directory we use { }.
* Example: mkdir {jan,feb}\_{1..5}.

9.Deleting files and folders Part-1

* "rm" - To remove file or folder.

10.Deleting files and folders Part-2

* Options

|  |  |
| --- | --- |
| -f | Force all |
| -i | Prompts before deleting |
| -r | Delete directory with subcontents |
| -d | Delete empty directory |
| -v | Verbose output |
| -rmdir | to delete empty directory |

11.Copy files and folders

* "cp" - Copy file.
* "cp - r" - Copy folder.

12.Moving + renaming files

* "mv" - to move file from source to destination.
* Usage: mv <source> <destination> - renaming file name.
* Usage: mv <source path> <destination path> - move folder.

13.Editing files using Nano editor Part 1 & 2

* Nano is a command line based text editor.
* To use internal editor tools use "ctrl+" key.
* Nano's configuration file is stored in "/etc/nanorc".

14.The locate command Part 1

* " locate" command is used to locate the file and returns the path.
* Example: locate <filename>

|  |  |
| --- | --- |
| -i | Irrespective of case sensitive |
| -e | Existing in DB |
| -c | Count the entries |
| -n | Shows n number of entries |
| -S | Statistics of Database |

15.The locate command Part 2

* "updatedb" command is used to update locate database with latest directories and files.
* updatedb is a sudo user command and can only be executed by administrator.

16.The find command Part 1

* "find" command is used to search any file or folder present in the system.
* It doesn't require updation of the database like "locate" command.
* Usage: find <file/folder>
* Example: find /home/akhil/Documents/

It will print all the files and folders present in the path.

* Find will defaultly start at the current folder if no argument is passed.

17.The find command Part 2

* Options:

|  |  |
| --- | --- |
| -maxdepth <input> | Number of sub-directories it needs to find under the parent directory. |
| -type <options> | f : for files, d : directory |
| | wc -l | To count the list |

18.The find command Part 3

* Options:

|  |  |
| --- | --- |
| -name "<input>" | To search by name and regex can be used in input field |
| -i | Case in sensitive |
| -size <input> | + for more than and - for less than Ex: +100k files more than 100kb, more -size can be used |

19.The find command Part 4

|  |  |
| --- | --- |
| -exec | Execute a command after the find command |
|  | cp - to copy files to destination location  Mv - to move files to destination |
|  | Example: find / -size -50k -exec cp {} ~/Desktop/here \; |
| -ok | Same as exec but will prompt before execution |

20.The find command Part 5

* "touch ~/super\_secret\_stuff/Folder$(shuf -i 1-200 -n 1)/findme.txt" - To create file randomly in any folder.
* Execution of all the commands taught in part 1 - 4 is done here.

21.Viewing files Part 1 & 2

* "cat" command is used for viewing.
* Using wildcards cat can be used.
* To reverse a file use "tac" command. (vertically).
* To reverse a file use "rev" command (horizontally).
* Use "less" command to go through big data in simple form.
* Use "head" command to print data from start of the file.
* Use "tail" command to print data from the bottom of the file.

Example: tail -500 <filename> -- prints 500 lines from bottom of file.

22.Sorting files

* "sort" command is used to sort alphabetically from A to Z.
* To sort numerically use the -n option.
* To sort month data use the -M command.

23.Searching file content

* "grep" command is used to search through files.

|  |  |
| --- | --- |
| -V | Invert search |
| -F | It gives slash around directories |
| -c | Returns count |
| -I | Case-insensitive |

24.File archiving and compression

* "tar" command stands for tape archive.
* Usage: tar [options] [archivefile] [file or directory to be archieved]

|  |  |
| --- | --- |
| -c | Creates archives |
| -x | Extract the archives |
| -f | Creates archive with given name |
| -v | Displays verbose information |

Mastering task automation and scheduling - Duration : 53 min

1.Creating bash scripts Part 1

* ".sh" is extension for bash shell file.
* Head of the file must contain "#!/bin/bash".
* Type in the commands after 1 line and save it.
* To run use "bash" command and filename.

2.Creating bash scripts Part 2

* Error can be sent to a bit bucket. NULL location - " /dev/null"
* Example : to create backup file.

Tar -czf backup.tar.gz ~/{Documents,Desktop,…} 2>/dev/null

* "chmod" command is used for modifying permissions.
* "chmod +x" command makes a file executable.
* To make executable file run from anywhere, include it in PATH.
* Include this line in last of .bashrc file "PATH="$PATH:$HOME/bin"

3.Scheduled automation using cron Part 1

* Use "crontab -e" command to start timed command automation.
* Add the time for the command to be executed.
* Format: minute hour date\_of\_month month day\_of\_week command.
* Use "\*" in fields for the command to run every minute/hour/date/month/day.
* Example: 1 \* \* \* \* echo "Hello world" >> ~/Desktop/Hello.txt

It will append hello world in hello.txt file every minute.

4.Scheduled automation using cron Part 2

* Change default editor using ".selected\_editor" file or type "select-editor" command and select the number.
* We can use multiple input in one field seperated by ","
* "\*/15" input in field will automate in every 15 minutes or so.
* Crontab.guru is a useful website to know about crontab.
* Use bash file as command in crontab

Usage: in command part : bash <Path>

Mastering open source software Duration- 1hr 38min

* Freedom to run program
* Freedom to study how the program works, and change it.
* The freedom to redistribute
* The freedom to distribute copies
* "uname - o" command gives information regarding OS.

2.Compiling software from source code

* Download a program from GNU.org and extract it.
* "make" command to configure the installation file according to the system.
* "sudo apt-get install <name>" command to install the name from repository.
* "bash configure" will create a make file to compile.
* "sudo apt-get install make" to compile and install the program.
* "sudo make install" to install the make file.
* Four types of repositories
  + Main - Canonical-supported free and open-source software.
  + Universe - Community-maintained free and open-source software.
  + Restricted - Proprietary drivers for devices.
  + Multiverse - Software restricted by copyright or legal issues.
* Package list available at packages.ubuntu.com
* "lsb\_release -a" used to see the description of distribution. After that we can install software under the codename.
* "uname -m" command gives architecture of system.
* Package manager for ubuntu is apt (advanced packaging tool).
* To search for packages use "apt-cahe search <search-key>" command.
* Use grep to filter content.

Example: apt-cache search docx | grep text.

* "apt-cache show <package> | less" command will show details about the package in readable format.
* Path of package - "/var/lib/apt/lists".

5.Updating the cache and upgrading software

* "sudo apt-get update" command to update the package manager.
* "sudo apt-get upgrade" command to upgrade the the package manager.
* "sudo apt-get install <package-name>" command to install the package.
* "xman" command is used to open GUI of manual page.
* Uncomment all the codes with "deb" in /etc/apt/sources.list to download source code.
* To install source code we need to have "dpkg-dev" package installed.
* "sudo apt-get source <package-name>" command to download the source code of package.
* "sudo apt-get remove <package name>" command to remove the package but will leave behind the config file.
* "sudo apt-get purge <package name>" command to remove the package with all the config file with it.
* "sudo apt-get autoremove" command will automatically remove the files which are no longer required.
* "sudo apt-get clean" to clean the archives "/var/cache/apt/archives".
* "sudo apt-get autoclean" to clean the archives that are no longer accessible "/var/cache/apt/archives".

Setting up Linux virtual machine

* download oracle virtualBox for windows.
* install virtualbox on your pc.
* download ubuntu a linux distro from ubuntu site.
* file will be save in downloads as disk image file eg: ubuntu.img.
* open virtualbox and select new option and configure the virtual machine.
* create a virtual hard disk in vdi(virtualbox disk image).
* now open the settings and select ubuntu.img for controller:IDE field.
* then click on start to launch ubuntu.

2.Setting up linux virtual machine

* after the successfull booting of ubuntu select install ubuntu.
* then select "Erase disk and install Ubuntu";
* select location ,keyboard and then move on with username and password.
* now click on "continue" will start installing it will take 10-20 minutes.
* after installation is completed click on restart , now ubuntu is ready to use.