**Administering Open Source Systems Lab**

**By Udit Krishna Chaudhary**

**500062348**

**97**

**EXPERIMENT 1**

* **mkdir**
  + Create the DIRECTORY(ies), if they do not already exist.
  + **mkdir** [*OPTION*]... *DIRECTORY*...
* **cat**
  + Concatenate FILE(s), or standard input, to standard output.
  + **cat** [*OPTION*]... [*FILE*]...
* **tree**
  + *Tree* is a recursive directory listing program that produces a depth indented listing of files. With no arguments, *tree* lists the files in the current directory. When directory arguments are given, *tree* lists all the files and/or directories found in the given directories each in turn. Upon completion of listing all files/directories found, *tree* returns the total number of files and/or directories listed.
  + **tree** [*directory* ...]
* **ls**
  + List information about the FILEs (the current directory by default). Sort entries alphabetically if none of **-cftuvSUX** nor **--sort**.
  + **ls** [*OPTION*]... [*FILE*]...
* **cd**
  + Change the current working directory to the one given in argument
  + **cd** [DIRECTORY]
* **cal**
  + **Cal** displays a simple calendar. If arguments are not specified, the current month is displayed.
  + **cal** [**-smjy13**] [[[*day*] *month*] *year*]
* **date**
  + Display the current time in the given FORMAT, or set the system date.
  + **date** [*OPTION*]... [*+FORMAT*]
* **clear**
  + **clear** clears your screen if this is possible. It looks in the environment for the terminal type and then in the **terminfo** database to figure out how to clear the screen.
  + **clear**
* **who**
  + Print information about users who are currently logged in.
  + **who** [*OPTION*]... [ *FILE | ARG1 ARG2* ]
* **man**
  + Displays user manual pages for the specified arguments
  + **man**
* **uname**
  + Print certain system information. With no OPTION, same as **-s**.
  + **uname** [*OPTION*]...
* **history**
  + The *history* command can be used to list Bash's log of the commands you have typed

**EXPERIMENT 2**

* **head**
  + Print the first 10 lines of each FILE to standard output. With more than one FILE, precede each with a header giving the file name. With no FILE, or when FILE is -, read standard input.
  + **head** [*OPTION*]... [*FILE*]...
* **wc**
  + Print newline, word, and byte counts for each FILE, and a total line if more than one FILE is specified. With no FILE, or when FILE is -, read standard input.
  + **wc** [*OPTION*]... [*FILE*]...
* **tail**
  + Print the last 10 lines of each FILE to standard output. With more than one FILE, precede each with a header giving the file name. With no FILE, or when FILE is -, read standard input.
  + **tail** [*OPTION*]... [*FILE*]...
* **cut**
  + Print selected parts of lines from each FILE to standard output.
  + **cut** *OPTION*... [*FILE*]...
* **sort**
  + Write sorted concatenation of all **FILE**(s) to standard output.
  + **sort** [*OPTION*]... [*FILE*]...
* **grep**
  + **grep** searches the named input *FILE*s (or standard input if no files are named, or if a single hyphen-minus (**-**) is given as file name) for lines containing a match to the given *PATTERN*. By default, **grep** prints the matching lines.
  + **grep** [*OPTIONS*] *PATTERN* [*FILE*...]
* **tr**
  + Translate, squeeze, and/or delete characters from standard input, writing to standard output.
  + **tr** [*OPTION*]... *SET1* [*SET2*]

**EXPERIMENT 3**

* **ln**
  + In the 1st form, create a link to TARGET with the name LINK\_NAME. In the 2nd form, create a link to TARGET in the current directory. In the 3rd and 4th forms, create links to each TARGET in DIRECTORY. Create hard links by default, symbolic links with **--symbolic**. When creating hard links, each TARGET must exist. Symbolic links can hold arbitrary text; if later resolved, a relative link is interpreted in relation to its parent directory.
  + **ln** [*OPTION*]... [*-T*] *TARGET LINK\_NAME (1st form)*
  + **ln** [*OPTION*]... *TARGET (2nd form)*
  + **ln** [*OPTION*]... *TARGET*... *DIRECTORY (3rd form)*
  + **ln** [*OPTION*]... *-t DIRECTORY TARGET*... *(4th form)*
* **rm**
  + **rm** removes each specified file. By default, it does not remove directories.
  + **rm** [*OPTION*]... *FILE*...
* **chmod**
  + **chmod** changes the file mode bits of each given file according to *mode*, which can be either a symbolic representation of changes to make, or an octal number representing the bit pattern for the new mode bits.
  + **hmod** [*OPTION*]... *MODE*[*,MODE*]... *FILE*...
  + **chmod** [*OPTION*]... *OCTAL-MODE FILE*...
* **users**
  + Output who is currently logged in according to FILE. If FILE is not specified, use /var/run/utmp. /var/log/wtmp as FILE is common.
  + **users** [*OPTION*]... [*FILE*]
* **groups**
  + Print group memberships for each USERNAME or, if no USERNAME is specified, for the current process (which may differ if the groups database has changed).
  + **groups** [*OPTION*]... [*USERNAME*]...
* **usermod**
  + The **usermod** command modifies the system account files to reflect the changes that are specified on the command line.
  + **usermod** [*options*] *LOGIN*
* **chown**
  + **chown** changes the user and/or group ownership of each given file. If only an owner (a user name or numeric user ID) is given, that user is made the owner of each given file, and the files' group is not changed. If the owner is followed by a colon and a group name (or numeric group ID), with no spaces between them, the group ownership of the files is changed as well.
  + **chown** [*OPTION*]... [*OWNER*][*:*[*GROUP*]] *FILE*...

**EXPERIMENT 4**

* **addgroup**
  + The **addgroup** command creates a new group account using the values specified on the command line plus the default values from the system. The new group will be entered into the system files as needed.
  + **addgroup** [*options*] *group*
* **id**
  + Print user and group information for the specified USERNAME, or (when USERNAME omitted) for the current user.
  + **id** [*OPTION*]... [*USERNAME*]
* **adduser**
  + When invoked without the **-D** option, the **adduser** command creates a new user account using the values specified on the command line plus the default values from the system. Depending on command line options, the **adduser** command will update system files and may also create the new user's home directory and copy initial files.
  + **adduser** [*options*] *LOGIN*
* **passwd**
  + passwd is used for setting password of an already existing user
  + passwd [USERNAME]
* **deluser**
  + The **deluser** command modifies the system account files, deleting all entries that refer to the user name *LOGIN*. The named user must exist.
  + **deluser**[options] *LOGIN*

**EXPERIMENT 5**

**Q) Menu based calculator (Addition, Subtraction, Multiplication and Division)**

**Q) Check whether the given number is odd or even**

**EXPERIMENT 6**

* **ps**
  + **ps** displays information about a selection of the active processes. If you want a repetitive update of the selection and the displayed information, use**top** instead.
  + ps [options]
* **tar**
  + GNU 'tar' saves many files together into a single tape or disk archive, and can restore individual files from the archive.
  + **tar** [*OPTION*...] [*FILE*]...

**EXPERIMENT 7**

**Q)Configure IP address for your system as 192.168.0.x net mask 255.255.255.0 and gateway as 192.168.0.1.**

**Ans)** Steps:

1. ifconfig wlo1 192.168.0.5 netmask 255.255.255.0 up
2. route add default gw 192.168.0.1

**Q) Change DNS server name to 192.168.0.254**

**Ans)** Steps:

1. sudo vi /etc/resolv.conf

Modify or enter nameserver as follows:

nameserver 192.168.0.254

nameserver 192.168.0.254

**Q)Check the ip address of your system and check connectivity with the gateway server.**

**Ans)** Steps:

1. Ifconfig
2. ping 192.168.0.254

**Q)How do you monitor network traffic in linux environment to capture packet information in the network.**

**Ans)** Using ‘netstat’ command

**Q) Command used to track only SMTP traffic on the interface eth0**

**Ans)** netstat | grep smtp

**Q) Command used to check the name to ip resolution and vice versa**

**Ans)** nslookup

**Q) How to check the number of hops between two hosts**

**Ans)** mtr [IP ADDRESS OF OTHER HOST]

* Command: ifconfig
  + **Ifconfig** is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed.
  + **ifconfig [interface]**
* Command: route
  + **Route** manipulates the kernel's IP routing tables. Its primary use is to set up static routes to specific hosts or networks via an interface after it has been configured with the ifconfig program.
* Command: ip
  + show / manipulate routing, devices, policy routing and tunnels
  + **ip** [ *OPTIONS* ] *OBJECT* { *COMMAND* | **help** }
  + *OBJECT* := { **link** | **addr** | **addrlabel** | **route** | **rule** | **neigh** | **tunnel** | **maddr** | **mroute** | **monitor** }
  + *OPTIONS* := { **-V**[*ersion*] | **-s**[*tatistics*] | **-r**[*esolve*] | **-f**[*amily*] { **inet** | **inet6** | **ipx** | **dnet** | **link** } | **-o**[*neline*] }
* Command: nslookup
  + **Nslookup** is a program to query Internet domain name servers. **Nslookup** has two modes: interactive and non-interactive. Interactive mode allows the user to query name servers for information about various hosts and domains or to print a list of hosts in a domain. Non-interactive mode is used to print just the name and requested information for a host or domain.
  + **nslookup** [**-option**] [name | -] [server]
* Command: mtr
  + **mtr** combines the functionality of the **traceroute** and **ping** programs in a single network diagnostic tool. As **mtr** starts, it investigates the network connection between the host **mtr** runs on and **HOSTNAME**. by sending packets with purposly low TTLs. It continues to send packets with low TTL, noting the response time of the intervening routers. This allows **mtr** to print the response percentage and response times of the internet route to **HOSTNAME**. A sudden increase in packetloss or response time is often an indication of a bad (or simply overloaded) link.
  + **mtr** [**-hvrctglspniu46**] [**--help**] [**--version**] [**--report**] [**--report-wide**] [**--report-cycles COUNT**] [**--curses**] [**--split**] [**--raw**] [**--no-dns**] [**--gtk**] [**--address IP.ADD.RE.SS**] [**--interval SECONDS**] [**--psize BYTES | -s BYTES**] **HOSTNAME [PACKETSIZE]**
* Command: netstat
  + **Netstat** prints information about the Linux networking subsystem. The type of information printed is controlled by the first argument

**EXPERIMENT 8**

**Q)How to recover when the root password is forgotten.**

**Ans)** Steps:

1. First and foremost, to recover a lost root password, we need to restart the Linux host, assuming you can’t remember the password for root (or superuser).
2. Once the GRUB page appears, quickly select the “\*Advanced options for GNU/Linux” option by pressing the down arrow key and Enter button.
3. Now press e to edit the commands.You need to modify it or change it from “read-only” mode to “read-write” mode. Find the line beginning with “Linux.” After, look for “ro,” and change it “rw.” Add init=/bin/bash at the end of the line.
4. Press F10. This will display a screen with a prompt.
5. Mount your root filesystem in read-write mode: **mount -n -o remount,rw /**
6. You can now reset your lost root password by using the following command: **passwd root**

**Q)Error in fstab file, how do fix**

**Ans)**  Steps:

1. Enter grub in rescue mode
2. Open fstab file to edit:  **vi /etc/fstab**
3. Edit the file as necessary