**Experiment 4**

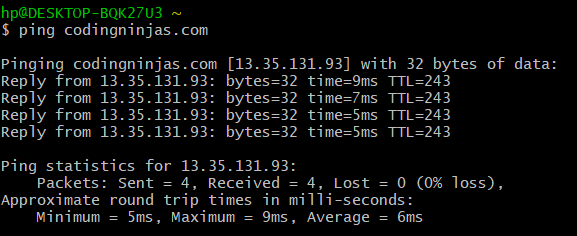
**Date:** 14 – 01 - 2021

**Aim:** To Explore More Advanced Linux Commands.

**Theory:**

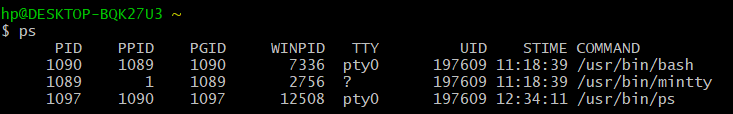
1. **ping:** The ping command lets you verify that you have network connectivity with another network device. It is commonly used to help troubleshoot networking issues. To use ping, provide the IP address or machine name of the other device.

**Syntax:** ping



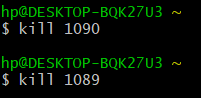
1. **ps:** The ps command lists running processes. Using ps without any options causes it to list the processes running in the current shell.

**Syntax:** ps



1. **kill:**

* To terminate a process use “kill”
* Rules are simple:
  + - * + You can kill all your own process.
        + Only root user can kill system level process.
        + Only root user can kill process started by other users.
* used to terminate processes manually. *kill* command sends a signal to a process which terminates the process. If the user doesn’t specify any signal which is to be sent along with kill command then default *TERM*signal is sent that terminates the process.
* **Syntax:** kill <pid>

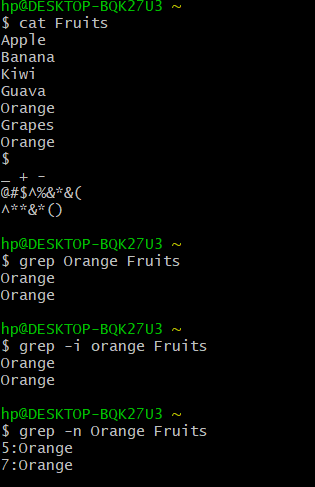


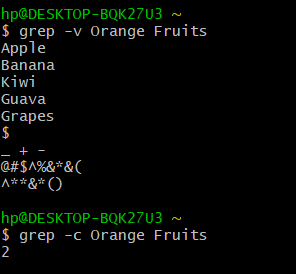
1. **grep:** It is used to grep searches the named input files for lines containing a match to the given pattern.

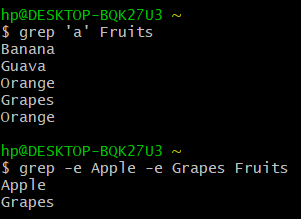
**Syntax:** grep <options> <filename>

**Note:** Options used in grep command are:

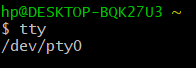
* + - -e: pattern
    - -i: Ignore uppercase vs. lowercase.
    - -v: Invert match.
    - -c: Output count of matching lines only.
    - -l: Output matching files only.
    - -n: Precede each matching line with a line number.
    - -b: A historical curiosity: precede each matching line with a block number.
    - -h: Output matching lines without preceding them by file names.
    - -s: Suppress error messages about nonexistent or unreadable files.
    - -x
    - -f file: Take regexes from a file.
    - -o: Output the matched parts of a matching line.





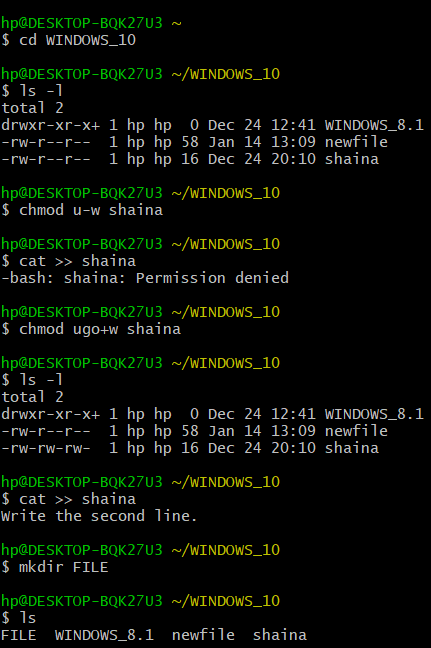


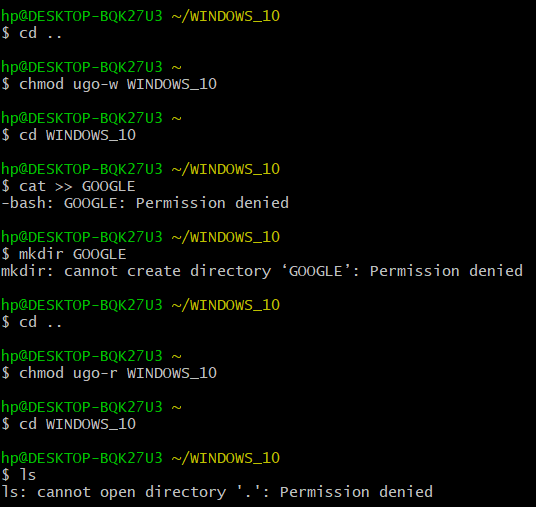
1. **tty:**
   * + tty is a command in Unix and Unix-like operating systems to print the file name of the terminal connected to standard input. tty stands for TeleTYpewriter.
     + The tty command basically prints the file name of the terminal connected to standard input.
     + **Syntax:** tty

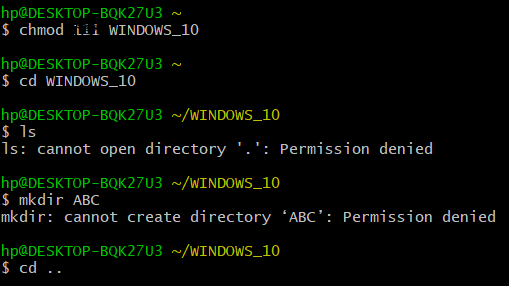


1. **chmod:**

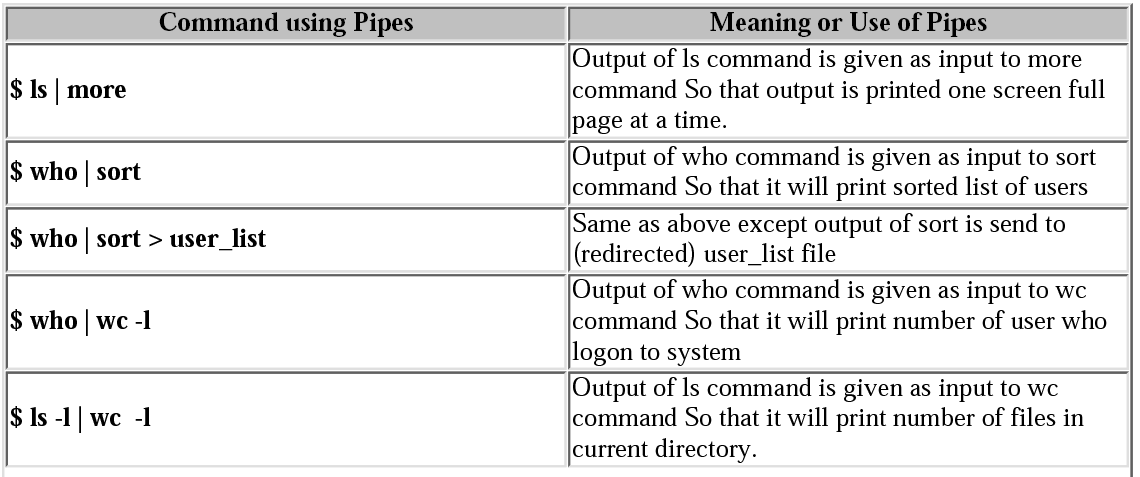
* There are three types of permissions: read (r), write (w), and execute (x).
* To read a file is to view its contents. For example, a text file must have read permission for someone to read the text within.
* If the user wants to add a sentence to that file, it needs write permission.
* The execute permission enables someone to run a file, such as a shell script or a binary program file.
* The ls -l command displays the permissions assigned to a file.
* *user*, *group*, and *other*.
* Each file is associated with an owner and a group and assigned with permission access rights for three different classes of users:
* The file owner.
* The group members.
* Others (everybody else)
* **Operation:**
  + - * - Removes the specified permissions.
      * + Adds specified permissions.
      * = Changes the current permissions to the specified permissions. If no permissions are specified after the = symbol, all permissions from the specified user class are removed.
* Directories are special types of files that contain other files and directories.
* The chmod command allows you to change the permissions on a file using either a symbolic or numeric mode or a reference file.
* **Symbolic (Text) Method:**
  + - * **Syntax:** chmod <options> <ugoa…><-+=perm s> <file>
      * The permissions (perms...) can be explicitly set using either zero or one or more of the following letters: r, w, x. Use a single letter from the set u, g, and o when copying permissions from one to another user class.
* **Numeric Method:**
  + - * The syntax of the chmod command when using numeric method has the following format.
      * **Syntax:** chmod <options> <Number File...>
      * When using the numeric mode, you can set the permissions for all three user classes (owner, group, and all others) at the same time. the first digit represents the permissions of the file’s owner, the second one the file’s group, and the last one all other users.
      * Each write, read, and execute permissions have the following number value:
        + r (read) = 4
        + w (write) = 2
        + x (execute) = 1
        + no permissions = 0
      * The permissions number of a specific user class is represented by the sum of the values of the permissions for that group.
      * calculate the totals for all users classes. For example, to give read, write and execute permission to the file’s owner, read and execute permissions to the file’s group and only read permissions to all other users you would do the following:
        + Owner: rwx=4+2+1=7
        + Group: r-x=4+0+1=5
        + Others: r-x=4+0+0=4
      * Using the method above we come up to the number 754, which represents the desired permissions.

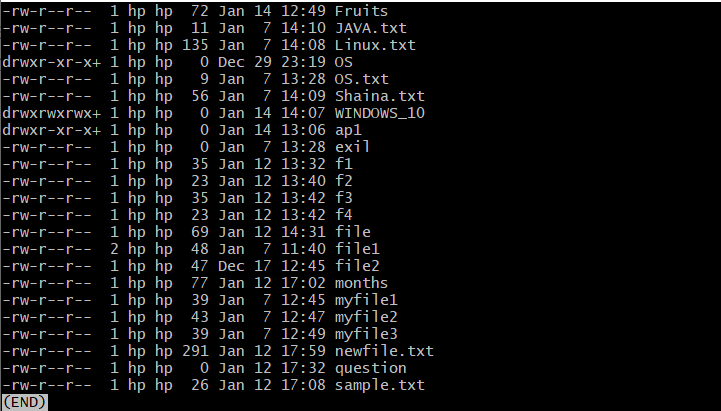
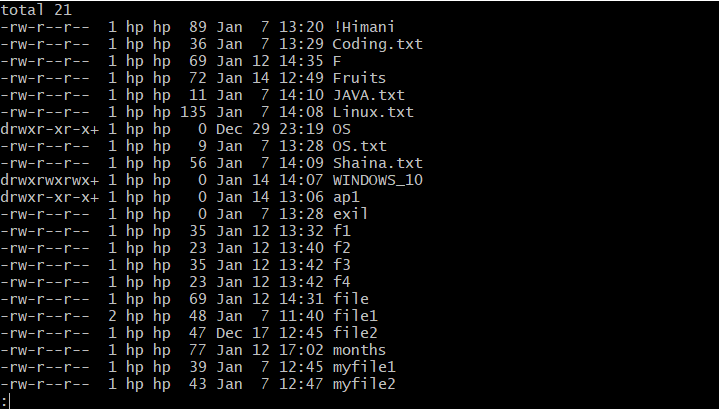
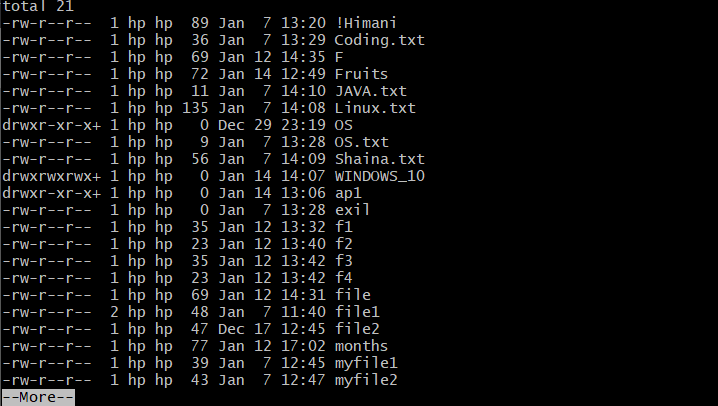






1. **Pipe ( | ):**
   * + A pipe is a form of redirection (transfer of standard output to some other destination) that is used in Linux and other Unix-like operating systems to send the output of one command/program/process to another command/program/process for further processing.
     + You can make it do so by using the pipe character ‘|’.
     + It can also be visualized as a temporary connection between two or more commands/ programs/ processes.
     + Pipes are unidirectional i.e. data flows from left to right through the pipeline.
     + The pipe acts as a container which takes the output of ls -l and gives it to more as input. This command does not use a disk to connect standard output of ls -l to the standard input of more because pipe is implemented in the main memory.
     + **Syntax:** command 1| command 2| command 3|……. | command n





**Note:**

1. **more:** more command is used to view the text files in the command prompt, displaying one screen at a time in case the file is large (For example log files). The more command also allows the user do scroll up and down through the page. The syntax along with options and command is as follows. Another application of more is to use it with some other command after a pipe. When the output is large, we can use more command to see output one by one.

**Syntax:** more <-options> <-num] ><+/pattern> <+linenum> <file\_name>

* + - **[-options]:** any option that you want to use in order to change the way the file is displayed. Choose any one from the followings: (-d, -l, -f, -p, -c, -s, -u)
    - **[-num]:** type the number of lines that you want to display per screen.
    - **[+/pattern]:** replace the pattern with any string that you want to find in the text file.
    - **[+linenum]:** use the line number from where you want to start displaying the text content.
    - **[file\_name]:** name of the file containing the text that you want to display on the screen.

1. **less:** Less command is linux utility which can be used to read contents of text file one page(one screen) per time. It has faster access because if file is large, it don’t access complete file, but access it page by page. For example, if it’s a large file and you are reading it using any text editor, then the complete file will be loaded to main memory, but less command don’t load entire file, but load it part by part, which makes it faster.

**Syntax:** less <filename>

Options used in less command are:

* + - -E : causes less to automatically exit the first time it reaches end of file.
    - -f : forces non-regular file to open.
    - -F : causes less to exit if entire file can be displayed on first screen
    - -g : highlight the string which was found by last search command
    - -G : suppresses all highlighting of strings found by search commands
    - -i : cause searches to ignore case
    - -n : suppresses line numbers
    - -p pattern : it tells less to start at the first occurrence of pattern in the file
    - -s : causes consecutive blank lines to be squeezed into a single blank line

**Result:** More linux commands has been explored successfully.