**Operating System 1**

**Lab 2**

**Basic Linux Command Introduction**

**Topics**

1. Explore the Linux File system structure and how devices are mounted under the parent root / node
2. Listing Essential Linux Commands
   * '**clear**'clear the screen
   * '**cd**' command
   * '**ls**' command
   * '**pwd**' command
   * '**cp**' command
   * '**mv**' command
   * ‘**rm**’ command
   * '**mkdir**' command
   * '**rmdir**' command
   * '**man**'read a manual page
   * '**date**'display the current date and time
   * '**echo**'display a message on the screen
   * '**cat**'display files
   * '**uname**'display the machine and operating system name
   * '**users**'list the current login sessions on the system
   * '**who**'who is currently logged in
   * '**more**' and '**less**'
3. Explaining how to use pipes to   
   ( | ) and redirecting (>>, >, < ) to combine more than one commands.

**Commands**

|  |  |
| --- | --- |
| clear  **Ex: $clear** | This command can be called to clear the current terminal screen and it can be redirected to clear the screen of some other terminal. |
| cd  **$cd /home** | This command changes the current directory to another one. |
| ls  **$ls**  **$ls –a ; to display all information about the listed files**  **$man ls ; to display the manual for *ls* to know another options.**  **q : quit the man command** | These programs list each given file or directory name. Directory contents are sorted alphabetically. For *ls*, files are by default listed in columns, sorted vertically, if the standard output is a terminal; otherwise, they are listed one per line. |
| pwd  **$pwd** | Display current user directory. |
| Cp  **$cp /home/myfile.txt /root/myfile.txt** | If the last argument names an existing directory, *cp* copies each other given file into a file with the same name in that directory. Otherwise, if only two files are given, it copies the first onto the second. It is an error if the last argument is not a directory and more than two files are given. By default, **it does not copy directories**. |
| Mv  **$mv /home/myfile.txt /root/myfile.txt**  **$mv /home/myfile.txt /root/myNewFile.txt ;**  **operate as renaming file** | If the last argument names an existing directory, *mv* moves each other given file into a file with the same name in that directory. Otherwise, if only two files are given, it moves the first onto the second. It is an error if the last argument is not a directory and more than two files are given. It can move only regular files across file systems. If a destination file is unwritable, the standard input is a *tty*, and the –f or *--force* option is not given, *mv* prompts the user for whether to overwrite the file. If the response does not begin with y or Y, the file is skipped. |
| Rm  **$rm \*.txt ; to delete a specific file**  **$rm –r /home/mydir ; to delete the directory and recursively delete its content** | *rm* removes each specified file. By default, **it does not remove directories**. If a file is unwritable, the standard input is a *tty*, and the -f or --force option is not given, *rm* prompts the user for whether to remove the file. If the response does not begin with y or Y, the file is skipped.  rm can be used to remove directories and its subdirectories and files recursively suing option -r |
| mkdir  **$mkdir /home/mydir** | *mkdir* creates a directory with each given name. By default, the mode of created directories is 0777 minus the bits set in the umask. |
| rmdir  **$rmdir /home/mydir** | *rmdir* removes each given **empty** directory. If any nonoption argument does not refer to an existing empty directory, it is an error. |
| date  **$date**  **$date 1220141414.55** | To display or to set the date and time of the system. The format for setting date is [*MMDDhhmm*[[*CC*]*YY*][*.ss*]] |
| echo  **$echo Ahmed ; will print Ahmed on the screen** | Prints its arguments to the screen. |
| cat  **$cat myfile.txt myfile2.txt**  **$cat myfile.txt ;to display the content of this file** | Concatenate files and print on the standard output. |
| uname  **$uname** | Gets name and information about the current kernel. |
| users  **$users** | Displays the current users that are logged on the system. |
| who  **$who**  **$who am I** | Output who is logged in on local machines. Similar to *users*. |
| more  **$more myfile.txt**  **$ls | more ;display the files in pages**  **$man sudo\_root | more** | Let us display and scroll down the output in one direction only. You can scroll page by page or line by line. |
| less  **$lessmyfile.txt;display the files in pages**  **$man sudo\_root | less** | Like more but more enhanced. It support scroll forward and backward (by arrows). |
| >  **$ls> myfiles.txt ;will overwrite the list of files into this file instead of the screen** | Redirect the output to be written to a file using the redirect > create/replace file operator.  If the file is not exist, it will be created. |
| >>  **$ls>> myfile.txt ;will append the list of files into this file instead of the screen** | Redirect the output to be written to a file using the redirect >> create/append to file operator.  If the file is not exist, it will be created. |
| <  **$cat < files.txt ;to take the filename from this file** | Redirect the input to be taken from a file. |
| |  **$ls | more ;display the files in pages** | Use pipes “ | “ to redirect the output of the previous command as in input to another command. |