Az-Ubuntu-Cli

Tuesday, October 22, 2024

23:30

* + **pwd**Print working directory, i.e., display the name of my current directory on the screen.
  + **hostname**   
    Print the name of the local host (the machine on which you are working). Use netconf (as root) to change the name of the machine.
  + **id** *username*   
    Print user id (uid) and his/her group id (gid), effective id (if different than the real id) and the supplementary groups.
  + **date**Print or change the operating system date and time. E.g., I could change the date and time to 2000-12-31 23:57 using this command:   
    date 123123572000   
    To set the hardware (BIOS) clock from the system (Linux) clock, use the command (as root) setclock
  + **who**Determine the users logged on the machine.
  + **finger**  *user\_name*   
    System info about a user. Try: finger root . displays the user's login name, real name, terminal name and write status (as a ``\*'' after the terminal name if write permission is denied), idle time, login time, office location .
  + **history** | more   
    Show the last (1000 or so) commands executed from the command line on the current account. The "| more" causes the display to stop after each screenful.
  + *any\_command* **--help** |moreDisplay a brief help on a command (works with most commands). "--help" works similar to DOS "/h" switch. The "more" pipe is needed if the output is longer than one screen.
  + **man***topic*Display the contents of the system manual pages (help) on the topic.
  + **Info** *topic*
  + information pages, which are generally more in-depth than manpages.
  + Fetch update software list by running the **sudo apt-get update** command
  + Update Ubuntu software by running the **sudo apt-get upgrade** command
  + Finally, reboot the Ubuntu box by running the **sudo reboot** command.
  + **ls -** Short listing of directory contents
  + **-a** list hidden files
  + -d list the name of the current directory
  + -F show directories with a trailing '/'
  + executable files with a trailing '\*'
  + **-g** show group ownership of file in long listing
  + **-i** print the inode number of each file
  + **-l** long listing giving details about files and directories
  + **-R** list all subdirectories encountered
  + **-t** sort by time modified instead of name
  + **cp** *file1 file2*
  + or
  + **cp** *myfile yourfile*
  + Copy the files "myfile" to the file "yourfile" in the current working directory. This command will create the file "yourfile" if it doesn't exist. It will normally overwrite it without warning if it exists.
  + **cp -i** *myfile yourfile*
  + With the "-i" option, if the file "yourfile" exists, you will be prompted before it is overwritten.
  + **mv** s*ource destination*   
    Move or rename files. The same command is used for moving and renaming files and directories. Ex: mv testdir newnamedir
  + **rm***files*Remove (delete) files. You must own the file in order to be able to remove it. On many systems, you will be asked or confirmation of deleation, if you don't want this, use the "-f" (=force) option, e.g., rm -f \*  will remove all files in my current working directory, no questions asked.
  + **mkdir** *directory*   
    Make a new directory.
  + **rmdir** *directory*   
    Remove an empty directory.
  + **rm -r** *files*   
    (recursive remove) Remove files, directories, and their subdirectories. Careful with this command as root--you can easily remove all files on the system with such a command executed on the top of your directory tree, and there is no undelete in Linux (yet). But if you really wanted to do it (reconsider), here is how (as root): rm -rf /\*
  + cat *filename*
  + View the content of a text file called "filename"
  + **find / -name *filename***Find the file called "filename" on your file system starting the search from the root directory "/". The "filename" may contain wildcards (\*,?).ex: find / -name "file.text"
  + **locate***filename*Find the file name of which contains the string "filename". Easier and faster than the previous command but depends on a database that normally rebuilds at night.
  + **adduser** *user\_name*   
    Create a new account (you must be root). E.g.,  adduser barbara  Don't forget to set up the password for the new user in the next step. The user home directory is /home/*user\_name*.
  + **useradd** *user\_name*   
    The same as the command " **adduser**  *user\_name* ".
  + **userdel** *user\_name*   
    Remove an account (you must be a root). The user's home directory and the undelivered mail must be dealt with separately (manually because you have to decide what to do with the files).
  + **groupadd** *group\_name*   
    Create a new group on your system. Non-essential but can be handy even on a home machine with a small number of users.
  + **passwd**  *user\_name* Change the password on your current account. If you are root, you can change the password for any user using:  passwd *user\_name*
  + ***File Permissions***
  + **chmod** *perm filename*
  + **chmod** command sets the permission of a file or folder. chmod command uses three digit code as an argument and the file or folder location.
  + In the example,
  + 7 – Owner(current user)
  + 5 – Group(set by owner)
  + 4 – anyone else
  + The fundamental concept:
  + Execute is 1, Write is 2 and Read is 4.
  + Sum of these basic ones makes combination of permissions:
  + 0 – no permission, this person cannot read, write or execute
  + 1 – execute only
  + 2 – write only
  + 3 – execute and write only (1 + 2)
  + 4 – read only
  + 5 – execute and read only (1 + 4)
  + 6 – write and read only (2 + 4)
  + 7 – execute, write and read (1 + 2 + 4)
  + examples:
  + **sudo useradd -m -s /bin/bash** *user\_Name*
  + explain: **-s** User's login shell (default /bin/bash)
  + **-m** Create the home directory
  + **-M** Do not create the home directory
  + 2- **sudo passwd** *user\_Name*
  + 3- unlock
  + **sudo passwd -u** *user\_Name*
  + 4- Add Full Name
  + **sudo usermod -c** **"***Full Name***"** *user\_Name*
  + 5- add group
  + **sudo groupadd** *group\_Name*
  + add user to group
  + sudo usermod -G group\_Name user\_Name
  + 6- Delete a user
  + **userdel -r** *user\_Name*
  + **id** Print user and group id's
  + .
  + **route –n**
  + Display Information of All Network Interfaces
  + **ifconfig –a**
  + How to Disable an Network Interface
  + **ifconfig eth0 down**
  + OR
  + **ifdown eth0**
  + How to Assign a IP Address to Network Interface
  + **ifconfig eth0 172.16.25.125**
  + 1- install :
  + **sudo apt-get install** *Package\_Name*
  + 2- remove :
  + **apt-get remove** *package\_Name*
  + To remove any unused packages, use the “autoremove” command, as shown in the following command.
  + **sudo apt-get autoremove**
  + You can combine the two commands for removing a program and removing dependencies that are no longer being used into one, as shown below (again, two dashes before “auto-remove”).
  + **sudo apt-get purge --auto-remove** gimp
  + If you’re short on space, you can use the “clean” command to remove downloaded archive files, as shown below.
  + **sudo apt-get clean**
  + df Show free disk space
  + mount Make a disk accessible
  + fsck Check a disk for errors
  + sync Flush disk caches
  + ls List files
  + cp Copy files
  + mv Rename files
  + rm Delete files
  + ln Link files
  + cd Change directory
  + pwd Print current directory
  + name
  + mkdir Create directory
  + rmdir Delete directory
  + ps List all processes
  + w List users’ processes
  + uptime View the system load
  + top Monitor processes
  + xload Monitor system load
  + free Display free memory
  + kill Terminate processes
  + nice Set process priorities
  + renice Change process priorities