Linux System Administration

Author Name	Michael A.
Batch	22
Branch	Online

Document History

Version	Release Date	Author	Description	Document Reviewer
22	09/2022	Michael A.	Created	Akin O.

Server Details:

Server Name	
IP	
Username	root
Password	*********
Operating System	

Introduction

Linux is an operating system that evolved from a kernel created by Linus Torvalds when he was a student at the University of Helsinki.

When Linus Torvalds was studying at the University of Helsinki, he was using a version of the UNIX operating system called 'Minix'. Linus and other users sent requests for modifications and improvements to Minix's creator, Andrew Tanenbaum, but he felt that they weren't necessary. That's when Linus decided to create his own operating system that would take into account users' comments and suggestions for improvements.

Linux Distributions

Following are the Linux distributions.

- **Redhat Linux**
- **Ubuntu.**
- **CentOS**
- **SuSe Linux**
- Debian.
- Fedora
- Gentoo
- Mandriva.
- Slackware

Download different Flavours of Linux distributions

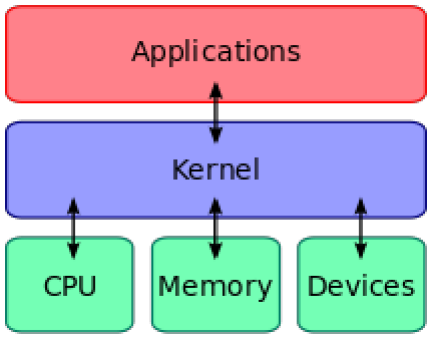
Download Linux from their respective home page. As it is an open source operating system, it’s free to download! Not all versions are free, have a look:

http://distrocenter.linux.com/
http://www.redhat.com
http://fedora.redhat.com/
http://www.debian.org
http://www.mandriva.com
http://www.suse.com
http://www.opensuse.org/
http://www.slackware.com
http://www.ubuntulinux.org/
http://www.ubuntu.com/
http://www.distrowatch.com
http://www.openbsd.org
http://www.freebsd.org
http://www.netbsd.org

© 2020 all rights Reserved by MyLandmark.tech +1 437 215 2483

Kernel and Shell

Kernel Architecture



File Directory System

Following are the directories in the Redhat OS.

1) bin: It contains the commands and binary files. User can access the commands. Ex: mkdir, ls, cd, ps….
2) boot:
3) dev: It contains the device files. Ex: USB or any device attached to the server.
4)etc: It contains the configuration files.
5)home : It contains home directories for all users to store their personal files.
Ex: /home/bhaskar or /home/mithun
6)lib : It contains the system libraries.
7) lost+found :
8) **media** :
9) **misc** :
10) mnt :
11) **net** :
12) opt :
13) proc : It contains the RAM, Process and Kernel information.
14)root :
15) sbin : Just like /bin It contains the commands and files, but only root user can access.
Ex: ifconfig, reboot, shutdown and swapon …
16) **selinux** :
17) **srv** :
18) **sys** :
19) **tmp** :
20) usr : It contains the binaries and libraries. WAS,IHS or DB2 etc related softwares will install in this directory.

21) var : It contains variable files. This includes system log files(/var/log), emails (/var/mail) and temp files needed across reboots (/var/tmp)
22) tmp : It contains the temporary files.

"/" called root directory.

/root is the home directory for the root user.

What is the difference between bin and sbin directory?
Ans)

Explain the Linux Directory/File hierarchy structure?
Ans)

Commands

mkdir : Create the DIRECTORY(ies), if they do not already exist.

Some options with mkdir are.

-m	Sets the access mode for the new directory.
-p	If the parent directories don't exist, this command creates them.
-v	Print a message for each created directory

e.g.

1) **# mkdir landm**
 # ls -lt
 # mkdir -m 777 Landmark
 # ls -lt

2) **# mkdir –p Test1/Test2/Test3 :** It will create the parent directories.
 # ls –lt

3) **# mkdir –v keren**

4) #mkdir test{1,2,3,4,5}

Note: Same like for **rmdir landmark{1,2,3,4,5}**.

5)#mkdir -m 777 Mithun : It will create the Mithun folder with 777 permissions.

---

ls : Lists the contents of your current working directory.

Some options with ls are :

-l list in long format

-a all files including those which begin with dot

-F It is used to classify the directories, executables. It will append indicator (one of */=>@|) to entries

-i list the inode number in the first column

-R recursively list all directories and subdirectories

-r Display in reverse order.

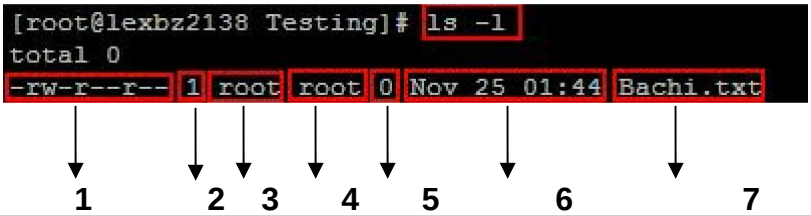
-t sorts files by access time.

-d shows the names of directories and not their contents.

-h Indicates human readable. This mentions file sizes in kilobytes, megabytes, or Gigabytes, instead of just bytes, which is the default setting. Use this optio with the
–l option only.

-S Sorts files by file size. This option is useful only when used together with the option –l

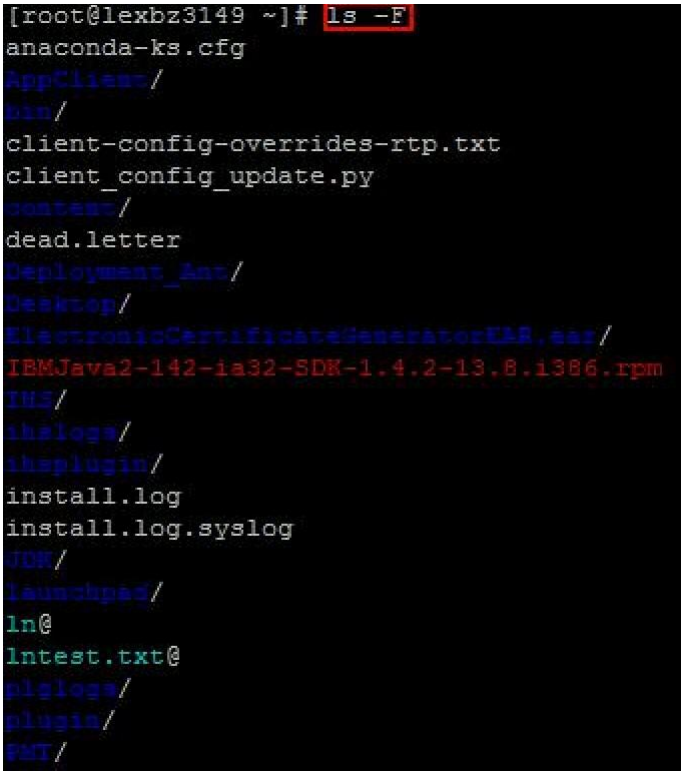
#ls –l



1) Permissions
2) Number of links
3) Owner
4) Group Owner
5) Size of the File in Bytes
6) Date and Time of created
7) File name/Dir Name

1) #ls –F

/home/Jessica -



Here / indicates the directories, * indicates the executable files, @ indicates soft link.

```
[root@lexbz3149 Testing]# ls
Bachi.txt  Bhaksar.txt  db2jcc.jar  JPA.pdf  TestDir1
[root@lexbz3149 Testing]# ls -a
.  ..  Bachi.txt  Bhaksar.txt  db2jcc.jar  JPA.pdf  TestDir1
[root@lexbz3149 Testing]# ls -l
total 1440
-rw-r--r-- 1 root root      84 Jul 16 08:14 Bachi.t
-rw-r--r-- 1 root root       0 Jul 16 08:18 Bhaksar
-rw-r--r-- 1 root root 1075286 Jan  6  2012 db2jcc.
-rw-r--r-- 1 root root  378622 Oct 25  2011 JPA.pdf
drwxrwxrwx 3 root root    4096 Jul 16 08:25 TestDir
[root@lexbz3149 Testing]# ls -R
.:
Bachi.txt  Bhaksar.txt  db2jcc.jar  JPA.pdf  TestDir1

./TestDir1:
TestDir2

./TestDir1/TestDir2:
[root@lexbz3149 Testing]# ls -t
TestDir1  Bhaksar.txt  Bachi.txt  db2jcc.jar  JPA.pdf
[root@lexbz3149 Testing]# ls -lt
total 1440
drwxrwxrwx 3 root root    4096 Jul 16 08:25 TestDir1
-rw-r--r-- 1 root root       0 Jul 16 08:18 Bhaksar.txt
-rw-r--r-- 1 root root      84 Jul 16 08:14 Bachi.txt
-rw-r--r-- 1 root root 1075286 Jan  6  2012 db2jcc.jar
-rw-r--r-- 1 root root  378622 Oct 25  2011 JPA.pdf
[root@lexbz3149 Testing]# ls -d
.
[root@lexbz3149 Testing]# ls -ld
drwxr-xr-x 3 root root 4096 Jul 16 08:25 .
[root@lexbz3149 Testing]# ls -h
Bachi.txt  Bhaksar.txt  db2jcc.jar  JPA.pdf  TestDir1
[root@lexbz3149 Testing]# ls -lh
total 1.5M
-rw-r--r-- 1 root root   84 Jul 16 08:14 Bachi.txt
-rw-r--r-- 1 root root    0 Jul 16 08:18 Bhaksar.txt
-rw-r--r-- 1 root root 1.1M Jan  6  2012 db2jcc.jar
-rw-r--r-- 1 root root 370K Oct 25  2011 JPA.pdf
drwxrwxrwx 3 root root 4.0K Jul 16 08:25 TestDir1
[root@lexbz3149 Testing]# ls -S
db2jcc.jar  JPA.pdf  TestDir1  Bachi.txt  Bhaksar.txt
```

ls -li : It will display the inode number.

What is inode number?

Ans) Inode is data structure that contain information about files that are created when a file system is created. Each file has an inode and is identified by an inode number in the file system.

When a new file is created, one inode number will be assigned to that file. The following information are stored in inode.

• Owner and group owner of the file.

• File type

• Permissions on the file.

• Date and time of creation, last read and change.

• Date and time this information has been changed in the inode.

• Number of links to this file

• File size

In linux which of the following information inode does not contain?

1. permission
2.File name
3.Owner ID
4.Size of the file.

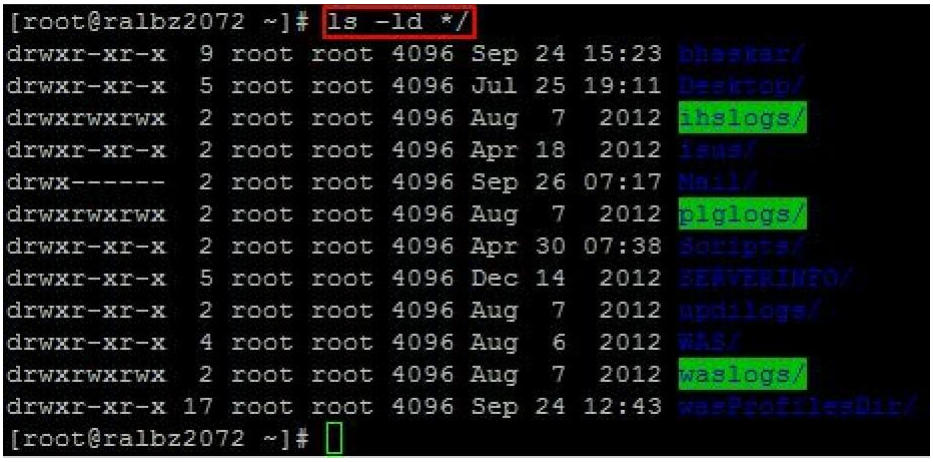
Ans) 2

#ls b* : List files starting with b.
#ls *b : List files ending with b.
#ls *b* : Files just having letter b.
#ls ? : List single lettered file
#ls ?? : List double lettered file

#ls -l | grep ^d: It will display only the subdirectories from the current directory.

```
[root@ralbz2072 ~]# ls -l | grep ^d
drwxr-xr-x 9 root root     4096 Sep 24 15:23 bhaskar
drwxr-xr-x 5 root root     4096 Jul 25 19:11 Desktop
drwxrwxrwx 2 root root     4096 Aug  7  2012 ihslogs
drwxr-xr-x 2 root root     4096 Apr 18  2012 isus
drwx------ 2 root root     4096 Sep 26 07:17 Mail
drwxrwxrwx 2 root root     4096 Aug  7  2012 plglogs
drwxr-xr-x 2 root root     4096 Apr 30 07:38 Scripts
drwxr-xr-x 5 root root     4096 Dec 14  2012 SERVERINFO
drwxr-xr-x 2 root root     4096 Aug  7  2012 updilogs
drwxr-xr-x 4 root root     4096 Aug  6  2012 WAS
drwxrwxrwx 2 root root     4096 Aug  7  2012 waslogs
[root@ralbz2072 ~]#
```

 # ls -ld */ : It will display only the subdirectories from the current directory along with soft link directories also.



#ls -l |awk '{ if (substr($1,1,1)== "d") {print}}'

#ls -p | grep "/" | xargs du –ks: It will display kilobytes used by each subdirectory of the current directory**.**

File types in a long list

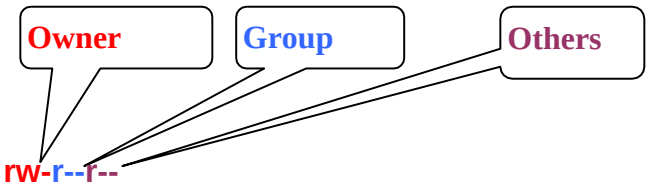
Symbol	Meaning
-	Regular file
d	Directory
l	Soft Link
C	Special file
S	Socket
B	Block device

Default Color schemas for different file types

Color	File Type
Blue	Directories
White	Text files
Pink	Images
Red	Compressed archives
Cyan	Soft Links
Green	Executables
Yellow	Devices
Flashing red	Broken links

#ls -l

Access Permissions



				r	w	x
r(Read)	w(Write)	x (Execute)	-----> Owner	4	2	1
1	1	0	----------- > Binary Notation			
4 +	2 +	0	= 6 ------> Decimal Notation			
r(Read)	w(Write)	x (Execute)	--------> Group			
1	0	0	----------- > Binary Notation			
r(Read)	w(Write)	x (Execute)	-------> Others			
1	0	0	----------- > Binary Notation			

---

cd : To Change working directory.

Syntax : cd << Directory >>
cd <<Path/ Directory>>

e.g. : # mkdir /root/Testing/TestDir1/BhaskarReddy
cd /root/Testing/TestDir1/BhaskarReddy

More about cd command
cd / takes to home directory
cd << dirname >>
cd << pathname >>
cd ..To move one level up (immediate parent directory)
cd ../.. To move two levels up (so on)
cd ~ It will takes to home directory.
cd It will takes to home directory.

cd - It will takes to previous directory.

Relative path cd local
Absoulte path cd /usr/local

What is the differance between absolute path and relative path?
Ans) An absolute path begins with the root directory.
Ex: /home/mithunreddy/devops/devops.txt
A relative path starts from the current working directory.
Ex: ./devops/devops.txt

---

pwd (Print Working Directory): To see the current working directory

e.g. : [root@mithuntechnologies TestDir1]# **pwd**

---

rmdir : Removes a directory.

Some conditions to use **rmdir** command

- **Must be empty before it is deleted**
- **Should not be the current directory or a directory at a higher level**
- **The directory name should exists.**
- **Should not be HOME directory of the user**

Options can be

- **-i interactively asks for confirming the deletion of files. It is useful in avoiding accidental erasure of the file**
- **-r option provides a convenient way to erase a directory even if it is not empty**
- **-f option will forcefully remove a file to which we don't have a write permission.**

rm: Removes (unlinks) files or directories.

Syntax: rm <<File/Directory>>

rm –rf /Testing -----> Removes directory which has contents.

rm test.txt---------> Removes the file.

rm *.txt ------> Removes all the files extension with .txt.

What is the difference between rm and rmdir?

Ans) rm removes the files and directories.
rmdir command removes the directories from the file system. The directory must be empty before it can be remove.

---

touch : Creates a new file with zero size or update the timestamp of an existing file.

#touch bhaskar.txt

We can use below command for creating a new file.

#cat > newcatfile.txt (type your text when done press Ctrl + D).

Note: If data is already exists in that file, touch command will not overwrite, just it will update the timestamp.

---

locate : Using locate command you can quickly search for the location of a specific file (or group of files .

#locate mithun.txt

The example below shows all files in the system that contains the word crontab in it.

#locate crontab

---

find : find command used to search and locate list of files and directories based on conditions you specify for files that match the arguments. Find can be used in variety of conditions like you can find files by permissions, users, groups, file type, date, size and other possible criteria.

#find ~ -name mithun.sh: It will search the mithun.sh file under logged in user home directory.

#find . -name mithun.sh: It will search the mithun.sh file under current directory.

#find . -iname mithun.sh: It will search the mithun.sh file under current directory ignoring the case.

find . -type f -perm 0777: It will search all the files which have 777 permissions.

find . -perm /a=x: It will search all the files which have execute permissions.

find . -type f -empty: It will search all the empty files.

find . -type f –name ".*": It will search all the hidden files in current directory.

find / -name <<FIleName>> : It will display the all files which we provide the name.

ex.**# find / -name apachectl :** It will display the all the locations, where the file apachectl is available.
#find / -iname apachectl : It will display the all the locations, where the file apachectl is available.
Here case insensitive.

#find . -mtime 1 : It will find all the files modified exact 1 day in current directory.

#find / -mtime 1 : It will find all the files modified exact 1 day in all directories.

#find . -mtime -1 : It will find all the files modified less than 1 day

#find . -mtime +1 : It will find all the files modified more than 1 day

#find / -mmin -10 : It will locate/display the files which modified less than 10 minutes ago.

#find / -perm 600 : It will displays the files and directories which have the 600 permissions.–perm option is used to find files based upon permissions.

#find . –name "*.java" | xargs grep “FTPReadXML” : It will displays java files which have the FTPReadXML word in those files.

 # find . -name "*.java" -exec chmod 700 '{}' \; : It will set the access permissions (700) for all java files.
The argument '{}' inserts each found file into the chmod command line. The \; argument indicates the exec command line has ended.

#find . -size +1000c -exec ls -l {} \; : It will display the files which size greater than 1000bytes.

#find /pportal/assethub/ats/logs/ -mmin +30 -exec rm -f {} \; : It will remove the logs which are created more than 30 minutes back.

#find . -name "*.*" -ls | awk '{print $7 " " $11}' | sort -rn | head -n 10 :
Find ten largest files in the current directory and recursion through all subdirectories

---

What is the difference between locate and find?
Ans) The find command has several options and is very configurable. There are many ways to reduce the depth and breadth of your search and make it more efficient.

locate uses a previously built database, If database is not updated then locate command will not show the output. to sync the database it is must to execute updatedb command.

By default, the system will run **updatedb** which takes a snapshot of the system files once a day, locate uses this snapshot to quickly report what files are where. However, recent file additions or removals (within 24 hours) are not recorded in the snapshot and are unknown to locate.

#updatedb : Only root user can able to run.

---

umask : User Mask or User file creation MASK : It is used to set the permissions for files/directories newly created ona Linux Machine.

- The default **umask 002** used for normal user. With this mask default directory permission are ***drwxrwxr-x (775)*** and default file permissions are ***-rw-rw--r-- (664)***.
- The default **umask for the root user is 022** result into default directory permissions are ***drwxr-xr-x (755)*** and default file permissions are ***-rw-r--r—(644)***.
- For directories, the **base permissions** are (rwxrwxrwx) 0777 and for files they are 0666 (rw-rw-rw).
- Umask setup is in **/etc/bashrc** or **/etc/profile** file as follows.

Below sample code from **/etc/bashrc** file.

```
# By default, we want this to get set.
# Even for non-interactive, non-login shells.
if [ $UID -gt 99 ] && [ "`id -gn`" = "`id -un`" ]; then
        umask 002
else
        umask 022
fi
```

Calculating the permissions for Files and directories are when umask value is 006.

For File:
Subtract the umask value from base permissions.
666 – 006 = 660 (rw-rw ---)
For Directory:
Subtract the umask value from base permissions.
777 – 006 = 771 (rwxrwx--x)

If you want to know the umask value simply type unmask as follows.

```
#umask
```

If you want to set the umask values as follows.

```
#umask 222
```

chmod : Change file access permissions.

Syntax : chmod << options >> <<access permissions >> << filename/directory >>

#mv Bachi.txt BachiReddt.txt
#mv BachiReddt.txt /root/Testing/TestDir1

777 --- > Change permission of the file, make it to executable by all (Owner, Group and Others).
400 --- > Change the permission of the file – only Owner can read.

#chmod –R 777 Testing

Here R represents recursively.

Following example shows another way of giving the permissions.

```
# clear
#  pwd
# ls -l
# chmod u-rwx Bachi.txt
# ls -l
# chmod og-rwx Bachi.txt
# ls -l
```

```
# chmod uog+rwx test.txt #
ls –l
# chmod uog-rwx test.txt #
ls –l
# chmod o+wx,u-x,g=r-x test.txt #
ls -l
```

+ represents adds the designated permission(s) to a file or directory.
- represents removes the designated permission(s) from a file or directory.
= represents Sets the designated permission(s)

#	Octal Permission Representation	Permission Reference
0	No permission	---
1	Execute permission	-x-
2	Write permission	-w-
3	Execute and write permission: 1 (execute) + 2 (write) = 3	-wx
4	Read permission	r--
5	Read and execute permission: 4 (read) + 1 (execute) = 5	r-x
6	Read and write permission: 4 (read) + 2 (write) = 6	rw-
7	All permissions: 4 (read) + 2 (write) + 1 (execute) = 7	rwx

We can change the permissions through WinSCP also as follows.
Once we logged/connect to remote server with WinSCP and right click on file/directory --- > Click on Properties, you will get the following window.

---

chown : Change file owner and group.

ls –l
chown simon test.txt
ls -l

ls –l
chown --from=root simon devops.txt
ls -l

Changing the group also.
 # ls –l
 # chown simon:staff devops.txt

© 2020 all rights Reserved by MyLandmark.tech +1 437 215 2483

 # ls -l

---

chgrp : Change file group.

ls –l
chgrp devops Bachi.txt
ls -l

---

cp: Copy files and directories.

e.g. cp Bachi.txt Bachi_cp.txt
cp Bachi.txt /root/Testing/TestDir1/BhaskarReddy

---

mv : Rename SOURCE to DEST, or move SOURCE to DIRECTORY/file.

Syntax : mv <<File/ Directory Actual Name>> << File/ Directory New Name>>

---

file : Determine file type.

#file /etc/hosts

---

wc : Print the number of newlines, words, and bytes in files

Syntax : wc [-c | -m] [-l] [-w] [-L] [File ...]

Options can be

- **-c Display number of bytes**
- **-m Display number of character.**
- **-l Display number of lines.**
- **-w Display number of words.**
- **-L Displays the length of the longest line.**

---

ln : Make links between files.

Two types of links:

1) Hard Links
2) Symbolic Links

The inode contains a list of all the blocks in which a file is stored, the owner information for that file, permissions, and all other attributes that are set for the file. In a sense, you could say that a file really *is* the inode, and names are attached to these inodes to make it easier for humans to work with them.

Use the **debugfs** command start to display the contents of the inode that you want to examine. For example, in this case you would type **stat <InodeNumber>**.

Use the **quit** command to close the **debugfs** interface.

Hard Links:

In Linux all directories and files have inodes. Creating hard link is nothing but add a new name to the inode. To do this we can use the **ln** command.
There is no difference between hard link and the original file.
Both files have the same inode number
Both files have the same size/contents.
Both files have the same access permissions.

#ln Bachi.txt Bachi_ln_test.txt

If you change one file contents it will update another file also

#ln Bachi.txt /root/Testing/Landmark/Bachi_ln_test.txt

Symbolic links:

The symbolic link and the original file have different inodes.
The size of the symbolic link is significantly different from the size of the real file.
The size of the symbolic link is the number of bytes in the name of the file it refers to, because no other information is available in the symbolic link.
The file type of the symbolic link is set to 1, which indicates that it is a symbolic link.

What is the difference between Hard Link and Symbolic Link?

A symbolic link is like a shortcut. It points to the original file and helps you find it easily. It breaks if you remove the original file.
A hard ink is like a copy of the original file that is synchronized continuously. There is no difference between the original file and the hard link; they both refer to the same blocks.
We can create symbolic link for both files and directories, but we cannot create hard links for directories, can create only for files.

Note: If you create hard links, inode number will be same for both files and if you create soft links, inode number will be different.

---

vim or vi : Vi IMproved, a programmers text editor.

#vi Bachi.txt
Then you can enter ‘Insert’ key or ‘i’. Now you are able to type text into that file(Bachi.txt). Once it finishes press ‘Esc’ button and type ’:’ and type ‘wq’. Option q! is for simply quit without save.

Now check with ls –l command.

See here size of the file is 54 bytes.

- Use “^” command to move to beginning of current line.
- Use “$” command to move to end of current line.
- Use “w” command to move forward by word.
- Use “b” command to move back by word.
- Use “e” command to move to end of word.
- Use “Ctrl-F” command to move to next screen.
- Use “Ctrl-D” command to move forward by half a screen.
- Use “Ctrl-B” command to move to previous screen.
- Position cursor to any opening bracket and press “%” key to see the matching closing bracket.
- Press “A” command to add data to end of current line.
- Use “yy” command to copy current line.
- Use “<n>yy” command to copy <n> lines from the current line.
- Use “p” to paste the lines cut using the “yy” or the “dd” command. The lines are pasted AFTER the current line.
- Use “P” to paste the lines cut using the “yy” or the “dd” command. The lines are pasted BEFORE the current line.
- Use “.” command to repeat the last add, update, delete or paste command.
- Use “x” command to delete current character.
- Use “<n>x” command to delete <n> characters beginning from the current character.
- Use “r<char>” command to replace current character with <char> character.
- Use “cw” command to change current word with new word(s). Enter new words and press ESC when done.
- Use “<n>cw” command to change <n> words beginning with current word with new word(s). Enter new words and press ESC when done.
- Use “C” command to replace remainder of the line. Enter new text and press ESC when done.
- Use “G” to move to end of file.
- Use “:/string” command to search the “string”.
- Use “n” command to repeat the previous search.
- Use “:s/old_string/new_string” command to substitute old_string with new_string in the current line.
- Use “:<m>,<n>s/old_string/new_string” command to substitute old_string with new_string in line “m” thru “n”.For first line set “m” to “1”. For last line, set “n” to “$”.

---

nano : This is also one of the editor like vi/vim.
Nano is a modeless editor so you can start typing immediately to insert text.

nano mithun.txt

Ctrl + O → For saving
Ctrl + X → Saving

---

echo : Display a line of text.

#echo Hi I am Mithun Reddy, working in Mithun Technologies Manyatha Tech park, Nagawara, Bangalore.
Hi I am Mithun Reddy L, working in Mithun Technologies Manyatha Tech park, Nagawara, Bangalore.
#echo I am Java application Developer and DevOps Engineer.
I am Java application Developer and DevOps Engineer. **---> See here it will not keep the more white spaces between am and Java**
[root@lexbz3149 ~]# echo "I am Java application Developer and Linux admin"
I am Java application Developer and Linux admin **---> See here it will keep the white spaces**

#echo $PATH
/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin:/root/bin:/root/Deployment_Ant/apache-ant-1.8.4/bin

[root@lexbz3149 ~]# echo "*"
*

---

cat : Concatenate files and print on the standard output.

cat simonlegah.txt

cat simonlegah.txt | tr a-z A-Z

cat -n hello.sh

tac : Does the same as cat, but displays the contents in reverse order

#tac simonlegah.txt

---

head : Print the first lines of each FILE to standard output.

Syntax: head [OPTION]... [FILE]...
This command is very useful when we want to see some lines in big file.

Options are
1) –n: output the last N lines, instead of the last 10.

e.g. 1) **#head simon.txt** : It will display the first 10 rows for specified file.

Note: By default, it will display 10 rows if don't provide any options for head command.

2) **#head -20 simon.txt** : It will display the first 20 rows for specified file.

(OR)

#head -n 20 simon.txt

Note: If you specify the more lines than the available in file, it will display the max lines in the file. See in the above example we specified the 20 lines in the command but only 15 lines there in the bhaskar.txt file.

---

tail : Print the last lines of each FILE to standard output.

Syntax: tail [OPTION]... [FILE]...

Options are

1) –f : output appended data as the file grows.
2) –n : output the last N lines, instead of the last 10.

e.g.

#tail simon.txt ---> It will display the last 10 rows.

#tail -f simon.txt : It will display last 10 lines of the growth of a file.

The # prompt does not return even after the work is over. With this option we have to abort the
process to exit to the shell. Use the interrupt key applicable on your machine. (Ctrl+c or q)

Note: By default, it will display 10 rows if don't provide any options for tail command.

#tail -n 3 simon.txt : It will display the last 3 rows for specified file.

(OR)

#tail -3 simon.txt

---

more : It is a filter for paging through text one screenful at a time (stop the display on each screen)

Syntax : more << file name >>

#more bhaskarreddy.txt

---

less : less is the same as more except you can scroll back and forward.

Syntax : less << file name >>

#less simon.txt

Following are options with less for navigation.

- CTRL+F – forward one window
- CTRL+B – backward one window
- CTRL+D – forward half window
- CTRL+U – backward half window
- G – go to the end of file
- g – go to the start of file
- q or ZZ – exit the less pager

---

sort : It is used to sort the output in numeric or alphabetic order .

---

tr: To translate the text from lower case to upper case.

---

sed : Sed is a Stream editor. A stream editor is used to perform basic text transformations on an input stream (a file or input from a pipeline).

FileName: sedfile.txt

Hi All,
My Name is Simon Legah, working as a DevOps Engineer in LT, Toronto. Currently DevOps is very good in market.
Learning DevOps tools are very easy. DevOps is not a technology. DevOps is a culture. DevOps is the combination of software development and operations team. DevOps helps an organization deploy software more frequently.

Replacing or substituting string

 # sed 's/DevOps/Java/' sedfile.txt
Here the ‘s’ represents the substitution operation.
‘/’ is delimeter.
‘DevOps’ is search pattern.
‘Java’ is the substitute string.

Note: By default, the sed command replaces the first occurrence of the pattern in each line and it won't replace the second, third...occurrence in the line.

Replacing the nth occurrence of a pattern in a line.

Use the /1, /2 etc flags to replace the first, second occurrence of a pattern in a line. The below command replaces the second occurrence of the word "unix" with "linux" in a line.

 # sed 's/DevOps/Java/1' sedfile.txt : It will replace the first occurrence.
 # sed 's/DevOps/Java/2' sedfile.txt: It will replace the second occurrence.

Replacing all the occurrence of the pattern in a line.
 # sed 's/DevOps/Java/g' sedfile.txt

'g' (global replacement is used to make the changes globally.

#sed -n "8p" simon_friends.txt : It will display the particular line number.
#sed -n "8, 12p" simon_friends.txt : It will display the particular line numbers.

To save the changes back to the file use the `-i` option:

http://www.folkstalk.com/2012/01/sed-command-in-unix-examples.html

---

grep : This command is used to search specific string in specified file. By default, grep prints the matching lines.

#grep "I am" simonlegah.txt

#grep -i "simon legah" simon.txt : Ignore case distinctions in both the PATTERN.

#grep -c -i "simon" bhaskarreddy.txt : With –c option it will display how many lines matches the given pattern/string.

#ps –ef | grep java : It will display all the java process.

who : It shows who is logged in the same machine.

#who -T :

#who -H :

#who am i : The use of the arguments with who am i causes the command to only lists the one user who typed the command.
The command is ***who*** and the arguments are ***am i***.

w : Show who is logged on and what they are doing.

The header shows, in this order, the current time, how long the system has been running, how many users are currently logged on, and the system load averages for the past 1, 5, and 15 minutes.

The following entries are displayed for each user: login name, the tty name, the remote host, login time, idle time, JCPU, PCPU, and the command line of their current process.

The JCPU time is the time used by all processes attached to the tty. It does not include past background jobs, but does include currently running back- ground jobs.

The PCPU time is the time used by the current process, named in the "what" field.

e.g. #w

#whoami : Print the user name associated with the current effective user ID. Same as id -un.

#id -un :

---

whereis: Path/locate the binary, source, and manual page files for a command.

Syntax : whereis << options >> << command >>

Options are:

-b Search only for binaries.

-m Search only for manual sections.

-s Search only for sources.

Examples:

 # whereis chmod
 # whereis -b chmod ---> Gives the path for binaries.
 # whereis -m chmod ---> Gives the path for manual sections.
 # whereis -s chmod ---> Gives the path for sources.

---

date : Displays or sets the date or time.

Syntax : date +[option]
The options can be:-

%D-displays date as mm/dd/yy
%a-displays abbreviated weekdays
%h-displays abbreviated month
%m-displays month of year
%d-displays day of the month
%y-displays last 2 digits of the month

%T-displays time as HH:MM:SS
%H-displays hour as between 00 to 23
%M-displays the minute between 00 to 59
%S-displays second as between 00 to 59

e.g.: date +%D
date +%a
date +%h
date +%m
date +%d
date +%y
date +%T
date +%H
date +%M
date +%S

#date +%Y%m%d -s "19840504"

#date -s "05 April 1984 04:00:00"

Note: It changed from EDT to EST.

#date +%T -s "10:12:12"

#date +%T%p -s "12:54:24 AM"

%p locale's equivalent of either AM or PM,

---

df : Report file system disk space usage.

Syntax: df [OPTION]... [FILE]...

OPTIONS

Show information about the file system on which each FILE resides, or all file systems by default.

Mandatory arguments to long options are mandatory for short options too.

-a, --all
include dummy file systems

-B, --block-size=SIZE use SIZE-byte blocks

-h, --human-readable
print sizes in human readable format (e.g., 1K 234M 2G)

-H, --si
likewise, but use powers of 1000 not 1024

-i, --inodes
list inode information instead of block usage

-k like --block-size=1K

-l, --local
limit listing to local file systems

--no-sync
do not invoke sync before getting usage info (default)

-P, --portability
use the POSIX output format

--sync invoke sync before getting usage info

-t, --type=TYPE
limit listing to file systems of type TYPE

-T, --print-type
print file system type

-x, --exclude-type=TYPE
limit listing to file systems not of type TYPE

-v (ignored)

--help display this help and exit

--version
output version information and exit

SIZE may be (or may be an integer optionally followed by) one of following: KB 1000, K 1024, MB 1000*1000, M 1024*1024, and so on for G, T, P, E, Z, Y.

#df –h

#df –a

---

du: Estimate file space usage.

Syntax: du [OPTION]... [FILE]...
du [OPTION]... --files0-from=F

DESCRIPTION
Summarize disk usage of each FILE, recursively for directories.

Mandatory arguments to long options are mandatory for short options too.

-a, --all
write counts for all files, not just directories

--apparent-size
print apparent sizes, rather than disk usage; although the apparent size is usually smaller, it may be larger due to holes in (âsparseâ) files,
internal fragmentation, indirect blocks, and the like

-B, --block-size=SIZE use SIZE-byte blocks

-b, --bytes
equivalent to â--apparent-size --block-size=1â

-c, --total
produce a grand total

-D, --dereference-args
dereference FILEs that are symbolic links

--files0-from=F
summarize disk usage of the NUL-terminated file names specified in file F

-H like --si, but also evokes a warning; will soon change to be equivalent to –dereference-args (-D)

-h, --human-readable
print sizes in human readable format (e.g., 1K 234M 2G)

--si like -h, but use powers of 1000 not 1024

-k like --block-size=1K

-l, --count-links
count sizes many times if hard linked

-m like --block-size=1M

-L, --dereference
dereference all symbolic links
-P, --no-dereference
donât follow any symbolic links (this is the default)

-0, --null
end each output line with 0 byte rather than newline

-S, --separate-dirs
do not include size of subdirectories

-s, --summarize
display only a total for each argument

-x, --one-file-system
skip directories on different file systems

-X FILE, --exclude-from=FILE
Exclude files that match any pattern in FILE.

--exclude=PATTERN Exclude files that match PATTERN.

--max-depth=N
print the total for a directory (or file, with --all) only if it is N or fewer levels below the command line argument; --max-depth=0 is the same as --summarize

--time show time of the last modification of any file in the directory, or any of its subdirectories

--time=WORD
show time as WORD instead of modification time: atime, access, use, ctime or status

--time-style=STYLE show times using style STYLE:
full-iso, long-iso, iso, +FORMAT FORMAT is interpreted like âdateâ

--help display this help and exit

--version
output version information and exit.

#du –h

Note: If no path is mentioned, default is current directory, where the action will be performed.

---

hostname: Show or set the systems host name.

#hostname
ralbz2072.cloud.dst.mt.com
#hostname ralbz2072
#hostname
ralbz2072

Once the hostname is changed, verify that it has changed the hostname successfully. As you see below, it has changed the hostname to ralbz2072.

If the hostname is not changed in /etc/hosts file, need to modify as follows.

#vi /etc/hosts
#vi /etc/sysconfig/network

Restart the server as follows.
#service network restart

How to find the IP address?
Ans) We can find the IP address as follows.

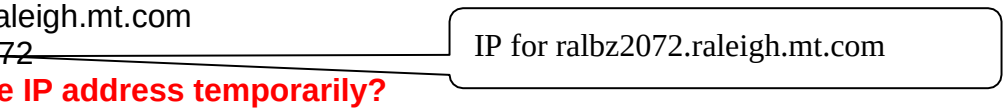
ifconfig: The "ifconfig" (interface configurator) command is used to setup network interfaces and allow the user to view information about the configured network interfaces.

#ifconfig

 # hostname -i
9.42.158.72

 # nslookup ralbz2072
Server: 9.0.6.1
Address: 9 0 6.1#53

Non-authoritative answer:
ralbz2072.cloud.dst.mt.com canonical name = ralbz2072.raleigh.mt.com.
Name: ralbz2072.raleigh.mt.com
Address: 9.42.158.72



How to change the IP address temporarily?

Ans) We can use ifconfig command to change IP Address as follows.
#ifconfig eth0 9.42.159.72

How to Change ip-address Permanently?
Ans) In /etc/sysconfig/network-scripts directory every machine have one file for network interface.
If the interface is “eth0”, you will see the ifcfg-eth0 file under /etc/sysconfig/network-scripts directory.
Change the IPADDR filed in the ifcfg-eth0 file.

Change the IP address in /etc/hosts file.

Restart the network as follows.
 # service network restart

#ifconfig –a : It will display details of all interfaces including disabled interfaces.

 # ifconfig eth0 down : Disable an interface.

 # ifconfig eth0 up : Enable an interface.

 # domainname (show or set the systems NIS/YP domain name**)**
(none)
 # dnsdomainname (show the systems DNS domain name**)**
lexington.mt.com
nisdomainname (show or set systems NIS/YP domain name**)**
(none)
 # ypdomainname (show or set the systems NIS/YP domain name**)**
(none)

---

nslookup : It will query the server name, ip address for specified host.

Syntax: nslookup << host name >>

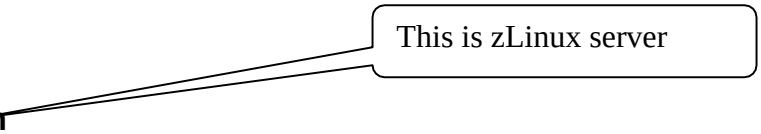
#nslookup ppdb2dev03
nslookup: parse of /etc/resolv.conf failed

#cat /etc/resolv.conf
DNS info generate from updateconfig
2011-10-30 12:23:26 PM
DO NOT MANUALLY MODIFY THIS FILE!

#
domain boulder.mt.com
search
nameserver 9.0.8.1
nameserver 9.0.9.1

Note : In resolv.conf file we need to delete **'serach'** word.

#nslookup ppdb2dev03
Server: 9.0.8.1
Address: 9 0 8.1#53



Non-authoritative answer:
Name: **ppdb2dev03.boulder.mt.com**
Address: 9.17.174.17

What is the difference between /etc/hosts and /etc/resolve.conf? What are the types of DNS records?

Ans) resolv.conf specifies the nameservers for resolver lookups, where it will actual use the DNS protocol for resolving the hostnames. Typically the hosts file is used for administrative purposes such as backend and internal functions, which is substantially more isolated in scope, as only the local server will reference it.

There are 5 types of DNS records:

A, CNAME, NS, MX, and PTR.

An "A" record, which stands for "address" is the most basic type of syntax used in DNS records, indicating the actual IP address of the domain. The "AAAA" record is an IPV6 address record that maps a hostname to a 128-bit Ipv6 address. Regular DNS addresses are mapped for 32-bit IPv4 addresses.

The "CNAME" record stands for "canonical name" and serves to make one domain an alias of another domain. CNAME is often used to associate new subdomains with an existing domain's DNS records.

The "MX" record stands for "mail exchange" and is basically a list of mail exchange servers that are to be used for the domain.

The "PTR" record stands for "pointer record" and maps an Ipv4 address to the CNAME on the host.

The "NS" record stands for "name server" and indicates which Name Server is authoritative for the domain.

An “SOA” record stands for “State of Authority” and is easily one of the most essential DSN records because it stores important information like when the domain was last updated and much more.

An “SRV” record stands for “service” and is used to define a TCP service on which the domain operates.

A “TXT” record lets the administrator insert any text they'd like into the DNS record, and it is often used for denoting facts about the domain.

Help Commands

man : Displays manual entries online.(To get help of the commands.)

Syantax : man <<Type any command>>

e.g. man man
man who

To search any word in “man” page use “/” and string name.
“n” for checking next search and “N” for previous search.
To quit press “:q”.

What is the differance between Ctrl +c and Ctrl + Z?
Ans)

Ctrl + c : It will kill the process with a signal SIGINT.

Ctrl + z : It will suspend the process with a signal SIGSTOP.

However when a process is suspended , we can resume it again by fg (resume in forground) and bg (resume in background) ,
but i cant resume a killed process , that is a difference between using Ctrl+C & Ctrl+Z.

By Using 'jobs' command we can see the suspended jobs.

By using fg command we can run the suspended job in foreground and using bg command we can run the suspended job in backroundas follows.

fg --> If only one job is suspended.
(OR)
fg %n --> If more than one job is suspended, we will use this command, here 'n' is the job number.

bg

(OR)
bg %n

By using kill command kill the susspended processas follows.

kill %n --> where n will be numbers displayed in jobs command , so if you want to kill 1st process use below command.

kill %1 --> It will kill the first job.

---

info : Gives the information about any command.

Syantax : info <<Type any command>>

e.g.
#info man
#info who

---

help : It gives a short explanation about how to use the command and a list of available options.

Syantax : <<Type any command>> --help

e.g.
#ls --help
#passwd --help

The Gnome Help Browser is very user friendly as well. You can start it selecting Applications->Help from the Gnome menu, by clicking the lifeguard icon on your desktop or by entering the command **gnome-help** in a terminal window. The system documentation and man pages are easily browsable with a plain interface.

You may want to try "info:info" in the *Location* address bar, and you will get a browsable Info page about the **info** command. Similarly, "man:ls" will present you with the man page for the **ls** command.

---

apropos : Search information about a command or keyword.

Suppose if you don't know how to start a browser, then you could enter the following command:

#apropos browser

#apropos editor

---

service : This command is used to find the service is running in the server or not.

Syntax: << command >> << options >>

Options are:
status:
stop
start
restart
status-all

Examples:

 # service sshd status: It will displays the status of the specified service.

 # service --status-all : It will display the status of all the services.

 # service sendmail stop : It will stop the specified service.

 # service sendmail start : It will start the specified service.

 # service sendmail restart : It will stop and restart the service.

#service crond status : It will check the status of cron is running or not?

Note: All services are available in /etc/rc.d/init.d directory.

 # /etc/init.d/nfs start

---

ps : This command can used to display the currently running processes on Linux systems.
ps is the shortage for Process Status.

 # ps : It will display the user's currently running processes

#ps -f : Full listning of the user's currently running processes

#ps -ef: Full listing of all processes, except kernel processes

#ps -A (OR) ps -Af : All processes, including kernel processes.
Here f means full listing. You can observer output without f option and with f option.

 # ps ux :
(OR)
 # ps U mithun :
} It display all processas owned by a specific user.

 # ps aux : It will display every process on the system.

pstree : It will display a tree of processes.
-A ---> Use ASCII characters to draw the tree
 # pstree -A :

 # pstree -G :
-G ---> Use VT100 line drawing characters.

Note : -A ---> This opion uses ASCII characters to draw the tree.
-G --->This opion uses VT100 line drawing characters.

In AIX:

p s -ef | grep java
ps -ef | grep httpd
ps -ef | grep db2
ps -ef | grep vnc

Here -e means select all processes and -f means full format listing.

ps -eo pid,cmd,%mem,%cpu,stat | less: It will display the process which are currently running in following format.

ps -eo pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,wchan:14,comm : It will display the user defined format

If you type Spacebar it will display screen by screen. If you go back you need to enter 'B'.

---

kill : The command will sends the specified signal to the specified process

#kill –l : It will display all signal names. These are found in /usr/include/linux/signal.h

#kill -9 <<Process ID>> : It will kill the specified process.

#kill -3 <<JavaProcess ID>> : It will generate the core dump for Java processes.

#kill all -u <<username>> : It will kill all the processes under specified user.

---

top : The top program provides a dynamic real-time view of a running system. It can display system summary information as well as a list of tasks currently being managed by the Linux kernel. The types of system summary information shown and the types, order and size of information dis-played for tasks are all user configurable and that configuration can be made persistent across restarts.

 # top

Press (**Shift+O**) to Sort field via field letter, for example press '**a**' letter to sort process with PID (**Process ID**).

Press '**z**' option in running top command will display running process in color which may help you to identified running process easily.

#top -u simon : It will display specific user process details.

Press '**c**' option in running top command, it will display absolute path of running process.

By default screen refresh interval is **3.0** seconds, same can be change pressing '**d**' option in running top command and change it as desired as shown below.

You can kill a process after finding **PID** of process by pressing '**k**' option in running top command without exiting from top window as shown below.

Press (**Shift+P**) to sort processes as per **CPU** utilization. See screenshot below.

Press (**Shift+W**) to save the running top command results under **/root/.toprc**.

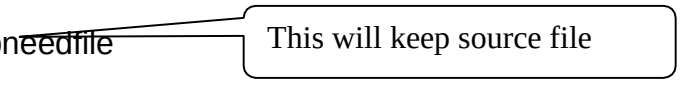
Top output keep refreshing until you press '**q**'. With below command top command will automatically exit after 10 number of repetition.

 # top -n 10

File Archieve/Extraction Commands

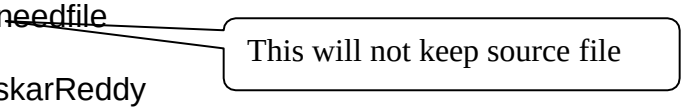
zip : zip, zipcloak, zipnote, zipsplit - package and compress (archive) files.

Syntax : zip -9 -r desiredname.zip zipneedfile



or

zip -rm desiredname.zip zipneedfile



e.g.: zip -9 -r BhaskarReddy.zip BhaskarReddy
zip -rm Bhaskar.zip BhaskarReddy

unzip : unzip - list, test and extract compressed files in a ZIP archive.

Syntax : unzip filename.zip

e.g. : unzip MithunReddy.zip

unzip text.zip -d /tmp:

---

tar : It is used to archive the directory/file.

Syntax: tar options directory/file

The Options may be follows.

- c – create a new archive
- v – verbosely list files which are processed.
- f – following is the archive file name

#tar cvf /home/isgproject/scripts/Java/test.tar Target/: Creating an uncompressed tar archive using option cvf

#tar cvzf test.tar.gz /home/isgproject/scripts/Java/Target/: Creating a tar gzipped archive using option cvzf
z – filter the archive through gzip

Note: .tgz is same as .tar.gz

#tar xvf test.tar : Extract a *.tar file using option xvf.
Here x means extract files from archive.

#tar xvfz test.tar.gz : Extract a gzipped tar archive (*.tar.gz) using option xvzf

#tar tvf test.tar : View the tar archive file content without extracting using option tvf

#tar tvf test.tar.gz : View the *.tar.gz file content without extracting using option tvzf

#tar rvf test.tar Source/ : Adding a file or directory to an existing archive using option –r

Note: You cannot add file or directory to a compressed archive. If you try to do so, you will get “tar: Cannot update compressed archives” error as shown below.

tar rvfz test.tar.gz Source/
Aborted

#tar -cf - Target/ | wc –c : Estimate the tar archive size.

User Administration Commands

useradd : Creates a new user account.

e.g. useradd chinnu

passwd : Changes a user's password.

e.g. passwd chinnu

To login

#login <<User Name>>
To logout
#logout <<User Name>>
#exit
#Ctrl+d

What is the range of system resvered uid in RHEL?
Ans)
UID_MIN 1000
UID_MAX 60000
This information availbe in /etc/login.defs file.
New user created on RHEL 7 has a UID above 1000, but on RHEL 6, it starts at 500.

How to lock and unlock user account in Linux

#passwd -u simon

Before password lock implemented, the entry for isgproject as below.
simon:1n7anFLQG$XDx/hR0YNbVw42.vylRF10:15835:0:99999:7:::

After setting password lock:
simon:!!1n7anFLQG$XDx/hR0YNbVw42.vylRF10:15835:0:99999:7:::

Now not able to login with simon user because !! in the password field of /etc/shadow file for user simon as shown above screen shot.

Try to login with **simon** user.

 # passwd -u simon

#passwd -d < User Name > : It will deletes the already existing password in /etc/shadow file for specified user, resulting it restricts the login into server.

#passwd -d simon

Note: # Indicates root user

& Indicates normal user

---

newusers: It is used to create or update new users in batch.
Create a file, formatted like /etc/passwd, that contains the usernames and plain text passwords of the users.
This command is intended to be used in a large system environment where many accounts are updated at a single time.

Syntax:
newusers <filename>

Following is the syntax of the parameters to be provided for each user in file.
<Username>:<Password>:<UID>:<GID>:<User Info/Comments>:<Home Dir>:<Default Shell>

- Username: User login name
- Password: User password
- UID: User Identifier
- GID: Group Identifier for user's primary group
- User Info: User information like full name, contact information etc.
- Home Dir: User's home directory
- Default Shell: User's default shell

Example:
File Name: UsersList
bhaskar:bhaspwd@2013:500:500:Bhaskar User ID:/home/bhaskar:/bin/bash
mithun:mithpwd@2013:501:501: Mithun User ID:/home/mithun:/bin/bash

Note:
Password should contain at least 8 characters length.
It should contain at least one special character like @, & …
It should contain at least one number.
It should contain at least one Capital letter.
It should contain at least one small letter.

#newusers UsersList

If you are using MD5 passwords(default), ensure that newusers will generate MD5 paswords, else the users will not be able to login.

This entry can be check in /etc/login.defs file as follows.

MD5_CRYPT_ENAB yes

Verify the below files got updated or not.

/etc/group
/etc/passwd
/etc/shadow

Check in /home path directory got created with username provided.

---

chage : It is used to see user related **"threshold details"** such as user disable time etc. The chage program requires a shadow password file to be available.
#chage -l

#chage simon

#chage -E never simon : It will make user 's(simon) account will not expire.

---

groupadd: Create a new group.

---

usermod : Changes user attributes.

usermod –G <<Group Name>> <<User Name>>

#usermod –G linuxadm simon

For locking:
#usermod -e : The date on which the user account will be disabled. The date is specified in the format YYYY-MM-DD.

#usermod -e 2021-05-23 simon

---

groupmod : Modify a group.

Syntax: groupmod -n << group name >> << new group name >>

#groupmod -n simonadmin linuxadmin

In the above example the groupmod command would change the group "linuxadmin" to " simonadmin".

---

newgrp : It is used to change the current group ID during a login session.

Syntax : newgrp << group name >>

For example if you are a member of the groups "linuxgrp" and "javagrpyou can switch to the other group with the following command:

Every Linux user can be assigned to multiple groups so they can obtain access to whatever files and directories they need to perform their work. Users can determine the groups to which they have access by using the id command.
#id
#newgrp mygroup #id

Now if you create any file or directory, it will assign the group as ***mygroup***, as shown below.

id : It is one more command which will show the user details such as his primary group and his secondary group.(Displays the system identifications of a specified user.)

Examples:

#id myland
#id landmark
#id simon
#id root
#id

lid: Display user's groups or group's users. → Only root user can execute

su : Switches the user between users and root. (Changes the user ID associated with a session.)

#su mark
#whoami
#exit
#whoami

users: Displays a compact list of the users currently logged on the system.

e.g.
#users

Note: See first it displays only one user name called root, after that I logged with bhaskar user in other terminal, then it shows two users.

---

groups : Displays group membership.
#groups <<User Name>>

---

userdel : Removes a user account.

Syntax: userdel << user name >>

Examples:

#userdel simon
#userdel -r simon ---> It will delete/remove the simon account and also the simon home directory, in this case (/home/simon).

---

groupdel: Deletes a new group.

Syntax: groupdel << group name >>

#groupdel linuxadmin

---

finger : Finger displays the users login name, real name, terminal name,shell and write status, idle time, login time, office location and office phone number.

#finger
#finger mark
#finger root

---

crontab : Crontab is the program used to install, deinstall or list the tables used to drive the cron daemon in Linux Operating system.

Crontab format:

# Minute (0-59)	Hour (0-23)	Day of Month (1-31)	Month (1-12 or Jan-Dec)	Day of Week (0-6 or Sun-Sat)	Command /usr/bin/find

Listing a crontab job
#crontab –l

Editing a crontab job
crontab -e

Removing a crontab job
#crontab –r

Example:

*/1 * * * * /home/bhaskar/devops/lscmd.sh >> /home/bhaskar/lscmd.log 2>&1
This will execute lscmd file every one minute.

To restrict any user from using cron job facility, enter their names in /etc/cron.deny and save it

#vim /etc/cron.deny

Now login as one of those users and try to use crontab.

If again want to allow bhaskar1user to use cron job facilities just remove bhaskar1 user from /etc/cron.deny file.

Assuming that we have 100 users in our system, putting all 98 names in /etc/cron.deny file is a time consuming process. Instead of that, we can create one more file /etc/cron.allow, in which we can assign names of those users who are allowed to use cron jobs. Remove the /etc/cron.deny file and create /etc/cron.allow, still if both files are existing cron.allow file will be having precedence over cron.deny file. Just to avoid confusion it is good to remove cron.deny file

Note: /etc/cron.deny file exists by default, but we need to create /cron.allow file. If your name is not there in cron.allow file then you will not be allowed to use cron jobs, and as mentioned above, if both files are existing cron.allow file will be having precedence over cron.deny file. If neither cron.deny nor cron.allow files exists then only root can use cron jobs.

To find RAM details in Linux

free : To find the amount of free and used RAM memory in the system

Syntax : free [-b|-k|-m|-g] [-l] [-o] [-t] [-s delay] [-c count] [-V]
-b,-k,-m,-g show output in bytes, KB, MB, or GB
-l show detailed low and high memory statistics
-o use old format (no -/+buffers/cache line)
-t display total for RAM + swap
-s update every [delay] seconds
-c update [count] times
-V display version information and exit

#free

#free -b

#free -k

#free -m

#free -g

#free -l

#free -o

#free -t

#free -V

 # dmidecode -t 16 : It gives the RAM information like location, Maximum capacity of RAM supported, number of RAM devices present and slot info.

How to enable a password login instaed of pem file when logging into EC2 Instance using SSH?

==========

Procedure to Enable the password login

Step 1) First login into ec2 instance with pem file, then switch to root user.

ssh -i "landmarktechnologies.pem" ec2-user@ec2-52-66-196-244.us-south-1.compute.amazonaws.com

sudo su - (OR) sudo -i

Step 2) Update the PasswordAuthentication parameter in the /etc/ssh/sshd_config file as follows.

vi /etc/ssh/sshd_config

PasswordAuthentication yes

Step 3) Restart the sshd service as follows.

service sshd restart

Step 4) Set a password for ec2-user as follows.

passwd ec2-user

---

ssh: ssh (SSH client) is a program for logging into a remote machine and for executing commands on a remote machine. It is intended to provide secure encrypted communications between two untrusted hosts over an insecure network.

Syntax: ssh Username@Hostname

---

scp : Secure copy (remote file copy program)

Sourece Server : dst02lp3.boulder.mt.com
Sourece Server Directory: /AHRefresh/data/lp3data/AHNEW/

Target Server : ahdev.pok.mt.com
Target Server Directory: /AHstagdata/

Ex:1 scp /AHRefresh/data/lp3data/AHNEW/20110816.zip root@ahdev.pok.mt.com:/AHstagdata/

2. scp -r /AHRefresh/data/lp3data/AHNEW/ root@ahdev.pok.mt.com:/AHstagdata/

---

write : Opens a line of communication to send messages to other users on the system in real time.

Syantax: # write <<username>>

e.g.: # write root
Hi
Ctrl + D

Note : Below screen shot shows the message got from the write command.

---

mesg: Permits or refuses write messages from other users.

Syantax : $ mesg <<options>>

Options are “n” or “y”.

mesg n ---> Allows only the root user the permission to send messages to your workstation.

mesg y ---> Allows all workstations on the local network the permission to send messages to your
workstation.
e.g.: $ mesg n

Now I tried with user simon to send the message to user simon. See the output.

Note: Here simon is normal user.

Here the user is root. Root can able to send the message to any user even they disabled.

---

wall : Writes a message to all users that are logged in.

Syntax : # wall [-a] [-g Group][Message]

---

mail : Sends and receives mail.

#mail mylandmarktech@gmail.com
Once you enter the to address It will ask Subject then Press Enter Body of mail then type Ctrl+D it will ask CC address, then enter CC address and press Enter button.

#mail : It is used to read the mails.
#mail -s "Hello" root < simon.txt

Here Hello is the Subject and contents in bhaskar.txt is the subject for mail.

#echo "This is for body of mail" | mail -c satya@in.mt.com -s "Sending mail using mail command" simon@in.mt.com -b mylandmarktech@in.mt.com

Here options are:

-s subject Use subject for the e-mail title
-c address Send copy of the mail to the specified address
-b address Send blind carbon copy to specified address

---

mutt: This command is used to send a mail along with attachment.

#mutt -s "Airport Photo" -a Airport.jpg simon@mt.com -c mylandmarktech@mt.com -b manan@mt.com \ < simon.txt

mutt always reads a text from standard input that will become the main body of the e-mail, the text before the attachment. If the mail should consist of attachments only, we can either specify /dev/null as the file to read, e.g.

#mutt -s "Airport Photo" -a Airport.jpg simon@mt.com -c mark@mt.com -b mylandmarktech@mt.com \ < /dev/null

or use an empty line as the mail body:

#echo | mutt -a Airport.jpg simon@mt.com -c mark@mt.com -b landmark@mt.com

---

clear : Clears the terminal screen.

---

exit (OR) Ctrl+d : Terminate the process, returning returnCode to the system as the exit status. If
returnCode isn't specified then it defaults to 0.

---

wget: The non-interactive network downloader.

Following example will download and store the file with name: **apache-ant-1.8.4-src.tar.gz**

e.g. # wget http://www.apache.org/dist/ant/source/apache-ant-1.8.4-src.tar.gz

Download in progress:

- %age of download completion (for e.g. 48% as shown above)
- Total amount of bytes downloaded so far (for e.g. 1,928,742 bytes as shown above)
- Current download speed (for e.g. 450K/s as shown above)
- Remaining time to download (for e.g. eta 8s as shown above)

Download in completed:

If you run the same command twice the file name will be saved suffix with .1 as follows.

If we want to customize the downloaded output file name, we -O option as:
wget -O AntSoftwares.zip http://www.apache.org/dist/ant/source/apache-ant-1.8.4-src.tar.gz

Using wget –c continue the incomplete download as follows.

Record your session

script : This command records your login session in a typescript (file) in the current directory. All the commands, their output and the error messages are stored in the file (typescript --> ASCII text file) for later viewing.

e.g.
 # script
script
Script started, file is typescript
 # ls
ls
anaconda-ks.cfg isus
Desktop plglogs pluginGSKitUpgradeLog.txt
ihsGSKitUpgradeLog.txt Read_Me_IC2_Default.properties
ihslogs typescript
install.log updilogs
install.log.syslog waslogs #
pwd
pwd
/root
 # whoami
whoami
root
 # exit
exit
exit
Script done, file is typescript

spell and **ispell** are used to spell-check a document. spell produces a list of misspelled words, while ispell lets you perform the corrections on-line.

With -a option we can store all session commands in customized file.

How to find Linux Flavours and Version

mikey@learnwithpridesolutions:/> cat /etc/SuSE-release (Production Server)

simon@b03zirdb12c:~> cat /etc/issue

[root@phlox /]# cat /etc/redhat-release (CVS Server)

[mikey@phlox ~]$ cat /etc/issue (CVS Server)

[root@ralbz2072 ~]# cat /etc/redhat-release (Dev Server)

[root@ralbz2072 ~]# cat /etc/issue

ppwasdev03:/home/LearnWithPride$ cat /etc/SuSE-release (UAT Server)

[root@ralbz2072 /]# cat /etc/redhat-release (Dev Server)

(OR)

 # lsb_release -a

~~**#cat /etc/*release**~~----------

uname : Print system information.

Syntax: uname [OPTION]...

DESCRIPTION

Print certain system information. With no OPTION, same as -s.

-a, --all
print all information, in the following order, except omit -p and -i if unknown:

-s, --kernel-name
print the kernel name

-n, --nodename
print the network node hostname

-r, --kernel-release
print the kernel release

-v, --kernel-version
print the kernel version

-m, --machine
print the machine hardware name

-p, --processor
print the processor type or "unknown"

-i, --hardware-platform
print the hardware platform or "unknown"

-o, --operating-system
print the operating system

--help display this help and exit

--version
output version information and exit

Below command also give the kernel version.

#cat /proc/version

To find Physiacl RAM(Linux)

#dmesg | grep mem

(OR)

#cat /proc/meminfo

To find Physiacl RAM(AIX)
#lsattr –E –l sys0 –a realmem

[root@ralbz2072 /]# cat /proc/cpuinfo |grep processor ---> It will display the CPUs names

(OR)

[root@ralbz2072 /]# cat /proc/cpuinfo | grep processor | wc –l ---> It will display the CPUs count

To find the Number of CPUs(AIX)
#lsdev -C | grep Proc

---

last: Show listing of last logged in users.

Syntax:
last [-R] [-num] [-n num] [-adiowx] [-f file] [-t YYYYMMDDHHMMSS] [name...] [tty...]

lastb [-R] [-num] [-n num] [-f file] [-adiowx] [name...] [tty...]

OPTIONS

-num This is a count telling last how many lines to show.

-n num The same.

-t YYYYMMDDHHMMSS
Display the state of logins as of the specified time. This is useful, e.g., to determine easily who was logged in at a particular time -- specify that time with -t and look for "still logged in".

-f file
Specifies a file to search other than /var/log/wtmp.

-R Suppresses the display of the hostname field.

-a Display the hostname in the last column. Useful in combination with the next flag.

-d For non-local logins, Linux stores not only the host name of the remote host but its IP number as well. This option translates the IP number back

into a hostname.

-i This option is like -d in that it displays the IP number of the remote host, but it displays the IP number in numbers-and-dots notation.

-o Read an old-type wtmp file (written by linux-libc5 applications).

-w Display full user and domain names in the output.

-x Display the system shutdown entries and run level changes.

#last

#cat /etc/shells : It will give all available shells in your system

---

#echo $SHELL : It will give you the current shell type.

How to change the Shell type?

```
[root@lexbz3149 ~]# echo $0
bash
[root@lexbz3149 ~]# sh
sh-3.2# bash
[root@lexbz3149 ~]# echo $0
bash
[root@lexbz3149 ~]# tcsh
[root@lexbz3149 ~]# echo $0
tcsh
[root@lexbz3149 ~]# csh
[root@lexbz3149 ~]# echo $0
csh
[root@lexbz3149 ~]# ksh
# echo $0
ksh
# bash
[root@lexbz3149 ~]#
```

Note: If you want to come out from shell type ‘exit’

In Linux default shell is /bin/bash and in AIX default shell is /usr/bin/ksh.

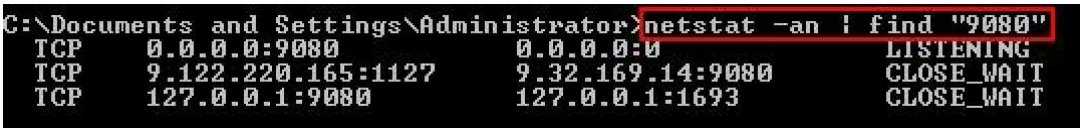
---

netstat : Netstat prints information about the Linux networking subsystem.

 # netstat -tupl

 # netstat –tup

netstat –an |find “9080”



 # netstat -an | grep 9060

 # netstat -a -n –o : Windows it will display all ports and which port is used by which pid

Linux : netstat -tulpn | grep :9080 it will display who is using porticular port

Windows: netstat -a -n -o Windows it will display all ports and which port is used by which pid

AIX : netstat -Aan | grep LISTEN | grep 50000 it will display socket ID
(EX: f1000e0008b1cbb8)
rmsock socket id we got withp revious command tcpcb
EX: rmsock f1000e0008b1cbb8 tcpcb
it will display the pid

---

lsof:

 # lsof -i | grep 9060

---

cal : Displays a calendar.

#cal -3

#cal 10 1984

#cal 8 1984

---

clock : Determine processor time.

---

history : Displays the recently executed commands.

Typically when you type history from command line, it displays the command # and the command. For auditing purpose, it may be beneficial to display the time stamp along with the command as shown below.

#export HISTTIMEFORMAT='%F %T '
#history :

When you've already executed a very long command, you can simply search history using a keyword and re-execute the same command without having to type it fully. Press Control+R and type the keyword. In the following example, I searched for red, which displayed the previous command "cat /etc/redhat-release" in the history that contained the word red.

 # [Press Ctrl+R from the command prompt,which will display the reverse-i-search prompt]
(reverse-i-search)`red': cat /etc/redhat-release
Note: Press enter when you see your command,which will execute the command from the history
 # cat /etc/redhat-release
Fedora release 9 (Sulphur)
Sometimes you want to edit a command from history before executing it. For e.g. you can search for httpd, which will display service httpd stop from the command history, select this command and change the stop to start and re-execute it again as shown below.

 # [Press Ctrl+R from the command prompt,Which will display the reverse-i-search prompt]
(reverse-i-search)`httpd': service httpd stop
[Note: Press either left arrow or right arrow key when you see your command, which will display the command for you to edit, before executing it]
 # service httpd start

- **Ctrl+P :** To check the previous command in history.
- **!! :** To check the previous command in history.
- **!-l :** To check the previous command in history.
- Use the **up arrow** to view the previous command and press enter to execute it.
- **!4 :** To execute the 4th command in the history commands.
- **!<type few letters of any command> :** Execute previous command that starts with a specific word.

Type ! followed by the starting few letters of the command that you would like to re-execute. In the following example, typing !who and enter, executed the previous command starting with who, which is 'who am i.
e.g. #!who

Control the total number of lines in the history using HISTSIZE
Append the following two lines to the .bash_profile and **relogin** to the bash shell again to see the change. In this example, only 500 command will be stored in the bash history.

```
# vi ~/.bash_profile
HISTSIZE=500
HISTFILESIZE=500
```

By default, history is stored in ~/.bash_history file.
If we want to open the **bash_history file** use the following command.

```
# vi ~/.bash_history
```

Change the history file name using HISTFILE
Add the following line to the .bash_profile and relogin to the bash shell, to store the history command in .commandline_warrior file instead of .bash_history file. I'm yet to figure out a practical use for this. I can see this getting used when you want to track commands executed from different terminals using different history file name.

```
# vi ~/.bash_profile
HISTFILE=/root/.commandline_warrior
```

Eliminate the continuous repeated entry from history using HISTCONTROL

In the following example **date** was typed three times, when you do history, you can see all the 3 continuous occurrences of it. To eliminate duplicates, set HISTCONTROL to ignoredups as shown below.

```
# date
Thu Apr 25 06:09:12 EDT 2013
# date
Thu Apr 25 06:09:14 EDT 2013
# date
Thu Apr 25 06:09:16 EDT 2013
# history | tail -10
    1  date
    2  date
    3  date
    4  history | tail -10
```

Note that there are three date commands in history, after executing date 3 times as shown above

```
# export HISTCONTROL=ignoredups
# date
```

```
Thu Apr 25 06:14:22 EDT 2013
# date
Thu Apr 25 06:14:23 EDT 2013
# date
Thu Apr 25 06:14:25 EDT 2013
#  history | tail -10
    1  date
    2  date
    3  date
    4  history | tail -10
    5  export HISTCONTROL=ignoredups
    6  date
    7   history | tail -10
```

Note that there is only one date command in the history, even after executing date 3 times as shown above

Erase duplicates across the whole history using HISTCONTROL

The ignoredups shown above removes duplicates only if they are consecutive commands. To eliminate duplicates across the whole history, set the HISTCONTROL to erasedups as shown below.

```
# export HISTCONTROL=erasedups
# date
Thu Apr 25 06:18:44 EDT 2013
# date
Thu Apr 25 06:18:45 EDT 2013
# date
Thu Apr 25 06:18:47 EDT 2013
# service sendmail stop
Shutting down sm-client:                   [  OK  ]
Shutting down sendmail:                    [  OK  ]
# history | tail -5
    3   history | tail -10
    4  export HISTCONTROL=erasedups
    5  date
    6  service sendmail stop
    7  history | tail -5



# pwd
/root
# service sendmail stop
Shutting down sendmail:                    [FAILED]
# history | tail -5
```

```
4  export HISTCONTROL=erasedups
5  date
6  pwd
7  service sendmail stop
8  history | tail -5
```

Note that the previous service sendmail stop after date got erased

Force history not to remember a particular command using HISTCONTROL

When you execute a command, you can instruct history to ignore the command by setting HISTCONTROL to ignorespace AND typing a space in front of the command as shown below. I can see lot of junior sysadmins getting excited about this, as they can hide a command from the history. It is good to understand how ignorespace works. But, as a best practice, don't hide purposefully anything from history.

```
# pwd
/root
# date
Thu Apr 25 07:43:27 EDT 2013
# whoami [Note that there is a space at the beginning of service,to ignore this command from history]
root
# history | tail -10
  1  export HISTCONTROL=ignorespace
  2  pwd
  3  date
  4  history | tail -10
```

history -c : To clear the history.

Disable the usage of history using HISTSIZE

If you want to disable history all together and don't want bash shell to remember the commands you've typed, set the HISTSIZE to 0 as shown below.

```
# export HISTSIZE=0
# history
# ------ > Note that history did not display anything.
```

Ignore specific commands from the history using HISTIGNORE

Sometimes you may not want to clutter your history with basic commands such as pwd and date. Use HISTIGNORE to specify all the commands that you want to ignore from the

history. Please note that adding ls to the HISTIGNORE ignores only ls and not ls -l. So, you have to provide the exact command that you would like to ignore from the history.

```
# export HISTIGNORE="pwd:date:whoami:ls:"
# pwd
/root
# date
Thu Apr 25 08:04:16 EDT 2013
# whoami
root
# ls
anaconda-ks.cfg                Desktop               ihslogs           isus
Read_Me_IC2_Default.properties  updilogs       waslogs
install.log        plglogs                SERVERINFO                    vpd.properties
client_config_update.py         ihsGSKitUpgradeLog.txt install.log.syslog
pluginGSKitUpgradeLog.txt  typescript                  WAS
# who
root     pts/0        2013-04-25 08:02
# history
   1  export HISTIGNORE="pwd:date:whoami:ls:"
   2  who
   3  history
```

Note that history did not record pwd, date, whoami and ls.

---

reboot : To reboot the system.

#reboot

reboot -f : To reboot the system forcibly.

---

poweroff (OR) halt –p (OR) init 0 (OR) shutdown -h now : To shutdown the system.

Once we execute the **init 0** command we will get the below error. Means server got shutdown.

---

type : Writes a description of the command type.

Syntax: type <<CommandName>>

#type ps
#type man
#type cd
#type java

---

uptime : Shows how long the system has been up.

Syntax: uptime

e.g : uptime

How to check the java version?

java -version

Keystroke	Function
Ctrl+s	Stops scrolling of screen output
Ctrl+q	Resumes scrolling of screen output
Delete	Interrupts a command (Use Ctrl+c if this fails)
Ctrl+d	Terminates login session (Use exit also)
Ctrl+h	Erases text (if backspace key does'nt work)
Ctrl+u	Kills command line without executing it
Ctrl+\	Kills command line without executing it
Ctrl+j	Alternative to Enter
Ctrl+m	Alternative to Enter
Sty sane	Restore terminal to normal status

> is used to remove/nullify the contents from file, without deleting the file.

#cat /dev/null > simon.txt : It will clear/null the contents of bhaskar.txt file.

#ls | tee teeoutput.txt: It will display the output in console as well into file also.

#ls | tee -a teeoutput.txt: It will append to the file.

#ls | tee -a teeoutput1.txt teeoutput2.txt teeoutput3.txt: It will append to the file.

---

jobs: Lists the jobs that you are running in the background and in the foreground.

Note: If the prompt is returned with no information no jobs are present.

jobs -l: list only jobs that have change status since last notified by their status

jobs -p: list only the PID of process group leader
jobs -s: restrict output to stopped jobs.
jobs -r:

---

env : Display environment, set environment for process
If you call **env** with no arguments, it displays the environment.

In Dev Server.

#env

How to set the class path?
Here we are setting the Java class path.

PATH=$PATH:*/usr/java14/jre/bin*
export PATH
echo $PATH

How do you see Linux server system reboot history?

#last reboot
#who --boot
#who –b

Provided the most succinct reply

Disply top 5 files/directories taking your disk space in current directory?

 # du -hsx * | sort -r | head -5

#tree: List contents of directories in a tree-like format.

Errors:

#useradd manan

This issue occurs if there is passwd.lock or passwd- or passwd~ file in the /etc/directory.

These types of files will create when someone must have run the useradd command and it did not finish correctly.

Solution 1: Exit from terminal and relogin.

For locking:
#chage -E 2013-05-23 simon

#usermod -s /sbin/nologin simon

 # cat /proc/meminfo

 # vmstat -s

---

fdisk: fdisk stands (for “fixed disk or format disk“) it is a most commonly used command-line based disk manipulation utility for a Linux/Unix systems. With the help of fdisk command you can view, create, resize, delete, change, copy and move partitions on a hard drive using its own user friendly text based menu driven interface.

#fdisk -l: It is used to view all available partitions on Linux

In RedHat OS

In Cent OS

#fdisk -l /dev/xvda: It will display all partitions of specific hard disk

#fdisk -l /dev/xvdb

https://www.tecmint.com/fdisk-commands-to-manage-linux-disk-partitions/

---

root@ralbz2072 ~]# dmidecode -t 17: It Give the RAM information like Type of RAM(SD RAM, DRAM or DDR2/3), Speed, Manufacture etc

---

cut : Cut command in unix (or linux) is used to remove sections from each line of files.

#cat cut_example.txt
Hi All,
I am Mithun Reddy L, working as a Java application developer in MT Manyatha Tech Park.
I am very good in Linux administration as well as WebSphere Application server also.

#cut –c4 cut_example.txt: With the -c option, cut will cut out specific characters (i.e., columns) in the input lines. This will displays 4th character from each line of a file cut_example.txt.

#cut -c1-3 cut_example.txt : It willextracts first 3 characters of each line from a file called cut_example.txt.

#cut -c3- cut_example.txt : This will extracts from 3rd character to end of each line from cut_example.txt.

#cut -c1-3 cut_example.txt : When you don't specify a number before or after the '-' , it will print entire line as follows.

#cut -c1-3 cut_example.txt : It will displays only first field of each lines from /etc/passwd file using the field delimiter : (colon). In this case, the 1st field is the username. The file

#grep "/bin/bash" /etc/passwd | cut -d':' -f1,6: It will displays username and home directory of users who has the login shell as "/bin/bash".

#grep "/bin/bash" /etc/passwd | cut -d':' -f1-4,6,7: It will displays range of fields specify start field(we are selecting field 1 through 4) and end field (6 and 7) who has the login shell as "/bin/bash".

---

rsync: rsync stands for remote sync.
rsync utility is used to synchronize the files and directories from one location to another in an effective way. Backup location could be on local server or on remote server.

Syntax : rsync options source destination

Source and destination could be either local or remote. In case of remote, specify the login name, remote server name and location.

Options may be: 1) z ---> It is to enable compression
2) v ----> Verbose
3) r---- > Indicates recursive
4) a ----> Archive Mode
-a option does the following,
- Recursive mode
- Preserves symbolic links
- Preserves permissions
- Preserves timestamp
- Preserves owner and group

5) u ----> Do not overwrite the Modified Files at the Destination
6) d ----> Synchronize only the Directory Tree Structure (not the files).
7) i ----> Displays the item changes from Source and Destination
8) W---- >
9) e ssh ---> It will enable the secured remote connection.
9) --progress---- > View the rsync Progress during Transfer
10) -- delete---- > Delete the Files Created at the Target
11) --existing --- > Update (Sync) only the existing files at the target
12) --include and --exclude ----> Include and exclude files or directories while doing synchronization.
13) –max-size--- > File Size.

Examples:
========
#rsync -azvr /home/simon/scripts/Java/Source/ /home/simon/scripts/Java/Target/

This command will sync the Source and Target directories.

#rsync -azvr /home/simon/scripts/Java/Source/ root@lexbz2133.cloud.dst.mt.com:/opt/

This command will sync the file from local server to remote server for specified directory.

 # rsync -azvr root@lexbz2133.cloud.dst.mt.com:/opt/FTPReadXML.java /home/isgproject/ scripts/Java/Target/

This command will sync the file from remote server to local server for specified file.

I have a directory which contains files with different kinds of extensions .Everyday a file with .log gets added to it, I want to extract the file with .log extension which is created with today's date.

ls | grep -e "`date "+%Y%m%d"`" | grep ".log"

If the date (in format YYYYMMDD) is not part of the log file then,

find . -mtime -1 -name "*.log"

---

test : This command is used to see if an expression is true, and if it is true it return zero(0), otherwise returns nonzero for false.

Syntax : test << EXPRESSION >> << Options >>

Assume here a varibale file have file name "/home/simon/scripts/ControlStatements/test.sh" whose size 775 Bytes and have read, write and execute permissions.

```
#!/bin/sh

file="/home/simon/scripts/ControlStatements/test.sh"

if test -r $file
then
   echo "File has read access"
else
   echo "File does not have read access"
fi

if test -w $file
then
   echo "File has write permission"
else
   echo "File does not have write permission"
fi

if test -x $file
```

```
then
   echo "File has execute permission"
else
   echo "File does not have execute permission"
fi

if test -f $file
then
   echo "File is an ordinary file"
else
   echo "This is sepcial file"
fi

if test -d $file
then
   echo "File is a directory"
else
   echo "This is not a directory"
fi

if test -s $file
then
   echo "File size is not zero"
else
   echo "File size is  zero"
fi

if test -e $file
then
   echo "File exists"
else
   echo "File does not exist"
```

Run the above script as follows.

```
#sh test.sh
```

It produces the below output.

```
File has read access
File has write permission
File has execute permission
File is an ordinary file
This is not a directory
File size is not zero
File exists
```

```
if test -f /usr/bin/java
then
        echo "Java exists in /usr/bin/"
fi
```

Using Square brackets

Square brackets can be used to replace the test command in most command shells. Here is an example of this. Note that the spaces between the brackets and the test operation are vital here.

```
if [ -f /usr/bin/java ]
then
        echo "Java exists in /usr/bin/"
fi
```

---

expr : The expr command reads the Expression parameter, evaluates it, and writes the result to standard output.

Syntax : expr << Argument 1 >> << Operator >> << Argument 2 >>

Examples :

```
#expr 2 + 2
#expr 2 - 2
#expr 2 \* 2
#expr 2 / 2

#expr 2 = 2
#expr 2 = 3
#expr 2 \< 2
#expr 2 \< 3
#expr 2 \> 2
#expr 2 \> 1
#expr 2 \! 1
#expr 2 \! 2
#expr 2 != 2

#expr Bhaskar : Bhaskar ---> Partial match and returns the number of characters matched.
#expr Bhaskar : Bhaskar
#expr Bhaskar : Bhas
#expr Bhaskar : '.*' ---> Regular expression to match any number of characters
#expr Bhaskar : '....\(...\)' ---> To print the matched characters instead of number of matching positions.
```

expr `echo "123671" | sed -e 's/[0-9]/ + &/g' -e 's/^ +//g'` : It will give sum of digits of a number.

How to find Apache version?

Ans)
find / -name apachectl
#cd /opt/MT/HTTPServer/bin/apachectl
./apachectl –v

Server version:MT_HTTP_Server/7.0.0.25 (Unix)
Server built: Jun 21 2012 19:26:56

How to run last executed find command in Linux?

Ans) !find will repeat the last find command executed. In fact "!" can be used with any command to invoke previous run of that command.

Managing Bash with Key Sequences

Ctrl+A : Move cursor to the beginning of the command line.

Ctrl+C : This key sequence is used to terminate the command that's running in your shell.

Ctrl+D : This key sequence is used to send the "eod of the file" (EOF) signal to the command. We can also use this key sequence to close the shell session.

Ctrl+E : Move cursor to the end of the command line.
Ctrl+H : Generate backspace character.

Ctrl+L : Clear this terminal.

Ctrl+R : This is the reversed search feature. It will open the "reverse I-search" prompt, which helps you locate commands that you used previously. This key sequence searches the bash history. This feature is useful when working with longer commands.
longer commands.

Ctrl+Z : This key sequence is used to suspend a process/command/job that is running interactively on the console.

We can start these jobs again with the **fg** and **bg** commands. To start the command again, we need to see the job number that command/job using. We can see the list of job numbers with **jobs** command.

Ctrl+U : Cut/Copy line.

Ctrl+Y : Paste whatis cut/copied.

Ctrl+W : Delete text after cursor.

Ctrl+K : Delete text before cursor.

Shift+**PageUp** and
Shift+**PageDown** Browse terminal buffer (to see text that has "scrolled off" the screen).

---

ping: It is a network tool that tests of whether a particular host is up and reachable on the network.

The default behavior of ping is to send a 64 byte ICMP (Internet Control Messaging Protocol) packet to the specified host every second until you cancel the operation with a Ctrl+c. When the ping operation is cancelled or done, ping will report summary statistics such as average pocket loss, number of pockets sent/received, etc.

#ping ralbz2072.cloud.dst.mt.com

Here I have canceled after 6 intervals.
The -c option allows us to specify a count of the number of ICMP packets ping should send before simply ending.

#ping –c 4 ralbz2072.cloud.dst.mt.com

The -a option will gives the audible ping.
#ping -a -c 5 devpcpdgdst02.w3-969.mt.com

```
Bhaskars-MacBook:~ BhaskarReddy$ ping -a -c 5 devpcpdgdst02.w3-969.ibm.com
PING devpcpdgdst02.podc.sl.dst.ibm.com (9.16.170.196): 56 data bytes
64 bytes from 9.16.170.196: icmp_seq=0 ttl=47 time=332.421 ms
64 bytes from 9.16.170.196: icmp_seq=1 ttl=47 time=295.101 ms
64 bytes from 9.16.170.196: icmp_seq=2 ttl=47 time=300.512 ms
64 bytes from 9.16.170.196: icmp_seq=3 ttl=47 time=353.802 ms
64 bytes from 9.16.170.196: icmp_seq=4 ttl=47 time=393.796 ms

--- devpcpdgdst02.podc.sl.dst.ibm.com ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 295.101/335.126/393.796/36.327 ms
Bhaskars-MacBook:~ BhaskarReddy$
```

With -i option we set the interval seconds between sending each packet. The default is to wait for one second between each packet normally.

#ping -i 3 -c 5 devpcpdgdst02.w3-969.mt.com

---

traceroute:
 # traceroute b03ziwas11c.boulder.mt.com

How to find weather system type is 32 bit or 64 bit?
Ans) Below command will give the weather system type is 32 bit or 64 bit
#**getconf LONG_BIT**

How to change the prompt disply?
Ans) Add the below lines in .bashrc (BASH) or .kshrc (KSH) in respective home directory.

PS1='$PWD $'
export PS1

---

sort: Sort allows us to sort data in numeric, alphabetic and reverse order whether it's of column or multiple columns with delimiters. Sort order can be restricted to individual columns and then merged to one single file. Sorting becomes handy in many situations.

#cat names.txt

#sort names.txt

#cat numbers.txt

#sort numbers.txt

#sort -n numbers.txt
#sort -nr numbers.txt

---

https://www.geeksforgeeks.org/sort-command-linuxunix-examples/