Linux Commands for User

General Commands Not Used Frequently

cal – The Calender

bc – The Calculator

file – knowing the file types

cmp – Comparing two files.

comm – Common in two files

dos2unix and unix2dos – Converting between dos and unix

stat - Display file or file system status such as Access, Modify,Change, users and more

stat <fileName> unix Command

General Commands Frequently Used

Script – recording your session

Passwd – change your password

Echo - Displaying the message

Printf - Displaying the message

who – Who are the Users

who am I – To know my user name

uname – Knowing your machine characteristics

uname –a - To Print all info.

tty – knowing your terminal

pwd – Checking Current Directory

cd - changing the current Directory

mkdir – Making directories

rmdir – removing directories

ls – Listing Directory Contents

ls –F – Marks executables with *, directories with / and symbolic links with @ at the end of file name.

ls – a – Shows all filesnames including beginning with a dot.

ls –R – Recursive display list of files and folders

ls -1 – one filename in each line

ls –l long listing + time of last file modification

ls –t Sorts filenames by last modification time

ls –u – Sorts filenames by last access time

ls –I – Displays inode number

ls –lc – Time of last inode modifcation

Cat – Displaying and creating Files

cat filename – Displaying file contents

cat –n filename - Displaying file contents with line numbers

cat > filename – To Create a file name

More – Paging Output

more filename – To to display the filename contents

Less – Paging Output

wc – Counting lines, Words and Characters

-l – Option counts only the number of lines

-w – Option counts only the words.

-c – Options counts only the characters.

cp – copying a files

Options

-p – Copy the files including preserve the mode, ownership and permissions.

-R – Copying Directory Structure

gzip and gunzip – Compressing and Decompressing Files

gzip filename

gzip –l – To find out how much of the compression was archived.

gzip –d filename – To Decompress a file

gunzip filename – To Decompress a file

gzip –R dir – To Compress Recursive

Tar – The archival Program

-c – To create a archive

-x – Extract files from archive

-t – Display fils in archive

-v – verbose

-f – name of the arch filename .tar

tar –cvf filename.tar filename1 filename2

tar –xvf filename.tar

Create tar and gzip

tar cvf - foodir | gzip > foo.tar.gz

Un-tar and gunzip

tar -zxvf apache-activemq-5.1.0-bin.tar.gz

Un-tar and uncompress using gunzip

gunzip -c apache-activemq-5.1.0-bin.tar.gz | tar xvf -

or alternatively:

gunzip apache-activemq-5.1.0-bin.tar.gz | tar xvf -

File Attributes

chmod – Changing File Permission

Abbreviations Used by chmod

Category	Operation	Permission	Assigned Number
u -user	+ - Assigns Permission	r – Read permission	4
g -group	- - Assigns Permission	w – Write Permission	2
o – other	= - Assigns Permission	x – Execute Permission	1

a –all(ugo)			

Example:

chmod u+x filename

chmod u+x filename

chmod ugo+x filename

chmod u+x filename1 filename2 filename3

chmod a-x,go+r filename

chmod 566 filename

chmod 755 filename

chmod –R 755 .

chmod –R a+x *

Chown – Changing File Ownership

chown username filename

More example needed.

Shell's Wild-Cards

Wild-Card	Matches
*	Any numbers of characters including none
?	A single character
[ijk]	A single character – either I,j or k
[x-z]	A single character that is within the ASCII range of the characters x and z
[!ijk]	A single character that is not an I,j, or k
[!x-z]	A single character that is not within the ASCII range of the characters x and z

Ps – Process Status

Options to ps

POSIX Options	Significance
-f	Full listing showing the PPID of each process
-e or –A	All Process including user and Systems
-u usr	Process of user usr only
-a	Processes of all users excluding processes not associated with terminal
-l	A long listing showing memory-related information

Example

ps -ef

ps –u sumit

ps –a

ps –e

ps -A - ps command will report a snapshot of the current processes. To select all processes use the -A

ps -Al - Show Long Format Output

ps -AlF - To turn on extra full mode (it will show command line arguments passed to process):

ps axu - Print All Process On The Server

ps -U vivek -u vivek u - See Every Process Running As User Vivek

ps -p 55977 -o comm= - Display The Name of PID 55977

ps -auxf | sort -nr -k 4 | head -10 - Find Out The Top 10 Memory Consuming Process

ps -auxf | sort -nr -k 3 | head -10 - Find Out top 10 CPU Consuming Process

$! – Store the PID of the last background jobs

echo $$ - To Know the PID of current Shell

echo $SHELL – To Know the Current Shell

System process easily identifies the ? in the TTY coloumn.

Mechanism of Process Creation

Fork

Exec

Wait

& and nohup – Ruunning jobs in background

nohup sort emp.lst $

nice – Job execution with low priority

Kernel decides how much processor time is required for a process based on the nice value.

Possible nice value range is: -20 to 20. A process that has a nice value of -20 is very high priority.

The process that has a nice value of 20 is very low priority.

Use ps axl to display the nice value of all running process as shown below.

ps axl

nice command is used with & operator to reduce the priority of jobs

nice script.sh &

./nice-test.sh & Default the nice value of 0

nice -10 ./nice-test.sh & Nice value is 10 Low priority.

nice --10 ./nice-test.sh & Nice value is -10 High priority.

Kill – killing the process

kill PID

kill PID1 PID2 PID3

kill $i – Killing the last background process

kill –s KILL PID – recommended way for killing (SIGKILL)

kill -9 PID – same as above but not recommended. (SIGKILL)

kill -9 $$ -$$ store the PID of current shell

kill –s KILL 0 – kill all the process including the login shell.

kill –l – To View the list of all signal names and numbers that are available on your machine.

kill %1 – Kill first background jobs

Jobs

List the background jobs in following fashion

[3] + running command

[1] - running command

[2] running command

bg – Convert Jobs to background

if you have invoked a command and the prompt has not yet return, you can suspend the job by pressing Ctrl-Z. Observe that job has not been terminated yet; its onl y suspended (“stopped”). Now you can use bg command to push the current foreground job in the background.

bg %2 – Sends second job to background

bg %sort - Sends sort job to background

fg – bring background job to foreground

fg – To bring most recent job to forground

fg %1 – Bring First job to foreground

fg %2 – Bring second job to foreground

fg %sort - Bring sort job to foreground

at – On time execution

at 14:08

at > script.sh

[ctrl-d]

Batch – execute when system resources are available

batch < script.sh

Cron – Schedule and run jobs periodically

crontab –e – To Edit the cron tab

crontab –l – To Display the cron tab

crontab cron.tx – cron.txt contains cron commands

crontab –r – To Remove the cron

Format of crontab – TODO

Customizing the environment

set – set statemtent display a complete list of all environment variable

PATH =$PATH:/usr/xpg4/bin – Adding new value to old values

PS1 =”C> “ – To Change the prompt

PS1=’[$PWD] ‘ – To Change the prompt to pwd

alias cp=”cp –I” – To Set the alias in bash

history – To See the history

IFS – Field Separators for commands and arguments

!! – Repeat Previous commands

!2 – Repeat commands 2 from history output

!-2 – Execute the commands prior to previous one

!v – Exexture very last commands beginning withg v

$_ - Using last arugement of previous commands

mkdir raj

cd $_

ln – Hard Links and Softlinks

ln /usr/bin/perl /usr/local/bin/perl – To create a hard links from src to dest

ls –I - To Display the node number of files

ln –s /usr/bin/perl /usr/local/bin/perl – To Create a soft link

umask – default file and directory permission

When you create a files and directories, the permission assigned to them depends on rge system’s default setting..

unix has default 666 for regular files & 777 for directories

umask –ENTER

022

This is an octal number which has to be subtracted from system default to obtain the actual

default. This becomes 644 (666-022) for ordinary files and 755 (777 -022) for directories.

User can set the umask such as umask 023.

touch – Changing the time stamp

touch emp.lst – Create a file name called emp.lst

touch 02161430 emp.lst – To Change the time stamp of file for MMDDhhmm format

touch –a 02161430 emp.lst – To Change the access time stamp of file for MMDDhhmm format

touch –m 02161430 emp.lst – To Change the modification time stamp of file for MMDDhhmm format

find – Locating files

Expression Used by find.

Expression	Use
-inum n	Having inode number n
-type x	if of type x can include files, directories or symbolic link
-type f	If an ordinary file
-perm nnn	If octal permission match nnn completely
-links n	If having n links
-user usname	If owner by usname
-group gname	If owned by group gname
-size +x[C]	
-mtime –x	If modified in less than x days
-newer filename	If modified after filename

-mmin –x	If modified in less than x minutes
-atime +x	If accessed in more than x days
-amin +x	If accessed in more than x minutes
-name filename	Filename
Action	**Significance**
-print	Prints selected file on standard output
-ls	Executes ls –lids commands on selected files
-exec cmd	Executes UNIX command cmd followed by {} \;
-ok cmd	Like –exec, except that command is executed after user confirmation

Example

Change File Permissions Recursively

find . -type f -exec chmod 644 {} \;

find . -type d -exec chmod 755 {} \;

find . -name Configuration.php -exec chmod 666 {} \;

Find files modified in the last 48 hours, and in current folder and one level below

find -maxdepth 2 -type f -mtime -2

To find all files modified in the last 24 hours (last full day) in a particular specific directory and its sub-directories:

find /directory_path -mtime -1 –print

To find all files with regular file types only, and modified in the last 24 hours (last full day) in current

directory and its sub-directories:

find /directory_path -type f -mtime -1 –print

find . -type f -mtime -1 –print

To find all files that are modified today only (since start of day only, i.e. 12 am), in current directory and its sub-directories:

touch -t `date +%m%d0000` /tmp/$$

find /tmefndr/oravl01 -type f -newer /tmp/$$

rm /tmp/$$

To find all files in /home/user/demo directory

find /home/user/demo -type f –print

To find all files in /home/user/demo directory with permission 777, enter:

find /home/user/demo -type f -perm 777 –print

Apply new permission using the -exec option as follows:

find /home/user/demo -type f -perm 777 -print -exec chmod 755 {} \;

To select directories and subdirectories use the following syntax:

find /var/www/html -type d -perm 777 -print -exec chmod 755 {} \;

This first Linux find example searches through the root filesystem ("/") for the file named "Chapter1". If it finds the file, it prints the location to the screen.

find / -name Chapter1 -type f –print

A nice thing to know is that on Linux systems and modern Unix system you no longer need the -print option at the end of the find command, so you can issue it like this:

find / -name Chapter1 -type f

This next find command searches through the /usr and /home directories for the file named Chapter1:

find /usr /home -name Chapter1 -type f

To search in the current directory, and all subdirectories, just use the . character to reference the current directory in your find commands, like this:

find . -name Chapter1 -type f

This next command searches through the /usr directory for all files that begin with the letters Chapter, followed by anything else. The filename can end with any other combination of characters. It will match filenames such as Chapter, Chapter1, Chapter1.bad, Chapter-in-life, etc.:

find /usr -name "Chapter*" -type f

This next command searches through the /usr/local directory for files that end with the extension .html. These file locations are then printed to the screen.

find /usr/local -name "*.html" -type f

To find all directories named build under the current directory, use this command:

find . -type d -name build

This command searches through the htdocs and cgi-bin directories for files that end with the extension .cgi. When these files are found, their permission is changed to mode 755 (rwxr-xr-x). This example shows that the find command can easily search through multiple sub-directories (htdocs, cgi-bin) at one time.

find htdocs cgi-bin -name "*.cgi" -type f -exec chmod 755 {} \;

Find and display files last modified les than 90 days ago.

find . -name "*" -mtime -3 -print

find everything in your home that has been modified more recently than "abc.txt":

find $HOME -newer ~joeuser/lastbatch.txt

For finding only files from all directories recursively

find ./ -type f | wc -l

For finding only files from all directories recursively

find ./ - type d | wc -l

How to Return a message when a file is not found using find command? OR

Find command return type OR

when the file is not found i want it to return some value OR

```
# find . -name raj.txt > raj.txt
# counter= `cat abc.txt | wc -l'
# if [$counter -gt 0]
# then
#               echo "File is found"
# else
#               echo "File is not found"
# fi
```

That's nice, but what if I want to see the last modification time of these files, or their filesize? No problem, I just add the "ls -ld" command to my find command, like this:

```
find . -name "*.pl" -exec ls -ld {} \;
```

Count Total number of files in Directory and Subdirectory

```
# find . -type f | wc –l
```

Count Specific extention files in Directory and Subdirectory

```
# find . -type f -name \*.mnp |wc –l
```

Count only Directory

```
# find . -type d | wc –l
```

head – Displaying the beginning of a file

```
# head –n 3 filename
# vi `ls –t` | head –n 1` - Opens last modified file for editing
```

tail – Displaying the end of a file

 # tail -3 filename

 # tail –f filename Monitering file live

Cut

cut –c 6-12,24-32 filename – Cutting column

cut –d \| -f 2,3 filename - Cutting fields

cut –d "|" –f 1,4- filename –To cut out the fields numbered 1,4,5 and 6.

Sort – Ordering a file

Sort Options

Option	Description
-tchar	Use delimiter char to identify fields
-k n	Sorts on nth field
-k m,n	Start sort on mth field and end sort on nth field
-k m.n	Start sort on nth column of mth field
-u	Removes repeated lines
-n	Sort numerically
-r	Reverse sort order
-m list	Merge sorted files in list
-c	Checks of file sorted
-o filename	Place output in file filename

Examples:

sort –t"|" –k 2 shortlist

sort –t"|" –r –k 2 shortlist

sort –t "|" –k 3,3 –k 2,2 shortlist

sort –t"|" –k 5.7,5.8 shortlist

sort numfile

cut –d"|" –f3 filename | sort –u

sort –m foo1 foo2 foo3

Uniq – Locate repeated and non-repeated lines

sort dept.lst | uniq

cut –d"|" –f3 emp.lst | sort | uniq –u (-u selects only lines which is not repeated)

cut –d"|" –f3 emp.lst | sort | uniq –d (-d selects one copy of repeated lines)

cut –d"|" –f3 emp.lst | sort | uniq –c

tr – Translating characters

tr '|/' '~-' < emp.lst | head –n 3 – To replace the | with a ~ and the / with a -.

head –n 3 emp.lst | tr '[a-z]' '[A-Z]' – Change first three lines from lower to upper.

tr –d '|' < emp.lst | head –n 3 – To deleting characters

grep – Searching for a pattern

grep "director" filename1 filename2

grep "Rajesh Kumar" filename1

grep –i 'rajesh' filename1 To Ignore case

grep –v 'rajesh' filename1 > filename2 To Select all except lines containing patterns

grep –n 'rajesh' filename1 To Display line numbers

grep –c 'rajesh' filename Counting line containing pattern

grep –l 'rajesh' *.lst -l options display only the name of files containing pattern

grep –f pattern.lst emp.lst Taking patterns from files

Grep a file, but show several surrounding lines?

For BSD or GNU grep you can use -B num to set how many lines before the match and -A num for th number of lines after the match.

grep -B 3 -A 2 foo README.txt

If you want the same amount of lines before and after you can use -C num.

grep -C 3 foo README.txt

This will show 3 lines before and 3 lines after.

Basic Regular Expression tables

Symbols or Expressions	Matches
*	Zero or more occurrances of the previous character
g*	Nothing or g,gg,ggg, etc
.	A Single Character
.*	Nothing or any number of character
[pqr]	A single character p, q or r
[c1-c2]	A Single Character with ASCII range
[1-3]	A single digit between 1 and 3
[^pqr]	A Single character which is not a p,q or r
[^a-zA-Z]	A non-alphabetic character
^pat	Pattern pat at the beginning of the line
pat$	Pattern pat at the end of the line
bash$	Bash at the end of the line
^bash$	Bash as the only one word in line
^$	Line containing nothing
+	Matches one or more character of

	previous character
?	Matches zero or one occurrence of the previous character
|	Delimiter for multiple pattern
( )	Group pattern

Example:

grep “[aA]g[ar][ar]wal filename

grep “[aA]gg*[ar][ar]wal” filename

grep “j.*Saxena” filename

grep “^2” filename

grep “7…$” filename

grep “^[^2]” filename

grep –E “[aA]gg?arwal” filename

grep –E ‘sengupta|dasgupta’ filename

grep –E ‘(sen|das)gupta’ filename

Editor

Awk - http://www.thegeekstuff.com/2010/01/awk-introduction-tutorial-7-awk-print-examples/

Sed

vi

sed – The Stream Editor

sed ‘3q’ filename == head –n 3 Quit after line number 3

sed –n ‘1,2p’ filename Prints the first 2 lines. Must use –n with p

sed –n ‘$p’ filename print last line

sed –n ‘9,11p 7,9p $p’ filename

sed –n ‘3,$ip’ filename Don’t print line 3 to the end, display only line 1 and 2

sed –n ‘/director/p’ filename To print the lines which has pattern in filename

sed –n ‘/dasgupta/,/saksena/p’ filename

sed –n ‘1,/dasgupta/p’ filename

sed –n ‘/[aA]gg*[ar][ar]wal/p’ filename

sed –n ‘/sa[kx]s*ena/p /Gupta/p’ filename

sed –n ‘/50…..$/p’ filename

sed –n ‘director/w dlsit’ filename

sed –n ‘director/w dlist /manager/w mlist /executive/w elist’ filename

sed –n ‘1,500w foo1 501,$w foo2’ filename

sed ‘1i\

> #include <stdio.h>\

> #include<unistd.h>

>’ foo.c >> $$ Include these include on beginning of the program

sed ‘a\

‘ filename # insert after every line this blank line

sed “/director/d’ filename > filename2 -n option not to be used with d

== # grep –v “director” filename > filename2

sed –n ‘/director/!p’ filename1 > filename2

sed ‘s/|/:/’ filename | head -2

sed ‘s/|/:/g’ filename | head -2

sed ‘1,3s/|/:/g’ filename

sed ‘1,5s/director/member /’ filename

sed ‘s/^/2’ filename

sed ‘s/$.00/’ filename

Sed tables

Command	Description

l,a,c	**Insert, Appends and Changes text**
d	**Delete lines**
10q	**Quit after reading the first 10 lines**
p	**Print line on standard outputs**
3,$p	**Print lines 3 to the end. –n option is required**
$!p	**Prints all lines except last line. –n option required**
/begin/,/end/p	**Print lines enclosed between begin and end. –n option required**
q	**Quit after reading uo to the address line**

Vi - Editor

IMPORTANT – YOU SHOULD USE DOUBLE QUOTES ONLY WHEN PARAMETER EVALULATION OR command substitution is embedded within command

Write about command “w”