

# Unix Programming

## Popular Unix Commands



Nguyen Thanh Hung  
Software Engineering Department  
Hanoi University of Science and Technology





## tar command

**Create a new tar archive.**

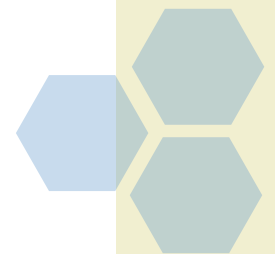
**\$ tar cvf archive\_name.tar dirname/**

**Extract from an existing tar archive.**

**\$ tar xvf archive\_name.tar**

**View an existing tar archive.**

**\$ tar tvf archive\_name.tar**





## grep command

**Search for a given string in a file (case insensitive search).**

```
$ grep -i "the" demo_file
```

**Print the matched line, along with the 3 lines after it.**

```
$ grep -A 3 -i "example" demo_text
```

**Search for a given string in all files recursively**

```
$ grep -r "ramesh" *
```





## find command

**Find files using file-name ( case in-sensitive find)**

```
# find -iname "MyCProgram.c"
```

**Execute commands on files found by the find command**

```
$ find -iname "MyCProgram.c" -exec  
md5sum {} \;
```

**Find all empty files in home directory**

```
# find ~ -empty
```





## **vim command**

**Go to the 143rd line of file**

**\$ vim +143 filename.txt**

**Go to the first match of the specified**

**\$ vim +/search-term filename.txt**

**Open the file in read only mode.**

**\$ vim -R /etc/passwd**





## diff command

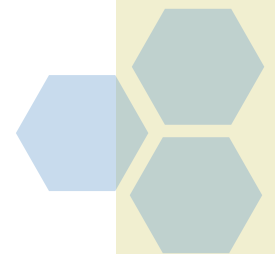
**Ignore white space while comparing.**

```
# diff -w name_list.txt name_list_new.txt
```

```
2c2,3
```

```
< John Doe --- > John M Doe
```

```
> Jason Bourne
```





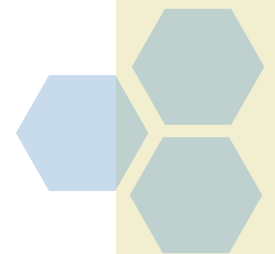
## sort command

**Sort a file in ascending order**

**\$ sort names.txt**

**Sort a file in descending order**

**\$ sort -r names.txt**





## export command

**To view oracle related environment variables.**

```
$ export | grep ORACLE
```

```
declare -x ORACLE_BASE="/u01/app/oracle"
```

```
declare -x ORACLE_HOME="/u01/app/oracle/product/10.2.0"
```

```
declare -x ORACLE_SID="med"
```

```
declare -x ORACLE_TERM="xterm"
```

**To export an environment variable:**

```
$ export ORACLE_HOME=/u01/app/oracle/product/10.2.0
```







## xargs command

**Copy all images to external hard-drive**

```
# ls *.jpg | xargs -n1 -i cp {} /external-hard-drive/directory
```

**Search all jpg images in the system and archive it.**

```
# find / -name *.jpg -type f -print | xargs tar -cvzf images.tar.gz
```

**Download all the URLs mentioned in the url-list.txt file**

```
# cat url-list.txt | xargs wget -c
```





## ls command

**Display filesize in human readable format (e.g. KB, MB etc.,)**

**\$ ls -lh**

**-rw-r----- 1 ramesh team-dev 8.9M Jun 12 15:27 arch-linux.txt.gz**

**Order Files Based on Last Modified Time (In Reverse Order) Using ls -ltr**

**\$ ls -ltr**

**Visual Classification of Files With Special Characters Using ls -F**

**\$ ls -F**





## gzip command

**To create a \*.gz compressed file:**

```
$ gzip test.txt
```

**To uncompress a \*.gz file:**

```
$ gzip -d test.txt.gz
```

**Display compression ratio of the compressed file using gzip -l**

```
$ gzip -l *.gz
```

compressed	uncompressed	ratio	uncompressed_name
23709	97975	75.8%	asp-patch-rpms.txt





## unzip command

**To extract a \*.zip compressed file:**

```
$ unzip test.zip
```

**View the contents of \*.zip file (Without unzipping it):**

```
$ unzip -l jasper.zip
```

**Archive: jasper.zip**

<b>Length</b>	<b>Date</b>	<b>Time</b>	<b>Name</b>
<b>-----</b>	<b>----</b>	<b>----</b>	<b>----</b>
<b>40995</b>	<b>11-30-98</b>	<b>23:50</b>	<b>META-INF/MANIFEST.MF</b>
<b>32169</b>	<b>08-25-98</b>	<b>21:07</b>	<b>classes_</b>
<b>15964</b>	<b>08-25-98</b>	<b>21:07</b>	<b>classes_names</b>
<b>10542</b>	<b>08-25-98</b>	<b>21:07</b>	<b>classes_ncomp</b>





## shutdown command

**Shutdown the system and turn the power off immediately.**

**# shutdown -h now**

**Shutdown the system after 10 minutes.**

**# shutdown -h +10**

**Reboot the system using shutdown command.**

**# shutdown -r now**

**Force the filesystem check during reboot.**

**# shutdown -Fr now**





## ftp command

Both ftp and secure ftp (sftp) has similar commands. To connect to a remote server and download multiple files, do the following.

```
$ ftp IP/hostname  
ftp> mget *.html
```

To view the file names located on the remote server before downloading, mls ftp command as shown below.

```
ftp> mls *.html -  
/ftptest/features.html  
/ftptest/index.html  
/ftptest/othertools.html  
/ftptest/samplereport.html  
/ftptest/usage.html
```





# crontab command

**View crontab entry for a specific user**

**# crontab -u john -l**

**Schedule a cron job every 10 minutes.**

***\*/10 \* \* \* \* /home/ramesh/check-disk-space***

**Schedule a cron job everyday at 11am and 16pm**

***00 11,16 \* \* \* /home/ramesh/bin/incremental-backup***

***00 – 0th Minute (Top of the hour)***

***11,16 – 11 AM and 4 PM***

***\* – Every day***

***\* – Every month***

***• – Every day of the week***

**Schedule a cron job everyday (including weekends) during the working hours 9 a.m – 6 p.m**

***00 09-18 \* \* \* /home/ramesh/bin/check-db-status***

***00 – 0th Minute (Top of the hour)***

***09-18 – 9 am, 10 am, 11 am, 12 am, 1 pm, 2 pm, 3 pm, 4 pm, 5 pm, 6 pm***

***\* – Every day***

***\* – Every month***

***\* – Every day of the week***

Field	Description	Allowed Value
MIN	Minute field	0 to 59
HOUR	Hour field	0 to 23
DOM	Day of Month	1-31
MON	Month field	1-12
DOW	Day Of Week	0-6
CMD	Command	Any command to be executed.





## ps command

ps command is used to display information about the processes that are running in the system.

While there are lot of arguments that could be passed to a ps command, following are some of the common ones.

To view current running processes.

```
$ ps -ef | more
```

To view current running processes in a tree structure. H option stands for process hierarchy.

```
$ ps -efH | more
```







# free command

This command is used to display the free, used, swap memory available in the system. Typical free command output. The output is displayed in bytes.

\$ free

	total	used	free	shared	buffers	cached
Mem:	3566408	1580220	1986188		0	203988
-/+ buffers/cache:		473272	3093136			
Swap:	4000176	0	4000176			

If you want to quickly check how many GB of RAM your system has use the -g option. -b option displays in bytes, -k in kilo bytes, -m in mega bytes.

\$ free -g

	total	used	free	shared	buffers	cached
Mem:	3	1	1	0	0	0
-/+ buffers/cache:		0	2			
Swap:	3	0	3			

If you want to see a total memory ( including the swap), use the -t switch, which will display a total line as shown below.

ramesh@ramesh-laptop:~\$ free -t

	total	used	free	shared	buffers	cached
Mem:	3566408	1592148	1974260		0	204260
-/+ buffers/cache:		475332	3091076			
Swap:	4000176	0	4000176			
Total:	7566584	1592148	5974436			





## top command

top command displays the top processes in the system ( by default sorted by cpu usage ). To sort top output by any column, Press O (upper-case O) , which will display all the possible columns that you can sort by as shown below.

Current Sort Field: P for window 1:Def

Select sort field via field letter, type any other key to return

a: PID	= Process Id	v: nDRT	= Dirty Pages
count			
d: UID	= User Id	y: WCHAN	= Sleeping in
Function			
e: USER	= User Name	z: Flags	= Task Flags
.....			

To displays only the processes that belong to a particular user use -u option. The following will show only the top processes that belongs to oracle user.

```
$ top -u oracle
```





## df command

Displays the file system disk space usage. By default df -k displays output in bytes.

```
$ df -k
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/sda1	29530400	3233104	24797232	12%	/
/dev/sda2	120367992	50171596	64082060	44%	/home

df -h displays output in human readable form. i.e size will be displayed in GB's.

```
ramesh@ramesh-laptop:~$ df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda1	29G	3.1G	24G	12%	/
/dev/sda2	115G	48G	62G	44%	/home

Use -T option to display what type of file system.

```
ramesh@ramesh-laptop:~$ df -T
```

Filesystem	Type	1K-blocks	Used	Available	Use%	Mounted on
/dev/sda1	ext4	29530400	3233120	24797216	12%	/
/dev/sda2	ext4	120367992	50171596	64082060	44%	/home





## kill command

Use kill command to terminate a process. First get the process id using ps -ef command, then use kill -9 to kill the running Linux process as shown below. You can also use killall, pkill, xkill to terminate a unix process.

```
$ ps -ef | grep vim
```

```
ramesh  7243  7222  9 22:43 pts/2    00:00:00 vim
```

```
$ kill -9 7243
```





## rm command

Get confirmation before removing the file.

```
$ rm -i filename.txt
```

It is very useful while giving shell metacharacters in the file name argument.

Print the filename and get confirmation before removing the file.

```
$ rm -i file*
```

Following example recursively removes all files and directories under the example directory. This also removes the example directory itself.

```
$ rm -r example
```





## **cp command**

**Copy file1 to file2 preserving the mode, ownership and timestamp.**

**\$ cp -p file1 file2**

**Copy file1 to file2. if file2 exists prompt for confirmation before overwriting it.**

**\$ cp -i file1 file2**





## **mv command**

**Rename file1 to file2. if file2 exists prompt for confirmation before overwriting it.**

```
$ mv -i file1 file2
```

**Note: mv -f is just the opposite, which will overwrite file2 without prompting.**

**mv -v will print what is happening during file rename, which is useful while specifying shell metacharacters in the file name argument.**

```
$ mv -v file1 file2
```





# cat command

You can view multiple files at the same time. Following example prints the content of file1 followed by file2 to stdout.

```
$ cat file1 file2
```

While displaying the file, following cat -n command will prepend the line number to each line of the output.

```
$ cat -n /etc/logrotate.conf
```

```
1    /var/log/btmp {
2        missingok
3        monthly
4        create 0660 root utmp
5        rotate 1
6    }
```







## mount command

To mount a file system, you should first create a directory and mount it as shown below.

```
# mkdir /u01
```

```
# mount /dev/sdb1 /u01
```

You can also add this to the fstab for automatic mounting. i.e Anytime system is restarted, the filesystem will be mounted.

```
/dev/sdb1 /u01 ext2 defaults 0 2
```





## chmod command

chmod command is used to change the permissions for a file or directory.

Give full access to user and group (i.e read, write and execute ) on a specific file.

```
$ chmod ug+rxw file.txt
```

Revoke all access for the group (i.e read, write and execute ) on a specific file.

```
$ chmod g-rwx file.txt
```

Apply the file permissions recursively to all the files in the sub-directories.

```
$ chmod -R ug+rxw file.txt
```





## passwd command

Change your password from command line using passwd. This will prompt for the old password followed by the new password.

```
$ passwd
```

Super user can use passwd command to reset others password. This will not prompt for current password of the user.

```
# passwd USERNAME
```

Remove password for a specific user. Root user can disable password for a specific user. Once the password is disabled, the user can login without entering the password.

```
# passwd -d USERNAME
```





## **mkdir command**

**Following example creates a directory called temp under your home directory.**

```
$ mkdir ~/temp
```

**Create nested directories using one mkdir command. If any of these directories exist already, it will not display any error. If any of these directories doesn't exist, it will create them.**

```
$ mkdir -p dir1/dir2/dir3/dir4/
```





## **ifconfig command**

**Use ifconfig command to view or configure a network interface on the Linux system.**

**View all the interfaces along with status.**

**\$ ifconfig -a**

**Start or stop a specific interface using up and down command as shown below.**

**\$ ifconfig eth0 up**

**\$ ifconfig eth0 down**





## uname command

Uname command displays important information about the system such as — Kernel name, Host name, Kernel release number, Processor type, etc.,

Sample uname output from a Ubuntu laptop is shown below.

```
$ uname -a
```

```
Linux john-laptop 2.6.32-24-generic #41-Ubuntu SMP Thu Aug 19 01:12:52 UTC 2010 i686 GNU/Linux
```





## whatis command

Whatis command displays a single line description about a command.

```
$ whatis ls
```

```
ls                (1) - list directory contents
```

```
$ whatis ifconfig
```

```
ifconfig (8)      - configure a network interface
```





## locate command

Using locate command you can quickly search for the location of a specific file (or group of files). Locate command uses the database created by updatedb.

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

```
$ locate crontab  
/etc/anacrontab  
/etc/crontab  
/usr/bin/crontab  
/usr/share/doc/cron/examples/crontab2english.pl.gz  
/usr/share/man/man1/crontab.1.gz  
/usr/share/man/man5/anacrontab.5.gz  
/usr/share/man/man5/crontab.5.gz  
/usr/share/vim/vim72/syntax/crontab.vim
```







# man command

Display the man page of a specific command.

```
$ man crontab
```

When a man page for a command is located under more than one section, you can view the man page for that command from a specific section as shown below.

```
$ man SECTION-NUMBER commandname
```

```
$ whatis crontab
```

```
crontab (1)      - maintain crontab files for individual users (V3)
crontab (5)      - tables for driving cron
```

```
$ man 5 crontab
```





## **tail command**

**Print the last 10 lines of a file by default.**

**\$ tail filename.txt**

**Print N number of lines from the file named filename.txt**

**\$ tail -n N filename.txt**

**View the content of the file in real time using tail -f. This is useful to view the log files, that keeps growing. The command can be terminated using CTRL-C.**

**\$ tail -f log-file**





## **less command**

**less is very efficient while viewing huge log files, as it doesn't need to load the full file while opening.**

**\$ less huge-log-file.log**

**Once you open a file using less command, following two keys are very helpful.**

**CTRL+F – forward one window**

**CTRL+B – backward one window**





## su command

Switch to a different user account using su command. Super user can switch to any other user without entering their password.

```
$ su - USERNAME
```

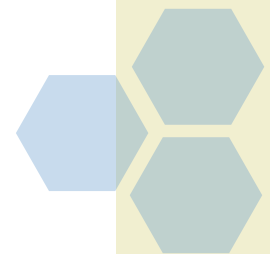
Execute a single command from a different account name. In the following example, john can execute the ls command as raj username. Once the command is executed, it will come back to john's account.

```
[john@dev-server]$ su - raj -c 'ls'
```

```
[john@dev-server]$
```

Login to a specified user account, and execute the specified shell instead of the default shell.

```
$ su -s 'SHELLNAME' USERNAME
```





## mysql command

mysql is probably the most widely used open source database on Linux. Even if you don't run a mysql database on your server, you might end-up using the mysql command ( client ) to connect to a mysql database running on the remote server.

To connect to a remote mysql database. This will prompt for a password.

```
$ mysql -u root -p -h 192.168.1.2
```

To connect to a local mysql database.

```
$ mysql -u root -p
```

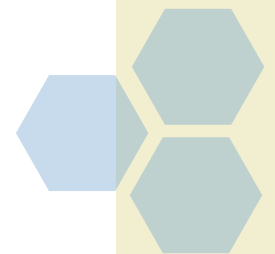
If you want to specify the mysql root password in the command line itself, enter it immediately after -p (without any space).





## ping command

**Ping a remote host by sending only 5 packets.**  
**\$ ping -c 5 gmail.com**





## wget command

The quick and effective method to download software, music, video from internet is using wget command.

```
$ wget
```

```
http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-3.2.1.tar.gz
```

Download and store it with a different name.

```
$ wget -O taglist.zip http://www.vim.org/scripts/download_script.php?src_id=7701
```

