UNIX commands

1. To find files bigger than a particular size say 4096 bytes

```
find . -type f -size +4096c
```

2. To find files smaller than a particular size say 4096 bytes.

```
find . -type f -size -4096c
```

3. Deleting files exceeding a particular size say 100kbytes.

```
find . -size +100k -exec rm{}\
```

here exec rm{}\ invokes the given command on every found file.

\ marks end of the command

{} placeholder for current file name, including path.

4. List top 10 largest files from current directory.

```
du .|sort -nr|head -n10
```

5. List 10 largest directories from current directory.

```
du -s *|sort -nr|head -n10
```

6. To delete files in current directory including its sub-directories.

```
rm -r*
```

7. To create thread dump for a particular process id.

```
kill -3 pid
```

8. List only invisible files in current directory.

```
find . -type f -iname “.*” -ls
```

Similarly for invisible directories, type

```
find . -type d -iname “.*” -ls
```

9. To get file type information.

```
ls -aF
```

10. What is . and ..

These are hidden directories present in all directories.

11. PS1 variable.

The characters that the shell displayes as your command prompt are stored in the variable PS1.

```
$PS1= '##>'
```

Your promt would become ##>

To set the value of PS1 so that it shows the current working directory.

```
PS1= "[\u@\h\w]\$"
```

12. To find files starting with some name say paris and ending with another name say london.

```
find . -name paris\*london.
```

13. To get the pathnames of all files and directories in the current directory and all subdirectories.

```
find . -print
```

```
find -print
```

```
find .
```

14. To find out which of the items are files and which are directories, specify -F option to ls command.

```
ls -F
```

15. List all directory entries in current directory.

```
ls -ld */
```

16. Find a file named master in current directory.

```
find . -type f -name master
```

17. Find a file named master in whole system.

```
find / -type f -name master
```

18. Find file/directory named master in whole system.

```
find / -name master
```

19. To search a file inside current directory only and not inside folders/subfolders

```
find -maxdepth 1 -type f -name master
```

If you want to search all files inside current directory only and not inside folders/subfolders then simply type

```
find -maxdepth 1 -type f
```

20. List all the files in the current directory.

```
find -type f
```

21. List all the files in the system.

```
find / -type f
```

22. List all the directories in the system.

```
find / -type d
```

23. Count the number of files having filename as master.

```
find / -name master -type f|wc -l
```

24. Replace a word by another in a file

```
sed 's/unix/linux/' file.txt
```

25. Replace the 2nd occurrence of a pattern in a file.

```
sed 's/unix/linux/2' file.txt
```

26. From 3rd occurrence, replace all the occurrence of unix with linux.

```
Sed 's/unix/linux/3g' file.txt
```

27. Replace the pattern in a specific line number.

```
Sed '4 s/unix/linux/' file.txt
```

28. Delete the lines by specifying line number.

```
sed '2d' file.txt
```

Note that this deletion does not take place in the original file. Specifying -i parameter tells sed to make the change in the original file.

```
sed -i '2d' file.txt
```

29. Delete everything between 4th and last line

```
sed '4,$d' file.txt
```

30. To run multiple sed commands in a single sed command, specify -e.

```
sed -e 's/unix/linux/' -e 's/os/system' file.txt
```

This will not make the change in the original file.To make the changes in original file specify -i parameter.

31. To display first 3 lines from a file

```
head -3 file.txt
```

32. To see jobs and process number

```
ps -f
```

33. Replace a pattern with another in current directory and all sub directories.

```
find . -type f -exec sed -i 's/unix/linux/' {} +
```

34. Replace a pattern with another in all files in current directory only.

```
sed -i 's/unix/linux/' *
```

Note that sed can also be used to search pattern in a file. For this, type

```
sed -n '/new/p' filename
```

new is the pattern that has to be searched.

-n prevents normal printing of matched lines.

35. Converts all lowercase names to uppercase

```
rename 'y/a-z/A-Z/' *
```

36. Converts all uppercase names to lowercase

```
rename 'y/A-Z/a-z/' *
```

37. To know whether your system is 32 bit or 64 bit

```
uname -a
```

38. To search pattern in all files in current directory.

```
grep 'pattern' *
```

39. To search recursively through an entire directory tree i.e to search all files in current directory and in all sub-directories.

```
grep -r 'pattern' *
```

Use -l to only print filenames of matching files and not the matching lines.

Use -n to print line number of matching lines.

Use -i for case insensitive search

Use -c to count the number of matching lines.

Use -v to look for lines that donot match pattern.

Use -H to print the filename as well as the pattern.

40. To search pattern in whole system

```
grep -r 'pattern' /
```

41. To search lines with unix, followed by zero or more other charachters, then followed by aug.

```
grep unix.*aug filename
```

42. If a file contains say four columns and you want to get only 2nd column, then use cut command.

```
1994    abcd    action  true lies

2004    efgh    adven   poc

2003    ijkl    comed   kindergarten
```

cut -c 9-13 filename

```
abcd

efgh

ijkl
```

use -c to print characters range

use -f print field range

43. cat -n filename

1. myfile
2. hi..!I have created this file
3. through internet
4.

You can ask cat to skip numbering blank lines using -b option.

Cat -b filename

1. myfile
2. hi..!I have created this file.

3. Through internet