# 1. Unix commands

Various useful unix-commands are listed in this chapter. Also, some of the examples of these commands are shown in the last section. Further, in next chapter, shell-scripting is discussed.

## 1.1. Unix commands cheatsheet

Following are the list of various useful unix-commands,

### 1.1.1. Basic file commands

| Commands | Details |
| --- | --- |
| **Basic opeations** | |
| ls | directory listing |
| ls -a | show hidden file |
| ls -l | show details for file |
| ln file1 file2 | create file2 (linked with file1). Changes will appear in both file |
| ln -s file1 file2 | create shortcut of file1 as file2 (soft link) |
| mkdir dirname | create directory dirname |
| cd dirname | go to dirname |
| cd | go to home directory |
| pwd | present working directory |
| rm filename | remove (delete) filename |
| rm -r dirname | remove directory dirname |
| rm -f filename | Force remove file filename |
| rm -rf dirname | Force remove directory dirname |
| cp file1 file2 | copy file1 to file2 |
| cp -r dir1 dir2 | copy dir1 to dir2 |
| mv file1 file2 | rename file1 to file2 |

| Commands | Details |
|---|---|
| touch filename | create filename (if not exist) |
| cat > filename | add content to file (ctrl-c to exit) |
| cat >> filename | add content at the end of the filename |
| cat filename | show content of filename (all at once) |
| more filename | show content of filename (fit to screen with more option) |
| head filename | show first 10 lines of filename |
| tail filename | show last 10 lines of filename |
| tail -f filename | last 10 lines of the filename (useful for log files) |
| command > filename | write output of command in the filename (overwrite the fie |
| command >> filname | append the output at the end of file name |
| wc filename (wc -l, -w, -c) | returns number of lines, word, character and filename. |
| chmod 777 filename | read(4), write(2) and execute(1) permission to all [4+2+1=7]<br><br>chmod [owner, group, others] filename |

## 1.1.2. Sort

| Commands | Details |
|---|---|
| **Sort** | |
| sort filename | sort the lines of the filename |
| sort -u filename | sort and eliminate duplicates |
| sort -r filename | reverse order sort |
| sort file1 -o file2<br><br>sort file1 > file2 | sort data of file1 and save to file2 |
| sort -n filename | sort data based on numeric value i.e. 12 > 4 |
| sort -M fielname | sort data with month name i.e. Jan, Feb, Mar etc. |

## 1.1.3. Cut

| Commands | Details |
|---|---|
| **Cut** | |

| Commands | Details |
| --- | --- |
| cut d"-" 1,4 filename | cut the 1st and 4th column of filename, where - is delimiter |
| cut -c3- filename | remove first 2 character from each line and printed the rest |
| cut -c-3-8 filename | print character 3-8 from each line |
| cut -c1-3,6-8,10- filename | print character 1-3, skip 4, print 6-8, skip 9, print 10 to end from each line |
| cut -d", -f3 filename | extact field-3 from each line with ',' as delimiter |
| cut -d", -f3,6 filename | extact field-3-to-6 from each line with ',' as delimiter |

## 1.1.4. Paste

| Commands | Details |
| --- | --- |
| **Paste** | |
| paste file1 file2 file3 | paste the lines of file2, then file3 beside the lines of file1 |
| paste -d", file1 file2 | put comma at the end of each file content |
| paste -s filename | merge all lines of filename |

## 1.1.5. Grep

| Commands | Details |
| --- | --- |
| **grep** | |
| grep pattern filenames | print the lines with pattern in filenames |
| grep '[0-9]' filenames | print the lines from filenames which have numbers 0-9 (regular expression) |
| grep -v pattern filenames | print all lines which do not contain pattern |
| grep -l pattern filenames | print the filenames which have pattern |
| grep -n pattern filenames | print the line numbers as well |

## 1.1.6. Sed

| Commands | Details |
| --- | --- |
| **sed** | |
| sed 's/word1/word2/' filename | replace word1 with word2 in filename |
| sed 's/word1/word2/' file1 > file2<br><br>mv file2 > file1 | use two steps to make change permanent |

| Commands | Details |
| --- | --- |
| sed -n '2,4p' filename | print line 2 to 4 of filename |
| sed -n /word1/p' filename | print lines which contain word1 in filename |

# 1.2. Multiple commands

## 1.2.1. Pipes

Pipes are used for sending output of one command to next command.

```
# total number of files in the directory
$ ls | wc -l
1

# word counts for file testfile.txt
$ cat testfile.txt | wc
      8      37     188
```

## 1.2.2. Run individual commands (;)

Commands can be run separately from one line by separating them using semicolon,

```
$ date; ls
Thu Feb 23 15:24:09 NZDT 2017
testfile.txt
```

# 1.3. Examples

## 1.3.1. File examples

```
$ ls
$ touch testfile.txt # create testfile.txt
$ ls
testfile.txt

$ cat testfile.txt

$ cal > testfile.txt  # write the calender to testfile.txt
$ cat testfile.txt
    February 2017
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28

$ date >> testfile.txt  # append date at the end of testfile.txt
$ cat testfile.txt
    February 2017
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28

Thu Feb 23 14:47:51 NZDT 2017

$ date > testfile.txt  # replace all the content of testfile with date
$ cat testfile.txt
Thu Feb 23 14:48:09 NZDT 2017

$ cal > testfile.txt  # write calender to file
$ wc testfile.txt     # count lines, word and character in testfile.txt
   8  37 188 testfile.txt

# other options for wc
$ wc -l testfile.txt  # lines
8 testfile.txt
$ wc -w testfile.txt  # words
37 testfile.txt
$ wc -c testfile.txt  # characters
188 testfile.txt
```

### 1.3.2. Sort example

```
# create a file
$ cat > test
Tiger
Lion
Cat
Eagle
^C

# see the content
$ cat test
Tiger
Lion
Cat
Eagle

# sort the content
$ sort test
Cat
Eagle
Lion
Tiger
dhriti@meher ~/Desktop $
```

### 1.3.3. Cut command

```
$ cat > test
first last age id
Meher Patel 20 101
Krishna patel 30 102
Tom Jerry 43 103
^C

$ cut -d" " -f 1,3,4 test
first age id
Meher 20 101
Krishna 30 102
Tom 43 103
```