passwd -> changing the user password
hostname -> prints the hostname on the terminal (mine is babe)
arch -> prints the system architecture (machine hardware name) (64 bit, 32 bit)
uname -> print system information
uname -m -> same as arch cmd

uptime -> print how long the system is running
whoami -> print the username of the current user
who -> print the logged in user details
id -> print user and group ids
last -> print info about last logged in users
w -> show who is logged on and what they are doing

mkdir [dir name] -> create a new folder

cal -> calendar of current month
cal 2000 -> calendar of year 2000

time -> run programs and summarize system resources usage
sleep 5 -> delay a program for 5 seconds

pwd -> prints the current working directory
echo $PWD -> prints the current working directory
echo $SHELL -> prints the bash directory
rev -> reverse words
ls -> list directory content(files and folders without files start with .)
ls -al -> list all files and folders in the direcory
ls -all -> list all files and folders with all informaion in the directory
ls | more -> list few files and directories and let show the rest of files and dirs by pressing ENTER
rm [filename] -> remove a file
rm -r [foldername] -> remove a folder

cat [textfile] -> print the content of the file
more [textfile] -> print a part of the file and let see the rest by pressing enter
less [textfile] -> same as more
cat [textfile] | grep [keyword] -> prints only the text lines which has the keyword
cat [textfile] | more -> same thing happens
head -25 [textfile] -> prints the first 25 lines of the file (first 20 lines by default)
tail -15 [textfile] -> prints the last 15 lines of the file (last 10 by default)
tail -f [textfile] -> prints real time updates also
touch [filename] -> make a new file

diff [file1] [file2] -> compare 2 files
wc [file] -> output the byte count, wordcount and the line count(newline count) of the file

jobs - shows a list of applications runs from the terminal

kill %[job_number] -> terminate the application

ps -> show running processes

ps -ef -> show all running processes

ps -awx -> modified cmd to see all processes

top -> dynamic rel time view of a running system (linux processes)

alias -> list all the defined alias

alias dir='ls -l' -> make alias for ls -l cmd

set -> list all setted environment variables

export [variable_name]=[value] -> set a new variable or overwrite a variable

nano -> opens a command line text editor

nano [file name] -> open the file in the terminal (it's editable)

history -> list of executed commands in the terminal

!! -> execute the previous executed cmd again

!182 -> execute the 182nd command executed in the terminal

!ec -> execute previous executed command begins with 'ec'

cp [source file] [destination file or folder] -> copy and paste a file

cp -i [source file] [destination file or folder] -> prompts before overwriting an existing file

cp -r [source file] [destination file or folder] -> recursively copies the entire directory structure to and past it in the destination

mv [file name] [new file name] -> rename the file

mv [source file] [destination folder or file] -> cut and paste (move) the file

file [file_name or directory_name] -> prints the file or directory type (such as textfile, presentation, etc)

which [cmd_name] -> print the path to the file that contains the specified commands

whereis [cmd_name] -> same as which but searches everywhere to find the file contains the specified command and give us where its source/config files are and where the man file is

apropos [cmd] -> print the manual page(s) name and description

man -k [cmd] -> equals to apropos

info [cmd] -> prints the infomation document of the command

ln [filename] [directory_name] ->create a hard link. then in the mentioned directory contains a shortcut to mentioned file

ln -s [filename1] [filename2] -> create a soft (symbolic) link. then the filename2 file is a shortcut for filename1 file

ls command's options

```
-i -> prints the index number of each file
-R -> list subdirectories and subfiles also recurcivly
-S -> sort by file size (largest first)
-t -> sorted by modification time
-r -> reverse order while sorting
-F -> append the indicator (ex- / is appended for directries)
```

ls command using regex

```
ls *sh - files and directories ends with 'sh'
ls *[sh] - files and directories ends with 's' or 'h'
ls [a-c]* - files and dirs starts with 'a' or 'b' or 'c'
ls b?c* - first letter should be 'b', second can be any one character,
third should be c
ls *.ppt -o *.pdf - last characters can be .ppt or .pdf
```

text file processing

```
tr -sc 'A-Za-z' '\012' < tomsawyer.txt -> list all word in a txt file
tr -sc 'A-Za-z- '\012' < [txtfile] | sort | uniq -c -> list distinct
words in a text file with count
sed 5q < [txtfile] -> prints first 5 line of the txtfile
tr -sc 'A-Za-z' '\012' < tomsawyer.txt | sed 5q -> list first 5 lines of
the output of tr -sc 'A-Za-z' '\012' < tomsawyer.txt command

tr ’a-z’ ’A-Z’ < genesis | tr -sc ’A-Z’ ’\012’ | sort | uniq –c -> make
all chars capital and list distict words with count (marging simple and
capital words)
```

Grep command with regex and options

```
grep gh     -> find lines containing ‘‘gh’’
grep ’ˆcon’ -> find lines beginning with ‘‘con’’
grep ’ing$’ -> find lines ending with ‘‘ing’’
grep –v gh  -> delete lines containing ‘‘gh’’
grep ’[A–Z]’ -> lines with an uppercase char
grep –i ’[ˆaeiou]$’ -> lines ending with a non-vowel
grep –i ’[aeiou].*[aeiou]’ -> lines with two or more vowels
grep –i ’ˆ[ˆaeiou]*[aeiou][ˆaeiou]*$’ -> lines with exactly one vowel
```

Regular Expressions

```
a     -> match the letter ‘‘a’’
[a–z] -> match any lowercase letter
```

```
[A–Z] -> match any uppercase letter
[0–9] -> match any digit
[0123456789]  -> match any digit
[aeiouAEIUO]  -> match any vowel
[ˆaeiouAEIOU] ->match any letter but a vowel
.     -> match any character
ˆ     -> beginning of line
$     -> end of line
x*    -> any number of x
x+    -> one or more of x (egrep only)
x|y   ->x or y (egrep only)
(x) override precedence rules (egrep only)
ls -l -> can view the list of files and directories with thier permisions
```

chmod u=rwx [filename] -> give the user to read write and execute permision for the file

chmod g=rwx,o=rx [filename]

chmod 777 -> give all (user, group and other) the permision to read, write and execute.

first 7 for user, second is for group and the third 7 is for others. 7 = 4+2+1. 4 is for read permision. 2 is for write permision and 1 is for execute permision.