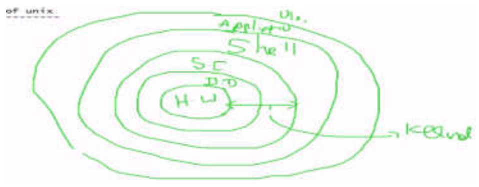
Concept	**Description**
Two types of Software	1. System software eg: Windows / Linux 2. Application software depends on system software eg: Web application
Two types of System Software	1. Command user interface CUI 2. Graphical user interface GUI
Command user interface CUI	Single user OS = MS -DOS Multi user OS = unix, solaris, hp ux, IBM AIX
Graphical user interface GUI	Single user OS = Windows 97/98 Multi user OS = Windows 2008 …. Unix, solaris, hp ux, IBM AIX
History	1969 AT & T bell labs designed a MULTICS (Multiplexed computing system) Aim to develop multi user OS. Work for 2 users developed UNICS (Uniplexed information computing system) for 100 users 1972, C language was introduced by Dennis retchie and modified the UNICS source code 1973, they renamed as UNIX, it is open source Solaris was designed by Sun microsystems using unix source code Linux was designed by Red hat AIX was designed by IBM using unix source code
Compare with windows	Unix is highly secured and virus free OS. It is stable OS. Because its performance stays stable Open source code
Architecture	H/W - hardware DD - Device drivers KERNEL -- Mouse/Keyboard drivers connected to OS SC - system calls KERNEL -- Kernel understands machine language 0/1 shell -- checks whatever the user type as commands, if it is valid pass to OS if not then display command not found. Application User

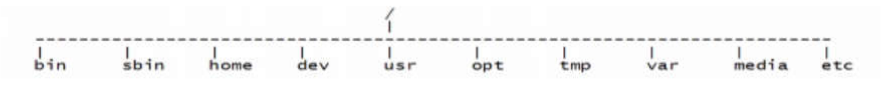


default shell **echo $SHELL**

bin/bash

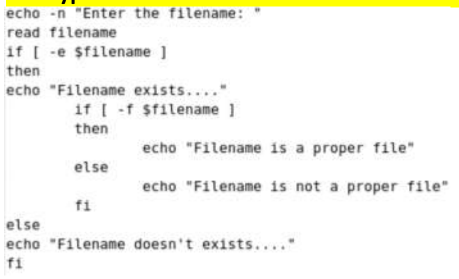
For IBM AIX: **bin/KSH**
For Solaris: **bin/SH**

File hierarchy



/ = root
/bin = user login, you will get $ symbol and contains all user level commands
/sbin = admin login, you will get # symbol and contains all admin level commands
/home = default home directory,if user admin create a new user it will create under home directory only
/dev = device files -- CSF (Char special file) -- BSF (Block special file)
/usr = perl and phyton packages and does not belong to OS.
/opt = optional directory using for to install any third party applications
/tmp = temporary files
/var = contains log files and email informations.... type **mail**
/etc = contains all system config files. **cat /etc/passwd =** contains user information
whereas **cat /etc/shadow** = contains password information

File types: To check whether a file is a proper file or directory



```
echo -n "Enter the filename: "
read filename
if [ -e $filename ]
then
echo "Filename exists...."
        if [ -f $filename ]
        then
                echo "Filename is a proper file"
        else
                echo "Filename is not a proper file"
        fi
else
echo "Filename doesn't exists...."
fi
```

Getting the filename.... N is to get the filename on the same line
Saves the input into a variable "filename"
-e checks whether the file exists
-f checks whether the file is a proper file

echo prints the message on the screen

3 types:

1. Regular files -- printable files -- txt files
2. Directory files -- Lists all files with i-node (identification) number in files and sub directories use ls -l to see
3. Device files -- Two types --
CSF (character special files) -- cat files -c
BSF (Block special file) -- -b -- data is stored in terms of box. such has 512 GB harddrive has
512 boxes. if a file is 1000 bytes then it covers in 2 boxes

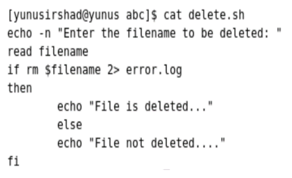
sample.sh: ` -- backtick to execute command
----
os_name=`uname`

elif [$os_name ==`Linux`] elif == else if in JAVA

cal	display calendar
cal 2015	display calendar of 2015
cal 01 01	cal "Jan" month year of 01
finger	this displays how much the user is idle, ip address, last login
finger yunus	LAST LOGIN ON SINCE: currently using
who | wc -l	displays how many user logged into the system
su username	switch user
echo $SHELL	display default shell in this OS
echo $0	display current shell
we can change by typing "ksh"	now current shell is "KSH"
Create a file	1. touch - to create empty files (multiple files) **f{1….10}** it will change the timestamp to current time of the file if youplan to create touch filename 2. cat - (concatenation) to create -- display -- append a file syntax: **cat mode filename** read mode: **<** write mode: **>** **after taking input click "CTRL+D"** append mode: **>>** 3. vi
grep -v "^$" filename	removes the blank records from the file…… do not redirect to same filename it is very danger… it deletes **grep -v "^$" new_filename**

Redirections

STDIN	Standard input: input redirection 0< 0 and < both are optional
STDOUT	Standard output: output redirection 1> 1 is optional but > is must
STDERR	error redirection 2> we can forward all error messages to filename
cat filename1 filename2 > output.log 2> error.log	Two files will come as output and error logs

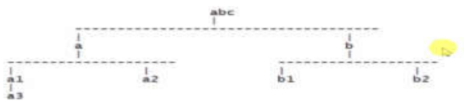


```
[yunusirshad@yunus abc]$ cat delete.sh
echo -n "Enter the filename to be deleted: "
read filename
if rm $filename 2> error.log
then
        echo "File is deleted..."
        else
        echo "File not deleted...."
fi
```

removes the file and redirects the error message to error.log

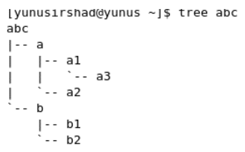
tr [a-z] [A-Z] < filename	STDIN is used here…….. It changes from lowercase to uppercase

mkdir dirname	creates a new directory



mkdir abc abc/a abc/b abc/a/a1 abc/a/a1/a3…	creation of total heirarchy in single step
mkdir -p abc/a/a1/a3 abc/a/a2 …..	"-p" is parent directory alternative to create a directories

tree abc



```
[yunusirshad@yunus ~]$ tree abc
abc
|-- a
|   |-- a1
|   |   `-- a3
|   `-- a2
`-- b
    |-- b1
    `-- b2
```

cd .	current directory
cd ..	parent directory
cd ../../../..	returns back to the parent directory
cd -	returns to the previous directory such as swap button in TV remote
cd or cd $HOME or cd ~	returns to the home directory
copy	
cp sample sample2	copying the file.. Default it will copy in same directory
cp dirname/* anotherdirname	copys all directories or files
cp -r dirname/* anotherdirname	copy of a directory with all sub directories and files
pwd	print working directory. Path of the directory
absolute path: cp sample home/yunus/abc/a	copy a file based on root directory
relative path: cp sample ../../../b/b1	copy a file based on directory. A sample file from a3 is moved to b1
cp ../../a/a1/a3/sample .	copy a sample file from a3 to b1. from staying in b1 directory…
cp -I sample sample2 Note: it is small "I"	it displays a message before overwriting **cp Overwrite "sample"? Y/N**
alias cp="cp -I"	cp works as cp -I
ls .ba* bash commands	
.bash_history --> history	history last executed commands
rmdir	removes the directory
rm filename	removes the file name
rm *	it deletes all files in directory
rm -I filename	it displays a message before deleting **rm remove regular file? Y/N**
rm -R directory	it will delete the directory even though it has files and sub directories
rm -rf f*	it will forcily delete all the files starts with "f*"

touch f{1-10}	creates the file using touch command

move

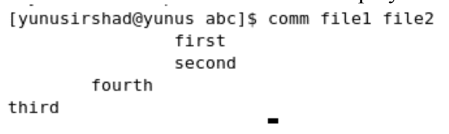
mv f* abc	moves all files which starts with "f" to abc directory
mv a a_1	it renames the file or directory
mv dirname/* anotherdirname	moves all the files and directories to another directory

VI editor

vi file …….		VI editor creates a file and inserts something into the file.
Enter..		
First…		
second to exit ESC :WQ		Saves the file
In Vi editor	**dd**	Delete a line
	x	delete a character
	shift+insert	paste the copied content
	:q!	doesn't save the file

comparing files

diff file1 file2	if there is a difference it will display otherwise it wont display
diff -arg folder1 folder2	summary of files which differ
diff -jars JAR1 JAR2	displays the difference between two jars
comm file1 file2	there are three fields or columns display



```
[yunusirshad@yunus abc]$ comm file1 file2
                first
                second
        fourth
third
```

comm -23 file1 file2	this will remove the fields and
cmp file1 file2	it compares, there is a difference then it will display which byte and line **file1 file2 differ: byte 14, line 3**

Listing

ls	list the files and directory
ls -a (ascending)	list all files including .files
ls -r (descending order)	

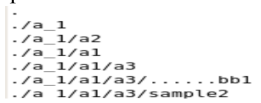
ls -t (Time order) **ls -rt (Reverse time order)**	list the files based on time which is created recently
ls -l or ls -lrt	long list with file permissions
ls -l | tail -3	lists last 3 files
ls list*	It matches wild card character suffix
ls [kfeg]* or ls [a-z]	ls k* f* e* …. Instead of writing this we can use
ls *list	
ls *.*	lists all extension files
ls ?list	matches just one character.... you can add multiple ? ???
ls l*d	lists all files starts with l and ends with d
ls -d */	lists alldirectories
emacs filename	creates a new file…. A new pop up window opens where you can write and save it
more filename	displays the contents of the file…. Click space to see some more content or q to quit.
chmod o+rx filename	Changing the directory permissions to read - write - execute drwx-wxr-x 2 username groupname others size_of_file time nameofthedir_file

chmod o-r filename (to remove permissions)	d = directory
	rwx = user permissions

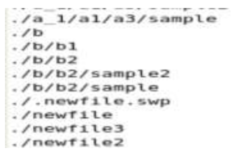
chmod [u/g/o] [+/-/=] [r/w/x] filename	-wx = group permissions r-x = other permissions
chmod g+wx, o+w filename	
chmod 777 filename	
gzip filename	zips the filename to filename.gz
gunzip filename	unzips the filename
zcat filename	reads the zip file
date	displays the current time with date
less filename	displays the less content of the file into the screen, click Space to get more content or q to quit
less filename **ENTER** **/<searchitem>**	simple way to search a item in the file
head filename	displays the content of the file into the screen but only 10 lines
head -15 filename	usually by default it displays 10 lines, if you want to display 15 lines use this command
tail filename	displays the content from the bottom of the file but only 10 lines it will display
tail -15 filename	use this to display 15 lines from bottom
wc filename	displays number of lines, words, characters in the file
wc "-w"	display number of words
wc "-l"	display number of lines
wc "-c"	display number of characters

display content of files

cat filename	displays the contents of the file
cat > filename **yunus** **irshad** **PRESS CTRL+D to come out and save**	inserting some data into the file
cat >> filename	appending some data into the file
jarhomepath tf jarfilename	viewing the contents of jar file
uname -a	displays the machine type with versions
uname -n	displays username
sort filename	sorts the data in the file
sort -u filename	delete all duplicate and display unique records with one instance
sort $fn | uniq -u >tmp	sorting and displaying unique values into tmp file
who	displays all the users who are logged into the system
whoami	displays current username
who am I	displays last person who logged early
man cat	displays the online manuals for cat command
whatis grep	displays the description of the grep command …what it does?
apropos grep	Displays all details which contains all grep commands…. Such as bzgrep, ungrep….all greps %grep%
grep <searchitem> filename	searches for the item in the file
grep -I <searchitem> filename	-I ignores the uppercase and lowercase
grep -v <searchitem> filename	displays the other contents than the searched item
grep -n <searchitem> filename	displays with line number
grep -c <searchitem> filename	prints the total number of the words of the searched item
grep -ivnc <searchitem> filename	you can use multiple commands at the same time
find . -print	prints the fields and subfolders of the directory



```
.
./a_1
./a_1/a2
./a_1/a1
./a_1/a1/a3
./a_1/a1/a3/......bb1
./a 1/a1/a3/sample2
```



```
./a_1/a1/a3/sample
./b
./b/b1
./b/b2
./b/b2/sample2
./b/b2/sample
./.newfile.swp
./newfile
./newfile3
./newfile2
```

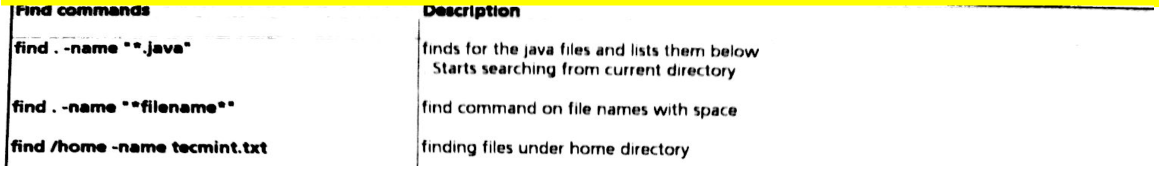
Processes and Jobs

ps	A process is an executing program identified by a unique PID (process identifier). View status of process A process may be in the foreground, in the background, or be suspended. In general the shell does not return the UNIX prompt until the current process has finished executing.
sleep 10	This will wait 10 seconds before returning the command prompt %. Until the command prompt is returned, you can do nothing except wait.
sleep 10 &	To run sleep in the background
CTRL + Z to stop the sleep	**[1] 2735** The user is be notified of a job number (numbered from 1) enclosed in square brackets, together with a PID and is notified when a background process is finished. Backgrounding is useful for jobs which will take a long time to complete.
sleep 10 **CTRL+C**	It is sometimes necessary to kill a process (for example, when an executing program is in an infinite loop)
kill jobnumber	To kill a suspended or background process, type
kill -9 jobnumber	Force kill
bg	After **sleep 10**, Backgrounding a current foreground process **Output: job is terminated,** if it completes 10 seconds sleep
jobs	Listing suspended and background processes
fg jobnumber	restart the job

Other commands

df .	reports on the space left on the file system. For example, to find out how much space is left on the fileserver
du -s *	The du command outputs the number of kilobyes used by each subdirectory.
file *	file classifies the named files according to the type of data they contain, for example ascii (text), pictures, compressed data, etc..

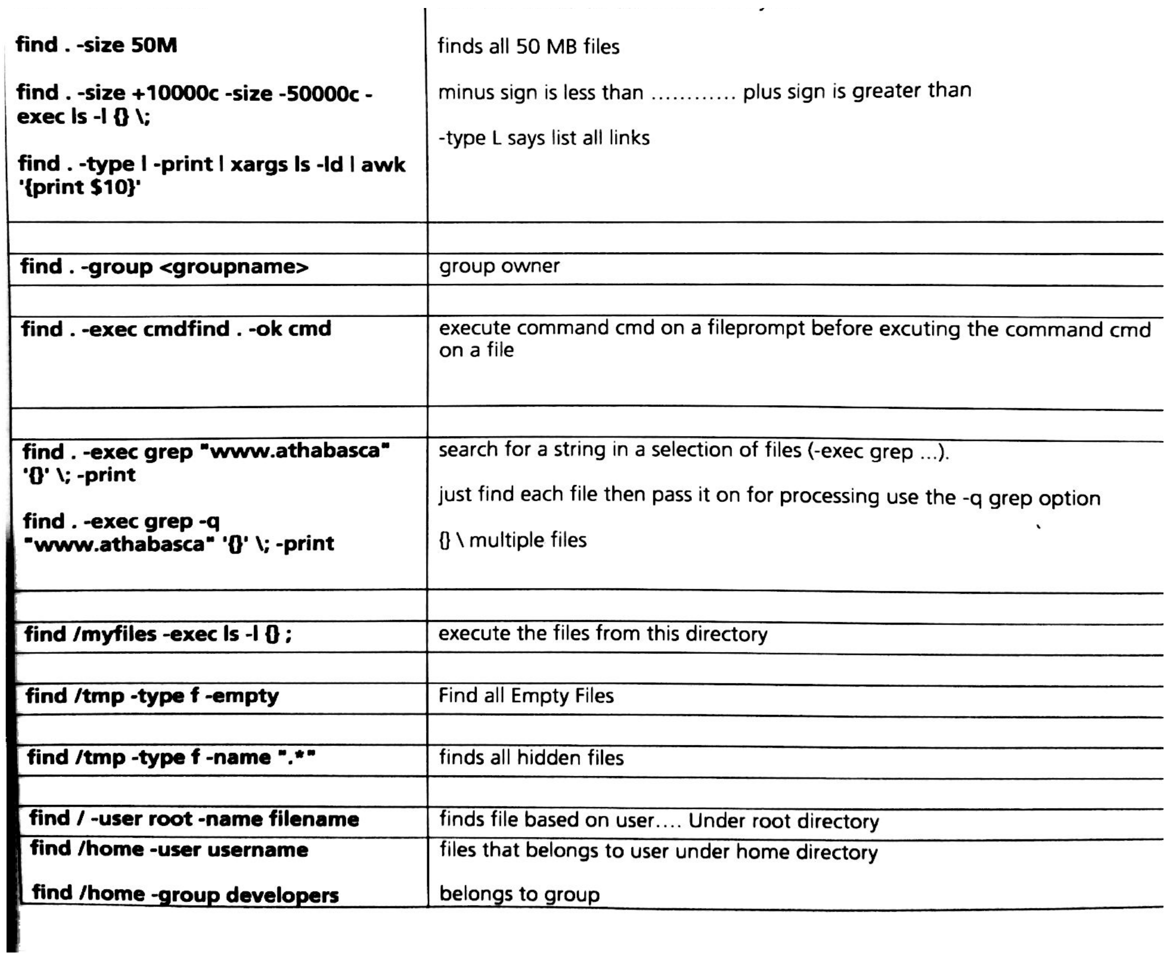
Find commands



Find commands	Description
find . -name "*.java"	finds for the java files and lists them below Starts searching from current directory
find . -name "*filename*"	find command on file names with space
find /home -name tecmint.txt	finding files under home directory

!find	display the last executed **find** command
find . -atime	file was **accessed** n days ago
find . -mtime 1	lists the file which got **modified** exact 1 day
find . -mtime -1	less than one day
find . -mtime +1	more than one day
-cmin -60	**changed** files in less than 60 minutes
find . -perm 644	finds files or directories based upon permissions
find . ! -perm 644	without permissions
	File permission in Numeric format: 0 – no permissions 1 – execute only 2 – write only 3 – write and execute 4 – read only 5 – read and execute 6 – read and write Default permission of file -- 666 (110 100 100) rw- r - r-- 7 – read, write and execute Default permission of directory -- 777
	Find Read Only Files
find / -perm /u=r	find all executable files
find . -iname "error" -print	case insensitive -i is for ignore find is extremely helpful while looking for errors and exceptions in log file. **-print0** = should be used to avoid any issue with white space in file name or path **-print** = display path name of matching files
find . -name "*.java" -print l xargs rm -f **use find . -delete "*.java"**	delete file using find command. Use of xargs along with find gives you immense power to do whatever you want with each search result.
find . -name "*.txt" -print l xargs grep "error"	this will search all java files starting from current directory for word "error"
find . -maxdepth L -type f -newer first_file	–type option can be used to specifiy search for only file, link or directory and maxdepth specifies how deep find has to search. **F for file and d for directory**
find . -type f -cmin 15 -prune	last modified 15 minutes ago, only look at the current directory. (No sub-directories)
find . -size +1000c	find files based on certain size in bytes
find . -size 50M	finds all 50 MB files
find . -size +10000c -size -50000c -exec ls -l {} \\;	minus sign is less than .... plus sign is greater than
find . -type l -print l xargs ls -ld l awk '{print $10}'	-type L says list all links
find . -group \<groupname\>	group owner

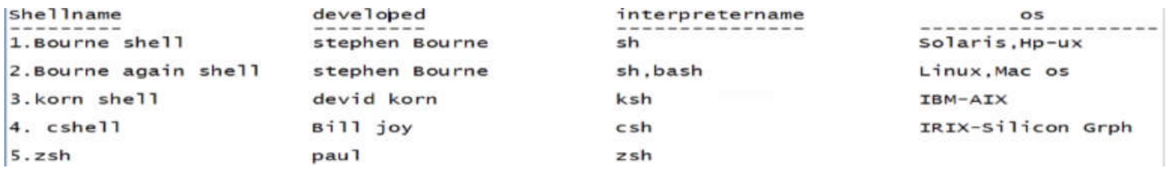
find . -name "*.java" -print l xargs rm -f **use find . -delete "*.java"**	delete file using find command. Use of xargs along with find gives you immense power to do whatever you want with each search result.
find . -name "*.txt" -print l xargs grep "error"	this will search all java files starting from current directory for word "error"
find . -maxdepth L -type f -newer first_file	–type option can be used to specifiy search for only file, link or directory and maxdepth specifies how deep find has to search. **F for file and d for directory**
find . -type f -cmin 15 -prune	last modified 15 minutes ago, only look at the current directory. (No sub-directories)
find . -size +1000c	find files based on certain size in bytes



find . -size 50M	finds all 50 MB files
find . -size +10000c -size -50000c -exec ls -l {} \;	minus sign is less than ………… plus sign is greater than
find . -type l -print | xargs ls -ld | awk '{print $10}'	-type L says list all links
find . -group <groupname>	group owner
find . -exec cmdfind . -ok cmd	execute command cmd on a fileprompt before excuting the command cmd on a file
find . -exec grep "www.athabasca" '{}' \; -print	search for a string in a selection of files (-exec grep ...).
find . -exec grep -q "www.athabasca" '{}' \; -print	just find each file then pass it on for processing use the -q grep option
	{} \ multiple files
find /myfiles -exec ls -l {} ;	execute the files from this directory
find /tmp -type f -empty	Find all Empty Files
find /tmp -type f -name ".*"	finds all hidden files
find / -user root -name filename	finds file based on user…. Under root directory
find /home -user username	files that belongs to user under home directory
find /home -group developers	belongs to group

Shell scripting

cat /etc/shells	display all list of shells
echo $SHELL	default shell with path
echo $0	current shell name
sh sample.sh	Execution: sample.sh must have a execute permission change it using chmod
echo "hello world"	
ksh sample.ksh	Execution of K-shell…..
print "hello world"	It will not execute using **./sample.ksh** then we should add the interpreter path **#! which ksh** (path of ksh) in VI



```
Shellname           developed          interpretername     os
----------          ----------         ---------------     -------------------
1.Bourne shell      stephen Bourne     sh                  Solaris,Hp-ux
2.Bourne again shell stephen Bourne    sh,bash             Linux,Mac os
3.korn shell        devid korn         ksh                 IBM-AIX
4. cshell           Bill joy           csh                 IRIX-Silicon Grph
5.zsh               paul               zsh
```

Restricted permission **vi /etc/passwd**

chmod +w /etc/passwd // this will show error "Operation not permitted

How to write Shell scripting?

Text editor : VI or nano or pico or ed

vi sample.sh //extension is must

#! /bin/bash // shebang line invokes the interpreter
# purpose of the script
# version // change the version same like SVN
# date // script start and end
# author name

echo "hello"

How to execute a script?

1. sh sample.sh
2. ./sample.sh --> permission denied because a script must have R+X permission use chmod
use shebang line

Comments in shell script

single line comment

File permissions

ls -lrt

- rwx r-- rw- 1 yunus yunus 02 jan 16 7:07 sample.sh

First one: directory (d) or file
rwx : belongs to owner or user (U)
r-- : belongs to group (G)
rw- : belongs to others (O)

default permission: **-rw- rw- rw- (numeric way it is 666) read = 4, write = 2, execute = 1 TOTAL=7 (rwx)**

Why read=4 write=2 execute=1 ? Octal numbers (base 8) --> r-- = 100 = 4

but based on umask (user creation mask) the permissions will change
umask --> display 0022
then **umask 0202** now permission of the file change. **umask - 666 = 464 (r-- rw- r--)**

0 --> sticky bit if it is 1, then owner can only delete the file
0 --> it deletes the permission from owner
2 --> it deletes permission from group
2 --> it deletes permission from other

Variables: no data types

int a=10;	java
a=10	shell scripting, by default variable type is string
echo "Variable is $a"	Variable is 10 // user defined variables should be in lowercase.
	1. local variables: by default
readonly a	2. constant or read only variables: we cannot change the value now
export a	3. global variables: converts from local to global. used from bash shell to KSH shell
set -o vi	Up arrows will work in ksh
echo $SHELL	System defined variables should be UPPERCASE
env or set	displays all system defined variables
echo "command: `pwd`"	command will work only when it is declared in between `....`
echo -e "newline entered"	new line gets inserted as \n
read a	taking input from the keyboard
read -p "Enter name: " name	prompting for taking input
echo $name	
read -s -p "Enter password:"	-s = security, it wont display the password .... hidden ...no stars
String	String limit is unlimited in unix.
Str = 'Hello world'	1. Single quoted string '......' = display as it is $a+$b
Str = 'Hello world don't' //error	2. Double quoted string "....." = variable execution takes place $a+$b = 10+20
Str = "Hello world don't"	
" \" "	= acceptable to display double quotes
	3. back quoted string `......` = used for OS commands
i=10	
j=20	
echo `expr $i+$j`	it converts string expression to integer expression
Operators	we do not have increment/ decrement operators
	1. Arithmetic operators:
	2. Relational operators:
	3. Logical operators:
	4. Assignment operators:
	5. short hand assignment operators:
	6. range operator
Arithmetic operators	"+ - * / %"

h=6 j=10	
echo `expr $h + $j`	converts string to integer variable, There must be a space between arithmetic operators
echo `expr $h \* $j`	multiplication operator use \ which eliminates wildcard character

bc	**bc = binary calculator,** it accepts all integer, float….
CTRL+D to terminate	It also check relational operators like 4>2 displays 1 which is true

h=2.5	Pipe: command one output to the command two input
j=3.5	
echo $h + $j | bc	6.0 is answer

#write a script accept 2 numbers and do arithmetic

```
echo -n "Enter a: "
read a
echo -n "Enter b: "
read b
echo "The sum is `expr $a + $b`"
```

Relational operators	**a) numeric comparison**
	-gt (greater than)
if [$a -gt $b]	-lt (less than)
	-ge (greater than or equal)
	-eq (equal)
	-le (less than or equal)
	-ne (not equal)
	b) String comparison
	==
	>
"abc" < "bcd" Ans: 1 First char b is greater	**<**
	>=
	<=
	!=

Logical operators	
-a	AND
-o	OR
-n or !	NOT

Assignment operator	assign a value to the variable
i=10	

short hand assignment operator	
+=	p +=2 // p = p+2
-=	
***=**	
/=	

range operators	low to high values we use in for loops
1..10	
a….z	

Conditional statements

simple if	if (condition)
if [$a -gt $b]	{…}
then	
…..	
fi	
if…else	
Else	
……	
fi	
nested if	
ladder if	
case statement	
id -u	id of username of Linux