

Commands Used In Linux:-

date (displays the current date, time and day).

echo (used as print statement in Linux).

ls (list of directories).

pwd (present working directory).

mkdir (create directory).

cd (to go into created directory).

cd\ (to get out of directory).

rmdir (to remove directory).

vi (to create file , and then click insert button and then esc and :wq to get out of the file(save and quit)).

cat 'name' (display written content)

touch (to create empty file).

cp (copy) [cp 'old file name' 'new file name'].

cat>'name' (ctrl+z).

mv (to move [mv 'old file name' 'new file name']).

clear (to clear screen).

.sh (for saving shell script extension).

who (to check the current host).

wc (to count number of characters).

cal 'year' (to display calendar).

ls :- Returns list of directories and files

mkdir :- Makes a new directory

rmdir :- Removes the specified directory

pwd :- Returns the present working directory.

wc :- Returns the word count of the file specified

touch :- Creates a new empty file

vi :- Create a new file that can be edited

cat :- Creates a new file that is edited on the same screen and we exit it by pressing ctrl+z

echo :- Print the statement

Shell Script:-

- 1) Write a shell script to display the number of users currently logged in the system, display the present working directory, and display the calendar of your own birth year and display the today's date and time.

Code:-

```
echo -e "No. of users in the system : `who` ";
```

```
echo -e "Present working directory : `pwd`";
```

```
echo -e "Calendar of birth month : `cal 06 2004` ";
```

```
echo -e "Current date and time : `date` ";
```

Write a shell script :

- 2) To create an empty file.

Copy the content of one file to another file.

Display the current terminal.

Count the number of word in file.

Code:-

```
echo -e "To create an empty file : `touch project`";
```

```
echo -e "To copy the file : `cp shradha sriya`";
```

```
echo -e "To display tty: `tty`";
```

```
echo -e "To count number of words : `wc shradha` ";
```

3] Write a shell script :

- ❖ To create a directory.
- ❖ Move the content of one file to another file
- ❖ Display the list of directories and files.
- ❖ To remove a directory

CODE:

```
echo -e "To create directory : `mkdir prac`";  
echo -e "To move content : `mv SAROJ das`";  
echo -e "To display list of directory : `ls`";  
echo -e "To remove directory : `rmdir prac`";
```

4] To accept value from users to solve the following expression:

$+, -, *, /, \%$.

CODE:

```
echo -e "Enter two numbers:";  
read a b;  
echo -e "a+b = `expr $a + $b`";  
echo -e "a-b = `expr $a - $b`";  
echo -e "a*b = `expr $a * $b`";  
echo -e "a/b = `expr $a / $b`";  
echo -e "a%b = `expr $a % $b`";
```

5] Write a shell script which read number and check whether negative or zero. Print appropriate message.

CODE:

```
echo -e "Enter a number";  
read a  
if [ $a -gt 0 ]
```

```
then echo -e "$a is positive";  
elif [ $a -lt 0 ]  
then echo -e "$a is negative";  
else  
echo -e "$a is equal to 0";  
fi
```

6] Write a shell script which read number and check whether even or odd.

CODE:

```
echo -e "Enter a number";  
read a  
if [ `expr $a % 2` == 0 ]  
then echo -e "$a is even";  
else  
echo -e "$a is odd";  
fi
```

7] Write a shell script to find the greater number between two.

CODE:

```
echo -e "Enter two numbers:";  
read a b  
if [ $a -gt $b ]  
then echo -e "$a is greater";  
else  
echo -e "$b is greater";  
fi
```

8] Write a shell script to check whether the entered number is divisible by 3,5 or 7.

CODE:

```
echo -e "Enter a number";
read a
if [ `expr $a % 3` == 0 ]
then echo -e "$a is divisible by 3";
elif [ `expr $a %5` == 0 ]
then echo -e "$a is divisible by 5";
elif [ `expr $a % 7` == 0 ]
then echo -e "$a is divisible by 7";
else
echo -e "Error";

fi
```

9] Write a shell script which reads 2 numbers a and b and check whether a is smaller than or greater than b or equal to b.

CODE:

```
echo -e "Enter teo numbers:";
read a b
if [ $a -gt $b ]
then echo -e "$a is greater";
elif [ $a -lt $b ]
then echo -e "$a is smaller";
elif [ $a == $b ]
then echo -e "$a is equal to $b";
```

```
else
```

```
echo -e "Invalid";
```

```
fi
```

10] Write a shell script to print week day name of the week corresponding to week day number. Give appropriate message if the number is not between 1 to 7.

CODE:

```
echo -e "Enter day number";
```

```
read a
```

```
case $a in
```

```
1) echo -e "Sunday";;
```

```
2) echo -e "Monday";;
```

```
3) echo -e "Tuesday";;
```

```
4) echo -e "Wednesday";;
```

```
5) echo -e "Thursday";;
```

```
6) echo -e "Friday";;
```

```
7) echo -e "Saturday";;
```

```
*) echo -e "Invalid output! Kindly enter number between 1 to 7";;
```

```
Esac
```

11] Using case, write a shell script to find out the entered character is vowel or not.

CODE:

```
echo -e "Enter character";
```

```
read a
```

```
case $a in
```

```
a) echo -e "VOWEL";;
```

```
e) echo -e "VOWEL";;  
i) echo -e "VOWEL";;  
o) echo -e "VOWEL";;  
u) echo -e "VOWEL";;  
A) echo -e "VOWEL";;  
E) echo -e "VOWEL";;  
I) echo -e "VOWEL";;  
O) echo -e "VOWEL";;  
U) echo -e "VOWEL";;  
*) echo -e "Invalid Input";;
```

Esac

12] Write a shell script to enter the number from user for choice 1 for +, 2 for -, 3 for *. Get two numbers again from the user and calculate the result.

CODE:

```
echo -e "Enter number between 1 and 3";  
read a  
echo -e "Enter two numbers";  
read b c  
case $a in  
1) echo -e "b+c = `expr $b + $c`";;  
2) echo -e "b-c = `expr $b - $c`";;  
3) echo -e "b*c = `expr $b \* $c`";;  
*) echo -e "Invalid Input";;  
Esac
```

13] Write a shell script to read a character n check whether it is upper case letter or lower case letter or digit or any other special character and give appropriate message.

CODE:

```
echo -e "Enter a character : ";
read n
case $n in
[A-Z]) echo -e "$n is an upper-case alphabet";;
[0-9]) echo -e "$n is a digit";;
[a-z]) echo -e "$n is a lower-case alphabet";;
*) echo -e "$n is a special character";;
esac
```

14] Write a shell script to print sum of n number using for and while loop.

CODE:

FOR:

```
echo -e "Enter size:";
read n
sum=0
echo -e "Enter numbers:";
for((i=1; i <= n; i++))
do
read num
sum=`expr $((sum+num))`
done
echo -e "Sum is $sum":
```


WHILE:

```
echo -e "Enter size:";
read n
sum=0
i=1
echo -e "Enter numbers:";
while [ $i -le $n ]
do
read num
sum=`expr $((sum+num))`
i=`expr $((i+1))`
done
echo -e "Sum is $sum":
```

C Programming in Linux:-

1. Write a C program to print sum of two numbers.

CODE:

```
#include<stdio.h>

int main()
{
int a,b ,c;
printf("Enter two numbers : ")
scanf("%d %d ",&a ,&b);
c=a+b;
printf("Sum is %d" , c);
return 0;
}
```

2] Write a shell script program to find even and odd number using C programming.

CODE:

```
#include<stdio.h>

int main()
{
    int a;
    printf("Enter a number : ");
    scanf("%d",&a);
    if(a%2==0){
        printf("Number is even: ", a);
    }
    else{
        printf("Number is odd :",a);
    }
    return 0;
}
```

3] Finding maximum between three number using C Programming.

CODE:

```
#include<stdio.h>

int main()
{
    int a,b,c;
    printf("Enter first number: \n ");
    scanf("%d",&a);
    printf("Enter second number:\n");
}
```

```

scanf("%d",&b);
printf("Enter third number:\n");
scanf("%d",&c);
if(a>b){
    if(a>c){
        printf("%d is max\n",a);
    }
    else{
        printf("%d is max\n", c);
    }
}
else{
    if(b>c){
        printf("%d is max\n",b);
    }
    else{
        Printf("%d is max\n",c);
    }
    return 0;
}

```

4] Find the factorial of number using C programming.

CODE:

```

#include <stdio.h>

int main()
{

```

```

int i;
int fact = 1;
int num;
printf("Enter a number:\n");
scanf("%d", &num);
for(i=1; i<=num; i++)
{
    fact=fact * i;
}
printf(" Factorial of %d is %d\n", num, fact);
return 0;
}

```

5. Write a C program in ubuntu to find the sum of two numbers.

CODE:

```

#include<stdio.h>
void main()
{
int a,b,c;
printf("Enter two numbers:");
scanf("%d %d", &a, &b);
c=a+b;
printf("Sum of %d and %d is %d\n", a, b, c);
}

```

Shell script program:-

1] Write a shell script to read marks in 5 subjects and find the percentage and

- a) If percentage is less than 35 then fail.
- b) If percentage is greater than 35 and less than 45 then grade C.
- c) If percentage is greater than 45 and less than 60 then grade B.
- d) If percentage is greater than 60 and less than 75 then grade A.
- e) If percentage is greater than 75 then distinction.

CODE:

```
echo "Enter five subject marks.";
read a b c d e;
total=`expr $a + $b + $c + $d + $e`
per=`expr $total / 5`
echo "Your percentage is $per%.";
if [ $per -lt 35 ]
then echo "You have failed.";
  elif [ $per -ge 35 -a $per -lt 45 ]
then echo "Grade C";
  elif [ $per -ge 45 -a $per -lt 60 ]
then echo "Grade B";
  elif [ $per -ge 60 -a $per -lt 75 ]
then echo "Grade A";
  else echo "Distinction";
fi
```

2. Write a program for a menu driven program with 5 choices.

A] To create a file with a name demo with some content.

B] To check the present working directory

C] To check then list of files including hidden files

D] To check content of demo

E] To exit

CODE:

```
echo -e "put your choice: 1 for creating file, 2 for recent working directory
, 3 for listing all files including hidden files, 4 to see the content of file, 5
to exit"
```

```
read c
```

```
case $c in
```

```
1) echo -e "`cat>demo` ";;
```

```
2) echo -e "`pwd` ";;
```

```
3) echo -e "`ls -a` ";;
```

```
4) echo -e "`cat demo` ";;
```

```
5) exit;;
```

```
*) echo -e "Invalid Option";;
```

```
Esac
```

1. Write a C program to create a child process and allow the parent to display “parent” and the child to display “child” on screen.

CODE:

```
#include <stdio.h>
```

```
#include <sys/wait.h> /* contains prototype for wait */
```

```
#include <stdlib.h>
```

```
#include <unistd.h>
```

```
int main(){
```

```
int pid;
```

```
int status;
```

```
printf("Hello World \n");
```

```
pid = fork();
```

```

if (pid == 1)
{

    perror("bad fork");
    exit(1);
}
if (pid == 0)
{
    printf("I am the child process \n");
}
else
{
    wait(&status);/*parent waits for child to finish */
    printf("I am the parent process");
}
}

```

3. Use different commands : cp, mv, chmod, rm, head, tail,tac, finger, date, calendar,more,string.

CODE:

Cat > "file name"

#content

Ctrl + Z

cp "old file name" "new file name"

cat > "new file name"

mv "old file name" "new file name"

rm "file name"

ls

head "file name"

tail "file name"

tac "file name"

finger

\$date

\$cal