

OS LAB

- ▼ List the names of the users logged in and their total count without displaying any further details

```
$ who -q
```

- ▼ Find out your terminal's device name.

```
$ tty
```

- ▼ Display current date in the form dd/mm/yyyy

```
$ date +%d/%m/%Y
```

- ▼ Find out your machine's name and the version of the operating system

```
$ uname -nr
```

- ▼ Clear the screen and place the cursor at row 12, column 25

```
$ clear
```

```
$ tput cup 12 25
```

- ▼ Find the decimal equivalent of 1101001

```
$ bc
```

```
ibase=2
```

- ▼ Find out the users who are idling

```
$ who -Hu
```

- ▼ Use man to get help

```
$ man tty
```

- ▼ Listing all files and directories and give it as input to more command

```
$ ls -l | more
```

- ▼ Use cat, grep, tee and wc command to read the particular entry from user and store in a file and print line count

```
$ ls -l | wc -l
```

```
$ cat sample2.txt | head -7 | tail -5
```

- ▼ Ensure that bc displays the results of all divisions using three decimal places

```
$ bc
```

```
Scale=3
```

- ▼ To view a single file

```
$cat filename
```

- ▼ To view multiple files

```
$cat file1 file2
```

- ▼ To view contents of a file preceding with line numbers

```
$cat -n filename
```

- ▼ Create a file

```
$ cat > newfile
```

- ▼ List all filenames starting with „a“ or „b“ or „m“

```
$ ls [abm] *
```

- ▼ List all filenames that end with a digit.

```
$ ls * [0-9]
```

- ▼ List all files in the current directory whose second character is a digit

```
$ ls ? [0-9]*
```

- ▼ Use command(s) to create a directory in your home directory called KeepOut whose contents can be read only by you

```
$ mkdir keepout ; chmod 400 keepout
```

- ▼ List all files beginning with character 'a' on the screen and also store them in a file called file1.

```
$ ls [a] * | tee file1
```

- ▼ Sort the output of who and display on screen along with total number of users. The same output except the number of users should be stored in a file file1.

```
$ who -q | sort ; who -Hu | cat >> file1
```

- ▼ Double space a file

```
$ pr -d file1
```

- ▼ Select lines 5 to 10 of a file

```
$ head -10 file1 | tail -n -5
```

- ▼ Find the user name and group id from the file /etc/passwd using the cut command.

```
$ cut -d ":" -f 1,4 /etc/passwd
```

- ▼ Extract the names of the users from /etc/passwd after ignoring the first 10 entries.

```
$ cut -d ":" -f 1 /etc/passwd | tail -n +11
```

- ▼ Sort the file /etc/passwd on GUID (primary) and UID (secondary) so that the users with the same GUID are placed together. User with a lower UID should be placed higher in the list

```
$ cut -d ":" -f 3,4 /etc/passwd | sort -n
```

- ▼ List from /etc/passwd the UID and the user having the highest UID.

```
$ sort -t ":" -r -n -k 3 /etc/passwd | cut -d ":" -f 1,3 | head -1
```

- ▼ Device a sequence which lists the five largest files in the current directory

```
$ ls -lS | head -6
```

- ▼ Remove duplicate lines from a file.

```
$ uniq file1
```

- ▼ Count the frequency of occurrences of words in a file.

```
$ sort file1 | uniq -c
```

- ▼ Find "long" listing of the smallest 5 files in the /etc directory whose name contains the string ".conf", sorted by increasing file size.

```
$ ls -lSr /etc/*.conf | head -5$ who | uniq |sort
```

- ▼ What would you type at the command line to get a sorted list, with no duplicates, of all the users logged into the local network?

```
$ who | uniq |sort
```

- ▼ What would you type at the command line to find all files in your home directory that are more than a week old and end with .bak?

```
$ find -mtime +7 -name "*.bak" -ls
```

- ▼ What would you type at the command line to find out how many total lines are contained in all the files ending in .c in the current directory, printing only the total number of lines?

```
$ wc -l *.c
```

- ▼ \$ cat state.txt

Andhra Pradesh

Arunachal Pradesh

Assam Bihar

Chhattisgarh

- ▼ \$ cut -c 2,5,7 state.txt

nr rah sm ir hti

- ▼ \$ cut -c 1- state.txt

Andhra Pradesh

Arunachal Pradesh

Assam Bihar

Chhattisgarh

- ▼ \$ cut -c -5 state.txt

Andhr

Aruna

Assam

Bihar

Chhat

- ▼ Find the user name and group id from the file /etc/passwd using the cut command

```
$ cut -d ":" -f 1,4 /etc/passwd
```

- ▼ Extract the names of the users from /etc/passwd after ignoring the first 10 entries.

```
$ cut -d ":" -f 1 /etc/passwd | tail -n +11
```

- ▼ Sort the file /etc/passwd on GUID (primary) and UID (secondary) so that the users with the same GUID are placed together. User with a lower UID should be placed higher in the list.

```
$ cut -d ":" -f 3,4 /etc/passwd | sort -n
```

- ▼ List from /etc/passwd the UID and the user having the highest UID.

```
$ sort -t ":" -r -n -k 3 /etc/passwd | cut -d ":" -f 1,3 | head -1
```

- ▼ Device a sequence which lists the five largest files in the current directory.

```
$ ls -lS | head -6
```

- ▼ Remove duplicate lines from a file.

```
$ uniq file1
```

- ▼ Count the frequency of occurrences of words in a file.

```
$ sort file1 | uniq -c
```

- ▼ Find "long" listing of the smallest 5 files in the /etc directory whose name contains the string ".conf", sorted by increasing file size.

```
$ ls -lSr /etc/*.conf | head -5
```

- ▼ What would you type at the command line to get a sorted list, with no duplicates, of all the users logged into the local network?

```
$ who | uniq | sort
```

- ▼ What would you type at the command line to find all files in your home directory that are more than a week old and end with .bak?

```
$ find -mtime +7 -name "*.bak" -ls
```

- ▼ What would you type at the command line to find out how many total lines are contained in all the files ending in .c in the current directory, printing only the total number of lines?

```
$ wc -l *.c
```

- ▼ Find out the PID of your login shell.

```
$ ps
```

- ▼ Remove the header line from the ps output.

```
$ps --no-headers
```

- ▼ List all processes that you are currently running on your machine, sorted by the command name in alphabetical order. The output should consist only of the processes you are running and nothing else (i.e. if you are running 6 processes, the output should only have 6 lines).

```
ps -e --sort=args
```

- ▼ Display the files in the current directory that contain the string MCA HITK in either upper- or lowercase

```
$ grep -il 'MCAHITK' *
```

- ▼ Store in a variable the number of lines containing the word MCA in the files mca1, mca2 and mca3

```
$ var=`grep MCA mca[1-3] | wc -l`
```

```
$ echo $var
```

- ▼ If you did not have the wc command, how would you use grep to count the number of users currently using the system?

```
who | grep -c ".*"
```

- ▼ Remove blank lines from a file using grep.

```
$ grep -v "^$" aa1
```

- ▼ List the ordinary files in your current directory that are not writable by the owner.

```
$ ls -l | grep -v '^..w'
```

- ▼ Locate lines ending and beginning with a dot and containing anything between them.

```
$ grep '\..*\.$' mca4
```

- ▼ Locate lines that are less than 100 characters in length

```
$ grep '\.{0,99}$' * file1
```

- ▼ Match all lines that start with 'hello'. E.g: "hello there"

```
$ grep "^hello" file1
```

- ▼ Match all lines that end with 'done'. E.g: "well done"

```
$ grep "done$" file1
```

- ▼ Match all lines that contain any of the letters 'a', 'b', 'c', 'd' or 'e'.

```
$ grep "[a-e]"
```

- ▼ Match all lines that do not contain a vowel

```
$ grep "[^aeiou]" file1
```

- ▼ Match all lines that start with a digit following zero or more spaces. E.g: " 1." or "2."

```
$ grep " *[0-9]" file1
```

- ▼ Match all lines that contain the word hello in upper-case or lower-case

```
$ grep -i "hello"
```

Shell Script

- ▼ Write a shell script to calculate addition of two numbers

```
clear
echo -n "Enter 1st number: "
read first_number
echo -n "Enter 2nd number: "
read second_number

sum=$((first_number + second_number))

echo "Sum of $first_number and $second_number: "$sum
```

- ▼ Write a shell script to show the maximum of three numbers.

```
#shell script to find the greatest of three numbers

echo "Enter Num1"
read num1
echo "Enter Num2"
read num2
echo "Enter Num3"
read num3

if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]
thenecho $num1
elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]
thenecho $num2
elsecho $num3
fi
```

- ▼ Write a shell script which displays the result of division of one integer by another integer and informs if the user tries to divide an integer by 0.

```
#!/bin/sh
## Shell script of dividing two number
while [ 1 ]
do
clear

echo -n "Enter the Numerator : "
read num
echo -n "Enter the Denominator : "
read den

if [ $den == 0 ]; then
echo -e "\nDenominator can not be zero."
sleep 2

else
res=`echo "scale=10; $num / $den" | bc`
echo "The answer is : $res"
exit
fi
```

- ▼ Rajesh's basic salary (BASIC) is input through the keyboard. His dearness allowance (DA) is 52% of BASIC. House rent allowance (HRA) is 15% of BASIC. Contributory provident fund is 12% of (BASIC + DA). Write a shell script to calculate his gross salary and take home salary using the following formula:

Gross salary = BASIC + DA + HRA

Take home salary = Gross salary - (BASIC + DA) * 0.12

```
echo "Please enter Basic Salary"
read basic
DA = $basic\*0.52
HR = $basic\*0.15
providentFund = $basic + $DA
echo "Provident Fund is: $providentFund"
grossSalary = $basic + $DA + $HRA
echo "Gross Salary: $grossSalary"
THS = $grossSalary-($basic+$DA)\*0.2
echo "Take Home Salary: $THS"
```

▼ Write a shell script that produces a shell calculator to perform the following operations:

□ Addition □ Subtraction □ Multiplication □ Division

```
# !/bin/bash
# Take user Input
echo "Enter Two numbers : "
read a
read b
echo "Enter Choice : "
echo "1. Addition"
echo "2. Subtraction"
echo "3. Multiplication"
echo "4. Division"
read ch

case $ch in
1)res=`echo $a + $b | bc`;
2)res=`echo $a - $b | bc`;
3)res=`echo $a \* $b | bc`;
4)res=`echo "scale=2; $a / $b" | bc`;
esac
echo "Result : $res"
```

▼ Write a shell script that displays a list of all files in the current directory to which you have read write and execute permissions.

```
#!/bin/sh
#Purpose: To display list of files in current directory to which owner have read, write and execute permission.
# USAGE: sh rwxperm.sh (no arguments required)
for var in *
do
if test -r $var -a -w $var -a -x $var -a ! -d $var
then
ls $var
fi done
#end of script
```

▼ Write a shell script that lists files by modification time when called with `lm` and by access time when called with `la`. By default, the script should show the listing of all files in the current directory

```
#!/bin/sh
```

```
# Purpose: To list the files according to modification or access time depending on the arguments in
```

command line.

```
# USAGE: sh filelist .sh lm(for modification time) or la (last access time)
case $1 in
lm) ls -lt ;;
la) ls -lut;;
*) ls -l;;
esac
#end of script
```

- ▼ Write a shell script to display the files created or updated within fourteen days from the current date

```
#!/bin/sh
# Purpose: To list the files created or updated within fourteen days from current date.
# USAGE: sh filecreat14.sh (no arguments required)
find -atime -14 -mtime -14 | sort -u
#end of script
```

- ▼ Develop a shell script that displays all files with all attributes those have been created or modified in the month of November

```
#!/bin/sh
#Purpose: To list those files created or updated in month of November.
# USAGE: sh createNOV.sh (no arguments required)
for var in *
do
set -- `ls -l $var` if test "$6" = "Nov"
then
ls -l $var
fi
done
#end of script
```

- ▼ Write a shell script to check if a given file (filename supplied as command line argument) is a regular file or not and find the total number of words, characters and lines in it.

```
file="$1"
w=`cat $file | wc -w`
c=`cat $file | wc -c`
l=`grep -c "." $file`
echo "Number of characters in $file is $c"
echo "Number of words in $file is $w"
echo "Number of lines in $file is $l"
```

- ▼ Write a shell script to check whether the given file is a blank file or not. If not found blank then display the contents of the file

```
if [ -s /tmp/myfile.txt ]
then
echo "File not empty"
else
echo "File empty"
```



```
cat /tmp/myfile.txt
fi
```

▼ Write a shell script to concatenate two files and count the number of characters, number of words and number of lines in the resultant file.

```
#!/bin/bash
echo Enter the filenames
read file1 file2
$ cat file1 file2 > file.txt
c=`cat $file | wc -c`
w=`cat $file | wc -w`
l=`grep -c "." $file`
echo Number of characters in $file is $c
echo Number of words in $file is $w
echo Number of lines in $file is $l
```

▼ Write a shell script, which gets executed the moment a user logs in. It should display the message “GOOD MORNING” or “GOOD AFTERNOON” or “GOOD EVENING” depending upon the time at which the user logs in.

```
hour=$(date +%H)
if [ $hour -ge 0 -a $hour -lt 12 ]
then
    greet="Good Morning, $USER"
elif [ $hour -ge 12 -a $hour -lt 18 ]
then
    greet="Good Afternoon, $USER"
else
    greet="Good evening, $USER"
fi
echo $greet
```

▼ Write a shell script, which reports names and sizes of all files in a directory (directory should be supplied as an argument to the shell script) whose size exceeds 100 bytes. The filenames should be printed in decreasing order of their sizes. The total number of such files should also be reported.

```
#!/bin/sh
#Purpose: To print the name of all files in a directory whose size exceeds 100 bytes alongwith total
number of such files.
# USAGE: sh file100byte.sh directoryname
if test $# -ne 1
then
    echo "Please give a directory name and try again"
    exit
fi
cd $1 find -size +100b | sort -nr
echo "Total number of such files:"
find -size +100b | grep -c ".*"
#end of script
```

- ▼ Write a shell script to list the name of files under the current directory that starts with a vowel

```
#!/bin/sh
#Purpose: To list the files starting with a vowel.
# USAGE : sh filevowel.sh (no arguments required)
echo "Required files are:"
ls | grep "[aeiou]"
#end of script
```

- ▼ Write a shell script which receives two filenames as arguments and checks whether the two file's contents are same or not. If they are same then the second file should be deleted

```
#!/bin/sh
#Purpose: To compare two files and delete the second file if they are equal.
# USAGE: sh filecomparedel.sh file1 file2
if test $# -ne 2
then
echo "Please give two filenames."
exit
fi
cmp -s $1 $2
if test $? -eq 0
then echo "$1 and $2 are same"
rm $2
else
echo "$1 and $2 are not same"
fi
#end of script
```

- ▼ A file called list consists of several words. Write a shell script which will receive a list of filenames, the first of which would be list. The shell script should report all occurrences of each word in list in the rest of the files supplied as arguments

```
if [ $# -eq 0 ]
then
echo "no arguments"
else
tr " " " "
" < $1 > temp
shift
for i in $*
do
tr " " " "
" < $i > temp1
y=`wc -l < temp`
j=1
while [ $j -le $y ]
do
x=`head -n $j temp | tail -1`
c=`grep -c "$x" temp1`
echo $x $c
j=`expr $j + 1`
done
```

```
done
fi
```

- ▼ Write a shell script which deletes all lines containing the word UNIX in the files supplied as arguments to this shell script.

```
#!/bin/sh
#Purpose: To delete the lines containing the word UNIX in the file supplied.
#USAGE: sh unixdelete.sh filename(s)
If test $# -eq 0
then
    echo "Please give filenames"
    exit
fi
for var is $*
do
    grep -v "UNIX" $var >ff
    cp ff $var
done
#end of script
```

- ▼ Write a shell script to display the list of users as well as the users connected to the system

```
#!/bin/bash

# Taking input from user
echo "Enter LOGNAME OR UID"
read input

# checking if input is a UID or LOGNAME
if [[ $input ]] && [ $input -eq $input 2>/dev/null ]

# If input is UID
then
    echo "Number of terminals are "
    cat /etc/passwd | grep $input -c

# If input is LOGNAME
else
    cat /etc/passwd>userlist
    echo "Number of terminals are "
    grep -c $input userlist
fi
```

- ▼ Write a shell script which counts the number of consonants and vowels in a given sentence.

```
#!/bin/sh

echo -n "Enter a line of text: "
read string

vowCount=$(echo $string | grep -o -i "[aeiou]" | wc --lines)
consCount=$(echo $string | grep -o -i "[bcdfghjklmnpqrstvwxyz]" | wc --lines)

echo "The given string has $vowCount vowels, $consCount consonants in it."
```

- ▼ Write a shell script to test whether a given string is palindrome or not.

```
echo "Input the string without space"
read str

for i in $(seq 0 ${#str}) ; do
    revstr=${str:$i:1}$revstr
done

echo "The given string is " $str
echo "Its reverse is " $revstr

if [ "$str" = "$revstr" ]; then
    echo "It is a palindrome."
else
    echo "It is not a palindrome."
fi
```

- ▼ Write a shell script to print the following pattern for any number of lines

```
p=5;

for((m=1; m<=p; m++))
do
    for((a=m; a<=p; a++))
    do
        echo -ne " ";
    done
    for((n=1; n<=m; n++))
    do
        echo -ne "#";
    done

    for((i=1; i<m; i++))
    do
        echo -ne "#";
    done

    echo;
done
```

- ▼ Write a shell script to find whether a number is divisible by 11

```
#!/bin/bash
echo "Enter any Number"
read n
r=`expr $y % 11`
if [ $r -eq 0 ]
then
echo $r " is divisible by 11"
else
echo $r " is not divisible by 11"
fi
```

- ▼ Write a shell script to check whether a given number is prime or not

```

echo "enter number"
read num
function prime
{
for((i=2; i<=num/2; i++))
do
    if [ $(num%i) -eq 0 ]
    then
        echo "$num is not a prime number."
        exit
    fi
done
echo "$num is a prime number."
}
r=`prime $number`
echo "$r"

```

- ▼ Write a shell program that takes a number from user and prints the reverse of the number.

```

echo enter n
read n
num=0
while [ $n -gt 0 ]
do
    num=$((expr $num \* 10))
    k=$((expr $n % 10))
    num=$((expr $num + $k))
    n=$((expr $n / 10))
done
echo number is $num

```

- ▼ Write a shell script to find the factorial value of any integer entered through the keyboard.

```

echo "Enter a number"
read num

fact=1

while [ $num -gt 1 ]
do
    fact=$((fact * num)) #fact = fact * num
    num=$((num - 1))     #num = num - 1
done
echo $fact

```

- ▼ Write a shell script to generate all combinations of 1, 2 and 3

```

clear
for i in 1 2 3
do
    for j in 1 2 3
    do
        for k in 1 2 3
        do
            echo $i $j $k
        done
    done
done

```

- ▼ Write a shell script to print all prime numbers in a given range.

```
echo enter m and n
read m n
for a in $(seq $m $n)
do
    k=0
    for i in $(seq 2 $(expr $a - 1))
    do
        if [ $(expr $a % $i) -eq 0 ]
        then
            k=1
            break
        fi
    done
    if [ $k -eq 0 ]
    then
        echo $a
    fi
done
```

- ▼ Write a shell script to find out whether an integer input through the keyboard is an odd number or an even number

```
clear
echo "---- EVEN OR ODD IN SHELL SCRIPT ----"
echo -n "Enter a number:"
read n
echo -n "RESULT: "
if [ `expr $n % 2` == 0 ]
then
    echo "$n is even"
else
    echo "$n is Odd"
fi
```

- ▼ Write a shell script to find out whether any year input through the keyboard is a leap year or not. If no argument is supplied the current year should be assumed.

```
leap=$(date +%Y)
echo taking year as $leap
if [ `expr $leap % 400` -eq 0 ]
then
    echo leap year
elif [ `expr $leap % 100` -eq 0 ]
then
    echo not a leap year
elif [ `expr $leap % 4` -eq 0 ]
then
    echo leap year
else
    echo not a leap year
fi
```