

# Unix Shell Programming

## Lab Record

Ajith Kumar G  
1BM19CS009



## B M S. COLLEGE OF ENGINEERING

(Autonomous Institution)

# RECORD OF PRACTICAL WORK

NAME : Ajith Kumar.G.....  
SUBJECT : USP.....  
SEMESTER : V BRANCH : CSE.....  
ROLL NO : ..... USN : 1BM19C5009

## -:- Program -1:

' Shell script to find if the given year is leap or not'

```

#!/bin/sh
echo "Enter the year below:"
read year
if [ $((year%100)) -eq 0 ]
then
    if [ $((year%400)) -eq 0 ]
    then
        echo "The year $year is leap year"
    else
        echo "The year $year is not leap year"
    fi
elif [ $((year%4)) -eq 0 ]
then
    echo "The year $year is leap year"
else
    echo "The year $year is not leap year"
fi

```

✓  
25/10/21

- Output:

- \$ sh program1.sh
- ↳ Entered the year below  
400  
The year 400 is leap year
- ↳ Entered the year below  
100  
The year 100 is not leap year
- ↳ Entered the year below  
16  
The year 16 is leap year
- ↳ Entered the year below  
17  
The year 17 is not leap year

### -:- Program - 2 :

' Shell script to find the area of a circle'

```
#!/bin/sh
radius = $1
pi = 3.14
ans = `echo $pi * $radius * $radius | bc`
echo "The area of circle is $ans"
```

### • terminal:

providing radius as command line argument.

N  
25/10/2021

- Output:

\$ sh program2.sh 40.4

The area of the circle is 5124.74

## :-) Program - 3:

' Shell script to check whether the number  
is zero / positive / negative '

```
#!/bin/sh
echo "Enter a Number:"
read n
if [ $n -gt 0 ]
then
    echo "The no $n is positive"
elif [ $n -lt 0 ]
then
    echo "The no $n is negative"
else
    echo "The no is zero"
```

✓  
M/10/21  
25/10/21

Output:

- \$ sh program3.sh
- ↳ Entered the Number:  
3  
The no 3 is positive
- ↳ Entered the Number:  
-4  
The no -4 is negative
- ↳ Entered the Number:  
0  
The no is zero



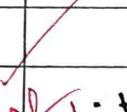
 Program - 4 :

'Shell script to find the biggest of three numbers'

```

#!/bin/sh
A = $1
B = $2
C = $3
max = $A
if [ $B -ge $max ]
then
    max = $B
fi
if [ $C -ge $max ]
then
    max = $C
fi
echo "The Max is $max"

```

 terminal :

25/10/21 Providing 3 nos as input through CLA.

• Output:

\$sh program4.sh 3 4 -1

The Max is 4

\$sh program4.sh 4 -1 3

The Max is 4

\$sh program4.sh -1 3 4

The Max is 4



## Program - 5:

' Shell script to find the factorial of a number'

```
V1: echo "Enter a positive Number"
```

```
fact=1
```

```
read n
```

```
for (( i=1 ; i<=$n ; i++ ))
```

```
do
```

```
fact=$((fact*i))
```

```
done
```

```
echo "The Factorial of $n is $fact"
```

```
V2: echo "Enter a number:"
```

```
fact=1
```

```
read n
```

```
if test $n -lt 0
```

```
then
```

echo "\$n is not greater than or equal to 0"; exit 1

fi

8/11/2021

```
for (( i=1 ; i<=$n ; i++ ))
```

```
do
```

```
fact=$((fact*i))
```

```
done
```

```
echo "The Factorial of $n is $fact"
```

- Output:

```
$ bash programs.sh  
Enter a positive number  
5  
The factorial of 5 is 120
```

Enter a number

-4

-4 is not greater than or equal to 0

### Program - 6 :

' Shell script to find the Gross Salary'

```
echo "Enter the basic salary"
read bs
DA='expr $bs * 10/100'
HRA='expr $bs * 20/100'
gross-salary='expr $bs + $DA + $HRA'
echo "The Gross salary is $gross-salary"
echo "Others: DA = $DA and HRA = $HRA"
```

✓  
8/11/2021

Teacher's Signature : \_\_\_\_\_

• Output:

\$ bash program6.sh

Enter the basic salary

2000

The Gross salary is 1800

Others: DA=100 and HRA=200

Program - 7:

' Shell script to convert from fahrenheit to celsius'

echo "Enter the temperature in fahrenheit"

read f

c='echo "scale=3 ; \$((\$f - 32) \* (5/9))" | bc'

echo "The \$f F in celsius is \$c"

✓  
8/11/2021

Teacher's Signature : \_\_\_\_\_

- Output:

```
$ bash program7.sh
```

```
Enter the temperature in fahrenheit
```

```
98
```

```
The 98F in celsius is 36.630C
```

-:- Program - 8 :

\* Shell script to perform arithmetic operations on two nos \*

```

echo "Enter the two numbers"
read x y
echo " ADD = $x + $y =" ; echo $x + $y | bc
echo " SUB = $x - $y =" ; echo $x - $y | bc
echo " MUL = $x * $y =" ; echo $x * $y | bc
echo " DIV = $x / $y =" ; echo "scale=3; $x / $y" | bc;
echo " MOD = " $x % $y ="; echo $x % $y | bc

```

✓  
8/11/2021

- Output:

\$ bash program8.sh

Enter the two numbers

8 7

ADD =  $8 + 7 =$

15

SUB =  $8 - 7 =$

1

MUL =  $8 \times 7 =$

56

DIV =  $8 / 7 =$

1.142

MOD =  $8 \% 7 =$

1



-:- Program - 9:

' Shell script to find sum of even numbers upto n'

```
echo "Enter the range number"
read n
sum = 0
for (( i=0 ; i<=n ; i=i+2))
do
    sum = $(($sum+i))
done
echo "The sum of even nos upto $n is $sum"
```

19/11/2021

- Output:

```
$ bash program9.sh  
Enter the range number  
10  
The sum of even nos upto 10 is 30
```

## Program - 10:

\* Shell script to print the combinations of  
the numbers 123\*

```
echo "Given Nos: 123"
list = (1 2 3)
for i in ${list[*]}
do
    for j in ${list[*]}
    do
        for k in ${list[*]}
        do
            echo $i $j $k
        done
    done
done
```

15/11/2021 done

- Output:

```
$ bash program10.sh
```

Given Nos: 123

1 1 1	→	2 1 1	→	3 1 1
1 1 2		2 1 2		3 1 2
1 1 3		2 1 3		3 1 3
1 2 1		2 2 1		3 2 1
1 2 2		2 2 2		3 2 2
1 2 3		2 2 3		3 2 3
1 3 1		2 3 1		3 3 1
1 3 2		2 3 2		3 3 2
1 3 3		2 3 3		3 3 3

→

→



## Program - 11:

\* Shell script to find the power of a number

```
V1: echo "Enter Number and its power"
read n p
ans=1
for (( i=p; i>0; i--))
do
    ans=$(echo "scale=3; $ans * $n" | bc)
done
echo "$n to power $p' is $ans"
```

```
V2: echo "Enter Number and its power"
read n p
result=1
if test $p -lt 0; then
    n=$(echo "scale=3; 1/$n" | bc)
    p=$(( -p ))
fi
for (( i=1; i<=$p; i++ ))
do
    result=$(echo "scale=3; $result * $n" | bc)
done
echo "Result = $result"
```

- Output:

```
$ bash program11.sh  
Enter Number and it's power  
2 6  
2 to power 6 is 64
```

Enter Number and it's power

2 -2

Result = 0.250

-2 -2

Result = 0.250

0 2

Result = 0

8 1

Result = 1

Program - 12 :

\* Shell script to find the sum of n natural numbers \*

```
echo "Enter Range no"
```

```
read n
```

```
sum = 0
```

```
for ((i=1; i<=n; i++))
```

```
do
```

```
    sum=$((sum+i))
```

```
done
```

```
echo "The sum of natural nos upto $n is $sum"
```

N  
15/11/21

Teacher's Signature : \_\_\_\_\_

• Output:

\$ bash program12.sh

Enter Range no

30

The sum of natural nos upto 30 is 465



## Program - 13

\* Shell script to find the pass class of a student \*

```

passed=0; failed=0
for ((i=1; i<=6; i++))
do
    echo -e "Enter the CIE (out of 50) and SEE (out of 100)
marks for sub-$i:"
read cie see
total=`echo "$cie + $see / 2" | bc`
echo "Total = $total"
case $total in
    100) echo "S Grade"; passed=$((passed+1));;
    9[0-9]) echo "S Grade"; passed=$((passed+1));;
    8[0-9]) echo "A Grade"; passed=$((passed+1));;
    7[0-9]) echo "B Grade"; passed=$((passed+1));;
    6[0-9]) echo "C Grade"; passed=$((passed+1));;
    5[0-9]) echo "D Grade"; passed=$((passed+1));;
    4[0-9]) echo "E Grade"; passed=$((passed+1));;
    [0-3][0-9]) echo "F Grade"; failed=$((failed+1));;
    *) echo "Error input, check it"; i=$((i-1));;
esac
done
echo "Total Passed = $passed and Total Failed = $failed"

```

Teacher's Signature : \_\_\_\_\_

- Output:

\$ bash program13.sh

Enter the CIE(out of 50) and SEE(out of 100) marks for sub-1:

46 88

Total = 90

S grade

Enter the CIE(out of 50) and SEE(out of 100) marks for sub-2:

48 75

Total = 85

A grade

Enter the CIE(out of 50) and SEE(out of 100) marks for sub-3:

49 89

Total = 93

S grade

Enter the CIE(out of 50) and SEE(out of 100) marks for sub-4:

47 80

Total = 87

A grade

Enter the CIE(out of 50) and SEE(out of 100) marks for sub-5:

48 85

Total = 90

S grade

Enter the CIE(out of 50) and SEE(out of 100) marks for sub-6:

49 93

Total = 95

S grade

Total passed = 6 and Total failed = 0

### Program - 14

\* Shell script to find the Fibonacci series up to n\*

```
echo "Enter the Range :"
read a
i=0; j=1
echo -e "Series is : \c"
while [ $a -ge 0 ]
do
    echo -e "$i \c"
    t=$i
    i=$((i+j))
    j=$((t+j))
    a=$((a-1))
done
echo -e "\n"
```

• Output:

\$ bash program14.sh

Enter Range:

10

Series is: 0 1 1 2 3 5 8 13 21 34 55

:-:

## Program - 15:

' Shell script to count the no of vowels in a string '

```

echo -e "Enter the string : \c"
read string
count=0
length=`expr "$string" : '.\*'`
for (( i=$length ; i>0 ; i-- ))
do
    ch=`echo $string | cut -c $i`
    case $ch in
        [aeiouAEIOU]) count=$((count+1));;
    esac
done
echo "The no of vowels is $count"

```

- Output:

\$ bash program15.sh

Enter the string: Good Morning

The no of vowels is 4

### Program - 16:

\* Shell script to check number of lines, words, characters in a file \*

```
echo -e "Enter the name of file:lc"
read fname
echo -e "Lines = `wc -l < $fname` \n"
echo -e "Words = `wc -w < $fname` \n"
echo -e "Characters = `wc -m < $fname` \n"
```

- Output:

```
$ bash program16.sh
```

```
Enter the name of file: program16.sh
```

```
Lines = 5
```

```
words = 35.
```

```
characters = 158
```

Program - 17 :

\* C/C++ program to output the contents of its environment list \*

```
#include <stdio.h>
int main (int argc, char *argv[])
{
    int i;
    char **ptr;
    extern char **environ;
    for (ptr = environ; *ptr != 0; ptr++)
        printf ("%s\n", *ptr);
    return 0;
}
```

- Output:

```
$ gcc program17.c -o program17
$ ./program17
SHELL = /bin/bash
SESSION-MANAGER = local[bnscce:@/tmp/.ICE-unix/1188,unix/bnscce
:/tmp/.ICE-unix/1188
QT_ACCESSIBILITY = 1
COLORTERM = truecolor
XDG_CONFIG_DIRS = /etc/xdg/xdg-ubuntu:/etc/xdg
SSH_AGENT_LAUNCHER = gnome-keyring
XDG_MENU_PREFIX = gnome-
GNOME_DESKTOP_SESSION_ID = this-is-deprecated
LANGUAGE = en_IN:en
GNOME_SHELL_SESSION_MODE = ubuntu
SSH_AUTH_SOCK = /run/user/1000/keyring/ssh
XMODIFIERS = @im=ibus
DESKTOP_SESSION = ubuntu
GTK_MODULES = gail:atk-bridge
PWD = /home/bnscce/Desktop
LOGNAME = bnscce
XDG_SESSION_DESKTOP = ubuntu
XDG_SESSION_TYPE = wayland
SYSTEMD_EXEC_PID = 2238
XAUTHORITY = /run/user/1000/.mutter-Xwaylandauth.SMS0G1
HOME = /home/bnscce
USERNAME = bnscce
IM_CONFIG_PHASE = 1
```

LANG=en\_IN

LS\_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01....

XDG\_CURRENT\_DESKTOP=ubuntu:GNOME

VTE\_VERSION=6402

WAYLOAD\_DISPLAY=wayload-0

G\_ENABLE\_DIAGNOSTIC=0

GNOME\_TERMINAL\_SCREEN=/org/gnome/terminal/screen/33d4-33-44d

c674643cab9

INVOCATION\_ID=5ddf5a4ee05e447313f5f07674ee0b8005

MANAGED\_ID=1052

GNOME\_SETUP\_DISPLAY=:1

LESSCLOSE=/usr/bin/lesspipe %s %s

OLDPWD=/home/bnsce/shell

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin

HOMESESSION=ubuntu

JOURNAL\_STREAM=8:48740

DISPLAY=:0

USER=bnsce

-:- Program - 18:

\* C/C++ Program to emulate the Unix ln  
command \*

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <string.h>

int main (int argc, char *argv[])
{
    if (argc < 3 || argc > 4 || (argc == 4 && strcmp(argv[1],
        "-s"))))
    {
        printf ("USAGE nos [-s] <org-file> <new-file> \n", argv[0]);
        return 1;
    }

    if (argc == 4)
    {
        if ((symlink(argv[2], argv[3])) == -1)
            printf ("Can't create symbolic link \n");
        else
            printf ("symbolic link created \n");
    }

    else
    {
        if ((link(argv[1], argv[2])) == -1)
            printf ("cannot create Hard link \n");
    }
}
```

Teacher's Signature: \_\_\_\_\_

```
else  
    printf (" Hard link created \n");  
}  
return 0;  
}
```

Teacher's Signature : \_\_\_\_\_

- Output:

```
$ gcc program18.c -o program18
$ ./program18
USAGE: ./program18 [-s] <org-file> <new-file>
$ ./program18 Hello.txt HardHello.txt
Hard link created
$ ./program18 -s Hello.txt softHello.txt
Symbolic link created.
```

:- Program - 19:

\* C POSIX compliant program to print the POSIX defined config options supported on given system \*

```
#define _POSIX_SOURCE
#define _POSIX_C_SOURCE 199309L
#include <stdio.h>
#include <unistd.h>
int main()
{
    #ifndef _POSIX_JOB_CONTROL
        printf(" System supports job control \n");
    #else
        printf(" System does not support job control \n");
    #endif
    #ifndef _POSIX_SAVED_IDS
        printf(" System supports saved set-UID, set-GID \n");
    #else
        printf(" System does not support saved set-UID, set-GID \n");
    #endif
    #ifndef _POSIX_CHOWN_RESTRICTED
        printf(" chown-restricted option is bad \n",
               "-POSIX_CHOWN_RESTRICTED");
    #else

```

Teacher's Signature : \_\_\_\_\_

```
printf(" System does not support chown_restricted option");  
#endif  
#ifdef _POSIX_NO_TRUNC  
printf(" Pathname trunc option is bad \n", _POSIX_NO_TRUNC);  
#else  
printf(" System does not support posix no trunc option");  
#endif  
#ifdef _POSIX_VDISABLE  
printf(" Disabled characters for terminal files are  
      %od \n", _POSIX_VDISABLE);  
#else  
printf(" System does not support _POSIX_VDISABLE \n");  
#endif  
return 0;  
}
```

• Output :

\$ gcc program19.c -o program19

\$ ./program19

System supports TIOControl

System supports saved set-UID, set-GID

chown-restricted option is 0

Pathname trunc option is 1

Disabled character for terminal files is 0

**Program -20:**

\* C program to demonstrate the IPC between reader & writer process using mkfifo, open, read, write and close api in program \*

```
#include <sys/types.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <string.h>
#include <errno.h>
#include <stdio.h>

int main(int argc, char* argv[])
{
    int fd;
    char buf[256];
    if (argc != 2 && argc != 3)
    {
        printf("USAGE : %s <file> [<arg>]\n", argv[0]);
        return 0;
    }

    if (argc == 2) // reader process
    {
        fd = open(argv[1], O_RDONLY | O_NONBLOCK);
        while (read(fd, buf, sizeof(buf))) {
            printf("%s", buf);
        }
    }
}
```

Teacher's Signature : \_\_\_\_\_

```
}

else           // writer process
{

    mkfifo(argv[1], S_IFIFO | S_IRWXU | S_IRWXG | S_IRWXO);
    fd = open(argv[1], O_WRONLY);
    write(fd, argv[2], strlen(argv[2]));
}

close(fd);
}
```

• Output:

Terminal-1: Write Process

```
$ gcc program20.c -o program20
$ ./program20
USAGE ./program20 <file> [loop]
$ ./program20 pipeNam "Good Morning"
    [waits for reader process]
$ bnsce@bnsce:$ [Ends after reader]
```

Terminal-2 : Reader Process

```
$ ./program20 pipeNam
Good Morning bnsce@bnsce:$
```