**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“JnanaSangama”, Belgaum -590014, Karnataka.**

****

**LAB REPORT**

**on**

**UNIX SHELL AND PROGRAMMING**

***Submitted by***

**Atharva Acharya (1BM20CS004)**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**



**B.M.S. COLLEGE OF ENGINEERING**

**(Autonomous Institution under VTU)**

**BENGALURU-560019**

**October-2022 to Feb-2023**

**B. M. S. College of Engineering,**

**Bull Temple Road, Bangalore 560019**

(Affiliated To Visvesvaraya Technological University, Belgaum)

**Department of Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the Lab work entitled “LAB COURSE **UNIX SHELL AND PROGRAMMING**” carried out by **Atharva Acharya (1BM20CS004),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **Unix Shell and Programming - (20CS5PCUSP)** work prescribed for the said degree.

**Dr.** **Dr. Jyothi S Nayak**

Assistant Professor Professor and Head

Department of CSE Department of CSE

BMSCE, Bengaluru BMSCE, Bengaluru

`

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Date** | **Experiment Title** | **Page No.** |
| **1.** |  | Shell script to find if the given year is leap or not |  |
| **2** |  | Shell script to find the area of a circle |  |
| **3** |  | Shell script to check whether the number is zero/ positive/ negative |  |
| **4** |  | Shell script to find the biggest of three numbers |  |
| **5** |  | Shell script to find the factorial of a number |  |
| **6** |  | Shell script to compute the gross salary of an employee |  |
| **7** |  | Shell script to convert the temperature Fahrenheit to Celsius |  |
| **8** |  | Shell script to perform arithmetic operations on given two numbers |  |
| **9** |  | Shell script to find the sum of even numbers up to n |  |
| **10** |  | Shell script to print the combinations of numbers 123 |  |
| **11** |  | Shell script to find the power of a number |  |
| **12** |  | Shell script to find the sum of n natural numbers |  |
| **13** |  | Shell script to display the pass class of a student |  |
| **14** |  | Shell script to find the Fibonacci series up to n |  |
| **15** |  | Shell script to count the number of vowels of a string |  |
| **16** |  | Shell script to check number of lines, words, characters in a file |  |
| **17** |  | Write a C/C++ program to that outputs the contents of its environment list |  |
| **18** |  | Write a C/C++ program to emulate the Unix ln command |  |
| **19** |  | Write a C/C++ POSIX compliant program that prints the POSIX defined Configuration options supported on any given system using feature test macros. |  |
| **20** |  | Write a C/C++ program which demonstrates Interprocess Communication between a reader process and a writer process. Use mkfifo, open, read, write and close apis in your program. |  |

Experiment No 1

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

#!/bin/bash

echo "Enter an Year: "

read year

if [ $((year % 4)) -eq 0 ]

then

if [ $((year % 100)) -eq 0 ]

then

if [ $((year % 400)) -eq 0 ]

then

echo "$year is a leap year"

else

echo "$year is not a leap year"

fi

else

echo "$year is a leap year"

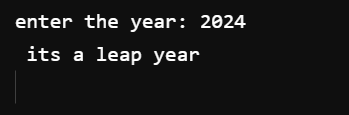
fi

else

echo "$year is not a leap year"

fi

**Output**



Experiment No 2

**Shell script to find the area of a circle**

#!/bin/bash

echo "\nEnter the radius of a circle : "

read r

d=$(echo "scale=2;2 \* $r"| bc) #Diameter

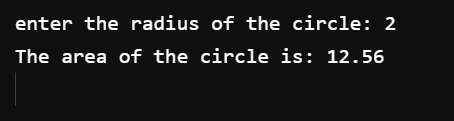
area=$(echo "scale=2; 22/7 \* ($r \* $r)" | bc)

circumference=$(echo "scale=2; 22/7 \* $d"| bc)

echo "\nArea of circle is : $area"

echo "\nCircumference of circle is : $circumference \n"

**Output**



Experiment No 3

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

#!/bin/bash

echo "Enter the number : "

read num

if [ $num -gt 0 ]

then

echo "$num is positive"

elif [ $num -lt 0 ]

then

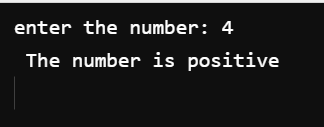
echo "$num is negative"

else

echo "$num is zero"

fi

**Output**



Experiment No 4

**Shell script to find the biggest of three numbers**

#!/bin/bash

echo "Enter first number : "

read num1

echo "Enter second number : "

read num2

echo "Enter third number : "

read num3

if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]

then

echo "\n$num1 is the greatest"

elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]

then

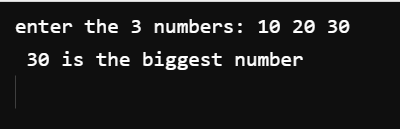
echo "\n$num2 is the greatest"

else

echo "\n$num3 is the greatest"

fi

**Output**



Experiment No 5

**Shell script to find the factorial of a number**

#!/bin/bash

echo "ENTER THE NUMBER: "

read n

fact=1

while [ $n -gt 1 ]

do

fact=$(( fact \* n))

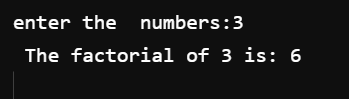
n=$((n-1 ))

done

echo "FACTORIAL IS: "

echo $fact

**Output**



Experiment No 6

**Shell script to compute the gross salary of an employee**

#!/bin/bash

echo "\nEnter name of Employee :"

read name

echo "\nEnter DA :"

read da

echo "\nEnter HRA:"

read hra

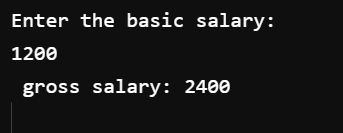
echo "\nEnter basic"

read basic

sal=$(( $da + $hra + $basic ))

echo "\nGross Salary of $name is $sal"

**Output**



Experiment No 7

**Shell script to convert the temperature Fahrenheit to Celsius**

#!/bin/bash

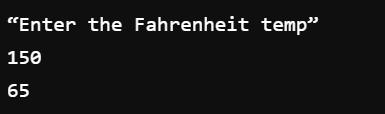
echo "Enter temperature in F : "

read f

c=$(echo "scale=2;(5/9)\*($f-32)"|bc)

echo "$f °F = $c °C"

**Output:**



Experiment No 8

**Shell script to perform arithmetic operations on given two numbers**

#!/bin/bash

echo "Enter 2 Numbers : "

read a

read b

echo "Enter Operation : \n"

echo "1) Addition"

echo "2) Subtraction"

echo "3) Multiplication"

echo "4) Division(Quotient)"

echo "5) Modulus(Remainder)\n"

read op

case $op in

1)echo "scale=3; $a + $b" | bc -l ;;

2)echo "scale=3; $a - $b" | bc -l ;;

3)echo "scale=3; $a \\* $b" | bc -l ;;

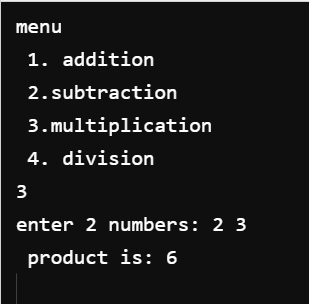
4)echo "scale=3; $a / $b" | bc -l ;;

5)echo "scale=3; $a % $b" | bc -l ;;

\*)echo "Choose a valid option"

esac

**Output**



Experiment No 9

**Shell script to find the sum of even numbers upto n**

**Program:**

#!/bin/bash

sum=0

read -p "Enter maximum limit of Even Numbers : " m

for ((i = 0; i < m; i++)); do

if [[ $i%2 -eq 0 ]]; then

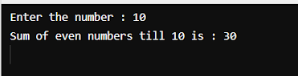
sum=$(expr $sum + $i)

fi

done

echo $sum

**Output**



Experiment No 10

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

**Program:**

#!/bin/bash

echo "Combinations for 123 :"

for ((i = 1; i <= 3; i++)); do

for ((j = 1; j <= 3; j++)); do

for ((k = 1; k <= 3; k++)); do

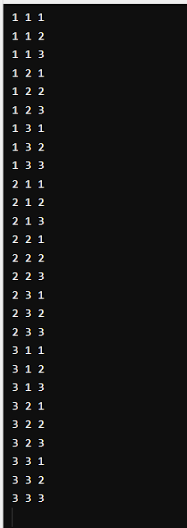
echo $i $j $k

done

done

done

**Output**



Experiment No 11

**Shell script to find the power of a number**

**Program:**

#!/bin/bash

echo "Enter base"

read a

echo "Enter power"

read b

res=1

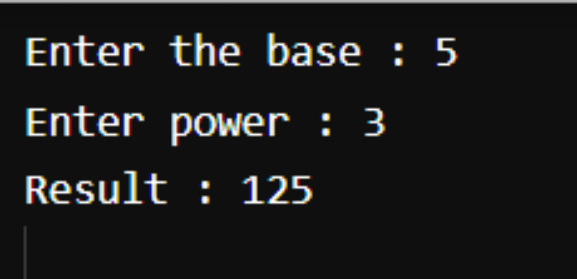
for ((i = 1; i <= b; i++)); do

res=`expr $res \\* $a`

done

echo $res

**Output**



Experiment No 12

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

**Program:**

#!/bin/bash

echo "Enter a number"

read n

i=1

sum=0

while [ $i -le $n ]

do

echo "$i"

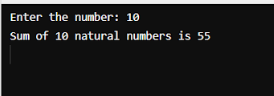
sum=$(( $sum + $i ))

i=$(( $i + 1 ))

done

echo "Sum=$sum"

**Output**



Experiment No 13

**Shell script to display the pass class of a student**

#!/bin/bash

echo "Enter m1:\c and Enter m2:\c "

read m1

echo "Enter m3:\c"

read m3

echo "Enter m4:\c"

read m4

echo "Enter m5:\c"

read m5

tot=`expr $m1 + $m2 + $m3 + $m4 + $m5`;

avg=`expr $tot / 5`;

echo "total : $tot \n avg : $avg"

if [ $avg -gt 85 ];then

echo " Grade: Distinction "

elif [ $avg -gt 65 ];then

echo " Grade: First Class "

elif [ $avg -gt 50 ];then

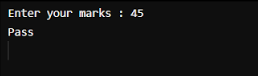
echo " Grade: Second Class "

elif [ $avg -gt 35 ];then

echo " Grade: Pass "

else echo " Grade: Fail"

fi



Experiment No 14

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

**Program:**

#!/bin/bash

read N

a=0

b=1

echo "The Fibonacci series is : "

for (( i=0; i<N; i++ ))

do

echo "$a"

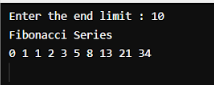
fib=$((a + b))

a=$b

b=$fib

done

**Output**



Experiment No 15

**Shell script to count the number of vowels of a string**

**Program:**

#!/bin/bash

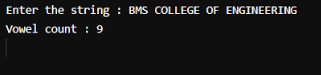
echo "enter filename"

read filename

vowels=`cat $filename | tr -cd 'aeiouAEIOU' | wc -c`

echo "Number of vowels in $filename: $vowels"

**Output**



Experiment No 16

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

#!/bin/bash

echo "Enter the filename or path to proceed"

read filename

words=`wc -w $filename`

lines=`wc -l $filename`

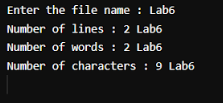
chars=`wc -c $filename`

echo "Words is $words"

echo "Lines is $lines"

echo "Characters is $chars"

**Output**



Experiment No 17

**Write a C/C++ program to that outputs the contents of its environment list**

#include&lt;stdio.h&gt;

#include&lt;unistd.h&gt;

int main(int argc,char \*argv[])

{

char \*\*ptr;

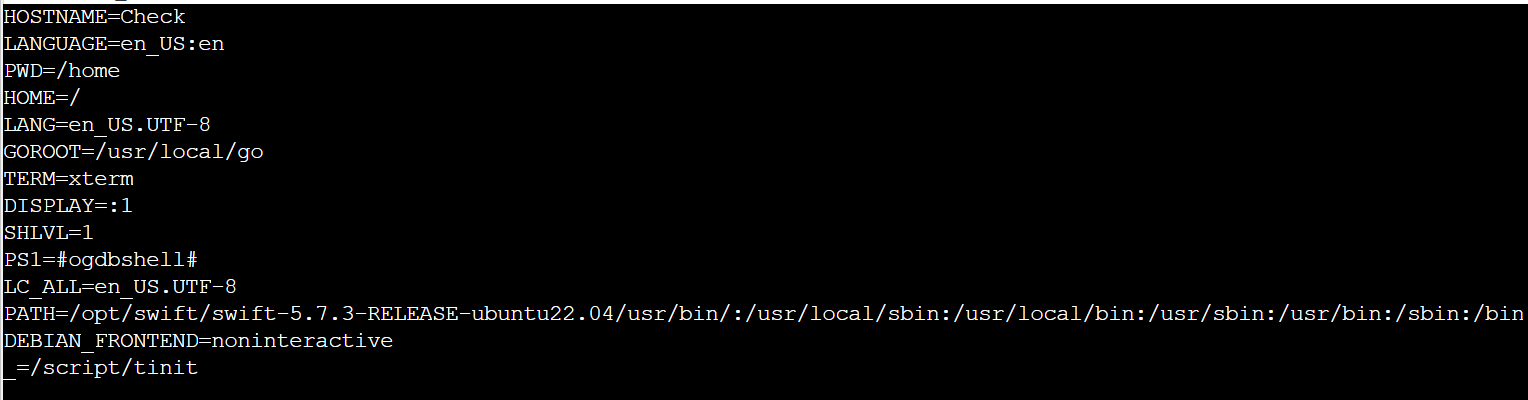
extern char \*\*environ;

for(ptr=environ; \*ptr; ptr++)

printf(&”%s\n&”,\*ptr);

return 0;

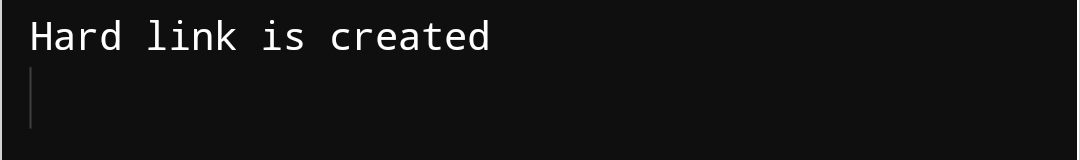
}



Experiment No 18

**Write a C/C++ program to emulate the Unix ln command**

#include<unistd.h>  
#include<stdio.h>  
#include<string.h>  
int main(int argc , char \* argv[])  
{  
        if(argc<3 || argc>4)  
        {  
                printf("Error in usage\n");  
                return -1;  
        }  
        if(argc==4 && strcmp(argv[1],"-s")!=0)  
        {  
                printf("for symbolic link use -s option");  
                return -1;  
        }  
        if(argc==4 && access(argv[2] , F\_OK)==-1)  
        {  
                printf("Source file does not exist");  
                return -1;  
        }  
        if(argc==3 && access(argv[1] , F\_OK)==-1)  
        {  
                printf("Source file does not exist");  
                return -1;  
        }  
        if(argc==4)  
        {  
                symlink(argv[2] , argv[3]);  
                printf("Symbolic link is created");  
                return 0;  
        }  
        if(argc==3)  
        {  
                link(argv[1] , argv[2]);  
                printf("Hard link is created");  
                return 0;  
        }  
}



Experiment No 19

**Write a C/C++ POSIX compliant program that prints the POSIX defined Configuration options supported on any given system using feature test macros.**

#define \_POSIX\_SOURCE

#define \_POSIX\_C\_SOURCE 199309L

#include&lt;iostream&gt;

#include&lt;unistd.h&gt;

int main()

{

using namespace std;

#ifdef \_POSIX\_JOB\_CONTROL

cout<<”System Supports Job Control feature”<< endl;

#else

cout<<”System does not support job control\n”;

#endif

#ifdef \_POSIX\_SAVED\_IDS

cout<<”System Supports saved set-UID and saved set-GID’’<<endl;

#else

cout<<“System doesnot support saved set-UID\n”;

#endif

#ifdef \_POSIX\_CHOWN\_RESTRICTED

cout<<“System Supports Change Ownership feature:”<<endl;

#else

cout<<“System doesnot support change Ownership feature\n”;

#endif

#ifdef \_POSIX\_NO\_TRUNC

cout<<“System Supports Path truncation option:”<<endl;

#else

cout<<“System doesnot support Path truncation \n”;

#endif

#ifdef \_POSIX\_VDISABLE

cout<<“System Supports Disable Character for files:”<<endl;

#else

cout<<“System doesnot support Disable Characters \n”;

#endif

return 0;

Experiment No 20

**Write a C/C++ program which demonstrates Interprocess Communication between a reader process and a writer process. Use mkfifo, open, read, write and close apis in your program.**

#include <sys/stat.h>

#include <string.h>

#include <fcntl.h>

#include <stdio.h>

#include <unistd.h>

int main(int argc, char \*argv[])

{

char buf[100];

int fd,n;

mkfifo (argv[1], S\_IFIFO |0777);

if (argc == 3){

fd = open (argv[1], O\_WRONLY);

write (fd, argv[2], strlen(argv[2]));

close(fd);}

if (argc ==2){

fd = open (argv[1], O\_RDONLY);

n= read (fd, buf, sizeof(buf));

buf[n]='\0';

printf ("%s", buf);

close(fd);

}

