

OPERATING SYSTEM LAB

Name: Thorat Amey Arun

Reg No.: 23MCS1004

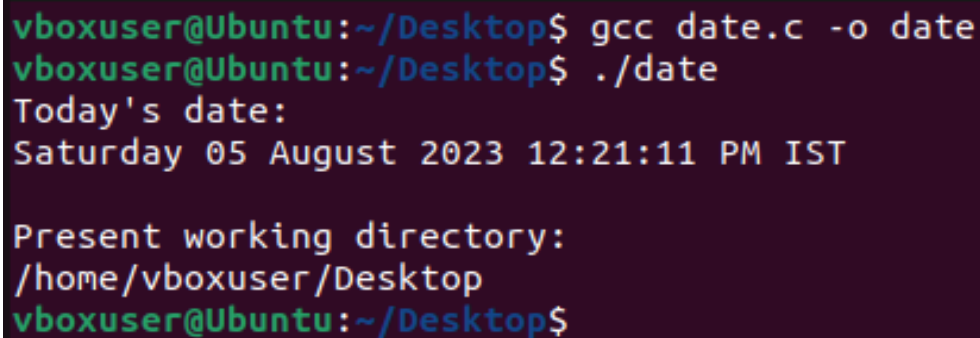
LAB EXPERIMENT 1

1. Develop a C program to display the following using system() library function (through which you can run linux commands from c program)
 - (i) today's date
 - (ii) present working directory

Program:

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    printf("Today's date:\n");
    system("date");
    printf("\nPresent working directory:\n");
    system("pwd");
    return 0;
}
```

Output:



```
vboxuser@Ubuntu:~/Desktop$ gcc date.c -o date
vboxuser@Ubuntu:~/Desktop$ ./date
Today's date:
Saturday 05 August 2023 12:21:11 PM IST

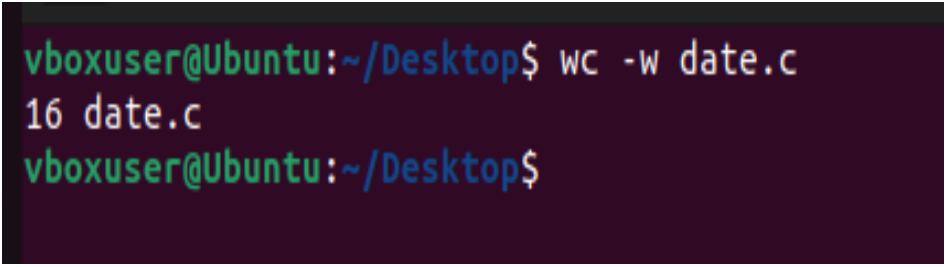
Present working directory:
/home/vboxuser/Desktop
vboxuser@Ubuntu:~/Desktop$
```

2. Display the word count of the file using
 1. appropriate linux command

OPERATING SYSTEM LAB

Name: Thorat Amey Arun

Reg No.: 23MCS1004



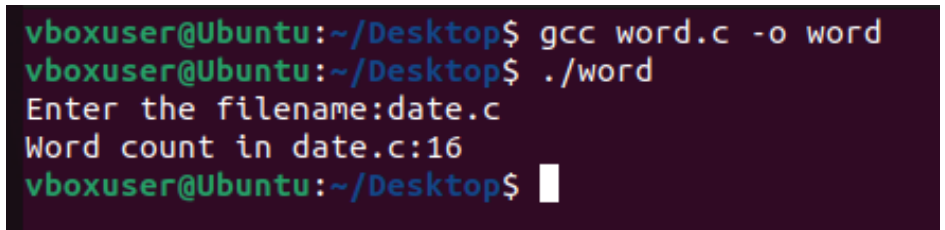
```
vboxuser@Ubuntu:~/Desktop$ wc -w date.c
16 date.c
vboxuser@Ubuntu:~/Desktop$
```

2. Write a C program to perform the count of words in the file.

Program:

```
#include <stdio.h>
int main() {
    char filename[100];
    printf("Enter the filename: ");
    scanf("%s", filename);
    FILE *file = fopen(filename, "r");
    if (file == NULL) {
        printf("Error opening the file.\n");
        return 1;
    }
    int wordCount = 0;
    char ch;
    while ((ch = fgetc(file)) != EOF) {
        if (ch == ' ' || ch == '\n' || ch == '\t') {
            wordCount++;
        }
    }
    printf("Word count in %s: %d\n", filename, wordCount);
    fclose(file);
    return 0;
}
```

Output:



```
vboxuser@Ubuntu:~/Desktop$ gcc word.c -o word
vboxuser@Ubuntu:~/Desktop$ ./word
Enter the filename:date.c
Word count in date.c:16
vboxuser@Ubuntu:~/Desktop$
```

OPERATING SYSTEM LAB

Name: Thorat Amey Arun


Reg No.: 23MCS1004

3. Write a C program to Read command-line arguments using argc and argv.

Program:

```
#include <stdio.h>
int main(int argc, char *argv[]) {
    printf("Number of arguments: %d\n", argc);
    printf("Arguments:\n");
    for (int i = 0; i < argc; i++) {
        printf("Argument %d: %s\n", i, argv[i]);
    }
    return 0;
}
```

Output:



```
vboxuser@Ubuntu:~/Desktop$ gcc argument.c -o argument
vboxuser@Ubuntu:~/Desktop$ ./argument arg1 arg2 arg3
Number of arguments:4
Argument 0:./argument
Argument 1:arg1
Argument 2:arg2
Argument 3:arg3
vboxuser@Ubuntu:~/Desktop$
```

4. Write a shell to reverse a number.

Program:

```
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)
```

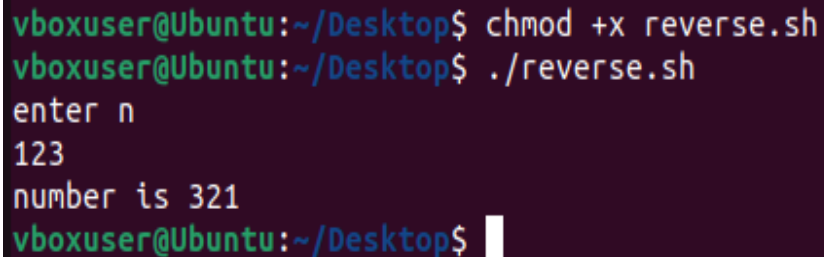
OPERATING SYSTEM LAB

Name: Thorat Amey Arun

Reg No.: 23MCS1004

```
done
echo number is $num
```

Output:



```
vboxuser@Ubuntu:~/Desktop$ chmod +x reverse.sh
vboxuser@Ubuntu:~/Desktop$ ./reverse.sh
enter n
123
number is 321
vboxuser@Ubuntu:~/Desktop$
```

5. Write a shell to find the greatest of three numbers.

Program:

```
#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 $num3 ]
then
    echo "the greatest no is: "$num1
elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]
then
    echo "the greatest no is: "$num2
else
    echo "the greatest no is: "$num3
fi
```

Output:

OPERATING SYSTEM LAB

Name: Thorat Amey Arun

Reg No.: 23MCS1004

```
vboxuser@Ubuntu:~/Desktop$ chmod +x great.sh
vboxuser@Ubuntu:~/Desktop$ ./great.sh
Enter Num1
12
Enter Num2
14
Enter Num3
10
the greatest no. is:14
vboxuser@Ubuntu:~/Desktop$
```