***Experiment 5***

***Aims:*Programs using the I/O system calls of Linux operating system( open, read, write etc).**

**Introduction :**

1. **Read :** From the file indicated by the file descriptor fd, the read() function reads cnt bytes of input into the memory area indicated by buf. A successful read() updates the access time for the file.

**Syntax:**

size\_t read (int fd, void\* buf, size\_t cnt);

**Program:**

**#include<stdio.h>**

**int main ()**

**{**

**int c;**

**FILE \*fptr;**

**fptr=fopen("text.txt","r");**

**if (fptr)**

**{**

**printf("file read....");**

**while((c=getc(fptr))!=EOF)**

**putchar(c);**

**fclose(fptr);**

**}**

**else**

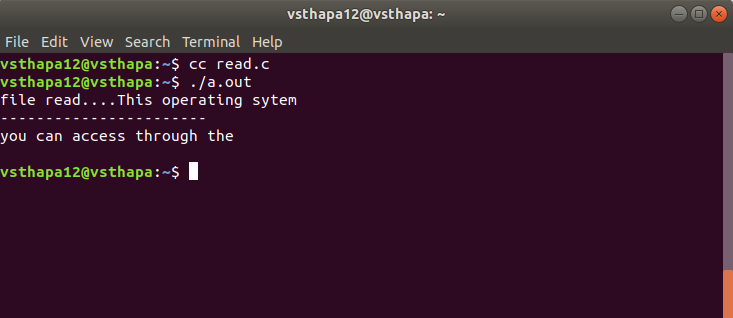
**{**

**printf("error");**

**}**

**return 0;}**

**Example:**



ii. Write:Writes cnt bytes from buf to the file or socket associated with fd. cnt should not be greater than INT\_MAX (defined in the limits.h header file). If cnt is zero, write() simply returns 0 without attempting any other action.

**Syntax**

size\_t write (int fd, void\* buf, size\_t cnt);

**Example**

#include<stdio.h>

**i**nt main ()

{

int c;

FILE \*fptr;

fptr=fopen("text.txt","w");

if (fptr)

{

printf("file read....");

while((c=getc(fptr))!=EOF)

putchar(c);

fclose(fptr);

}

else

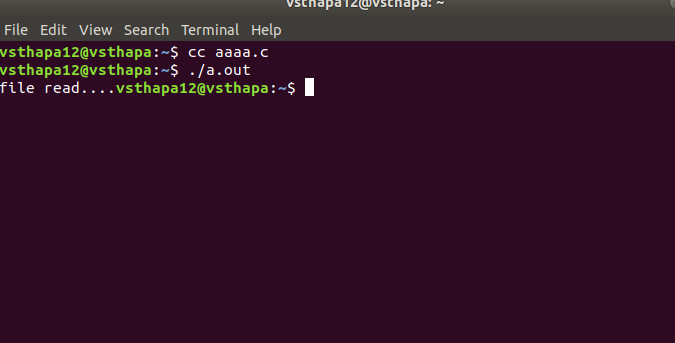
{

printf("error");

}

return 0;}

**Output:**

****