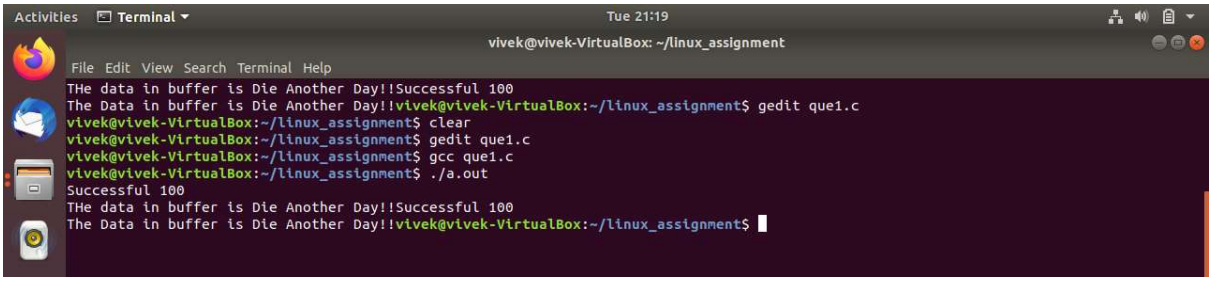
LINUX ASSIGNMENT: 1

1. **Write a program using file operations that demonstrates copying of data from input file and write into output file, until reaches end of file data.**

Code :

```
#include<stdio.h>
#include<fcntl.h>
#include<unistd.h>

int main()
{
 int fd1,fd2,len,pen;
 char rit_buf[100]="Die Another Day!!";
 char red_buf[100];

 fd1=open("/home/vivek/linux_assignment/data2.txt",O_CREAT | O_RDWR,0777);
 len=write(fd1,rit_buf,100);
 printf("Successful %d\n",len);

 lseek(fd1,0,SEEK_SET);
 read(fd1,red_buf,len);
 printf("The data in buffer is %s",red_buf);
 close(fd1);

 fd2=open("/home/vivek/linux_assignment/data2.txt",O_CREAT | O_RDWR,0777);
 pen=write(fd2,red_buf,100);
 printf("Successful %d\n",pen);
 lseek(fd2,0,SEEK_SET);
 read(fd2,red_buf,len);
 printf("The Data in buffer is %s",red_buf);
 close(fd2);
 return 0;
}
```

Output:



2. **Write a program that demonstrates repositioning of file offset using SEEK_SET, SEEK_END and SEEK_END.**

Code:

```
#include<stdio.h>
#include<fcntl.h>
#include<unistd.h>
int main()
{
 int fd1;
 char k[50];
 fd1=open("/home/vivek/linux_assignment/data1.txt",O_RDONLY,0775);
 lseek(fd1,0,SEEK_END);
 read(fd1,k,30);
 printf("%s\n",k);

 lseek(fd1,0,SEEK_SET);
 read(fd1,k,17);
 printf("%s\n",k);

 lseek(fd1,10,SEEK_CUR);
 read(fd1,k,7);
 printf("%s\n",k);

 close(fd1);

 return 0;

}
```

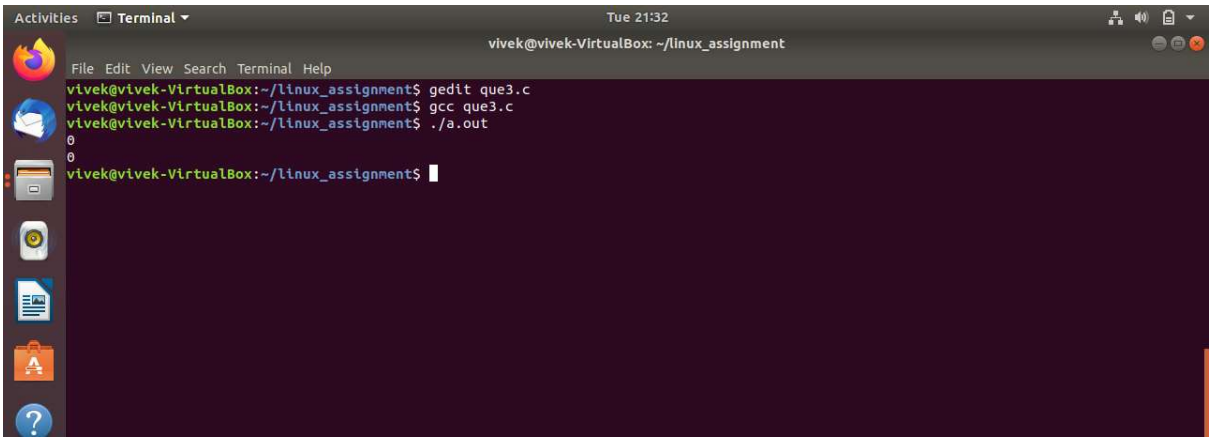
Output:

3. **Write program that returns “ls -l” kind of structure of information from an existing file or opend file.**

Code:

```
#include<stdio.h>
#include<unistd.h>
#include<fcntl.h>
#include<sys/stat.h>
#include<sys/types.h>
int main()
{
 int f1;
 struct stat st;
 f1 =open ("input.txt",O_RDONLY,0775);
 stat("inpur.txt",&st);
 printf("%lu\n",st.st_size);
 printf("%lu\n",st.st_ino);
 return 0;

}
```

Output:

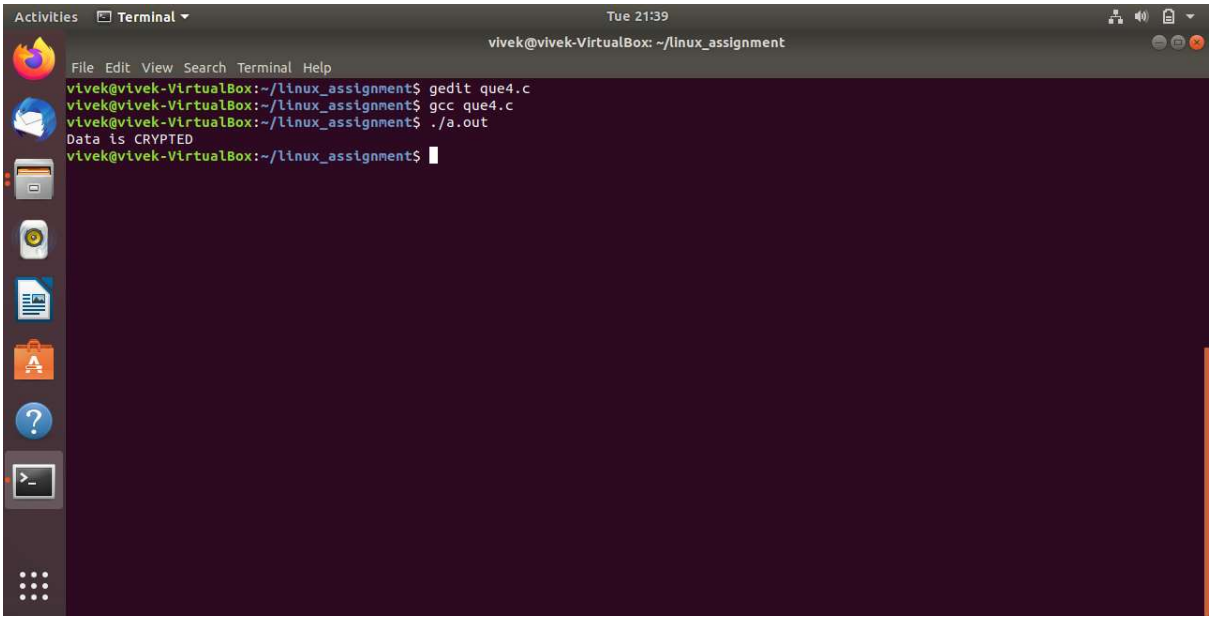


4. **Write a program that implements all file operations (open/creat/write/read/ Lseek/close).**

Code:

```
#include<stdio.h>
#include<fcntl.h>
#include<unistd.h>

int main()
{
 int fd1;
 char k[50]="ENCRYPTED";
 char p[50];
 creat("Lib.txt",0755);
 fd1=open("Lib.txt",O_RDWR,0775);
 write(fd1,k,50);
 close(fd1);
 fd1=open("Lib.txt",O_RDWR,0775);
 lseek(fd1,2,SEEK_CUR);
 read(fd1,p,30);
 printf("Data is %s\n",p);
 close(fd1);
 return 0;

}
```

Output:

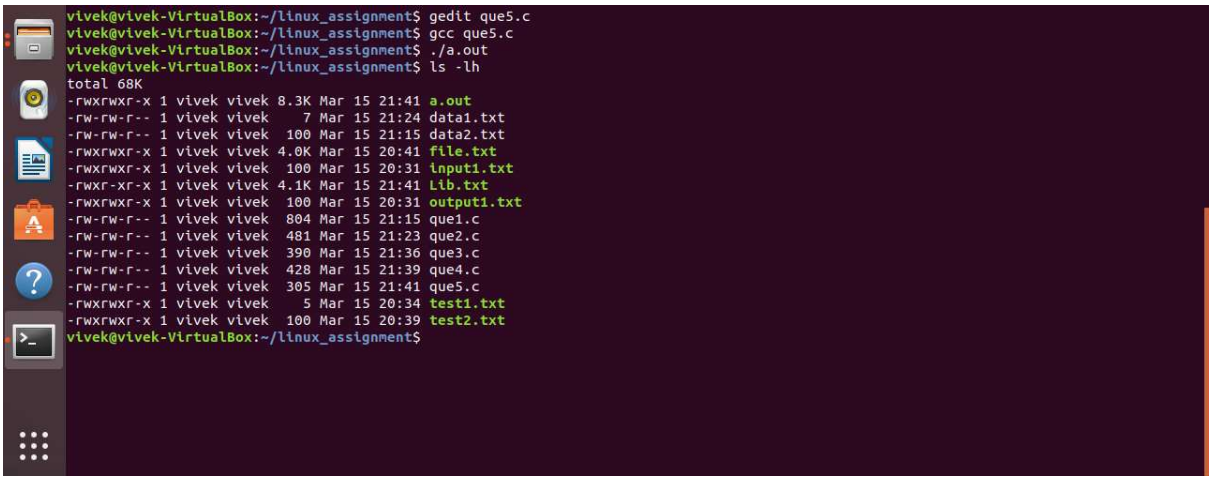


5. **Write a program that creates a file with a 4K bytes free space. (Such files are called files with holes.)**

Code:

```
#include<stdio.h>
#include<fcntl.h>
#include<unistd.h>
int main()
{
 int fd1;
 fd1=open("Lib.txt",O_RDWR,0775);
 lseek(fd1,4096,SEEK_END);
 write(fd1,"kdc",3);

 close(fd1);

 return 0;

}
```

Output:



GITHUB LINK : https://github.com/VIVEK0014/Linux_internals.git