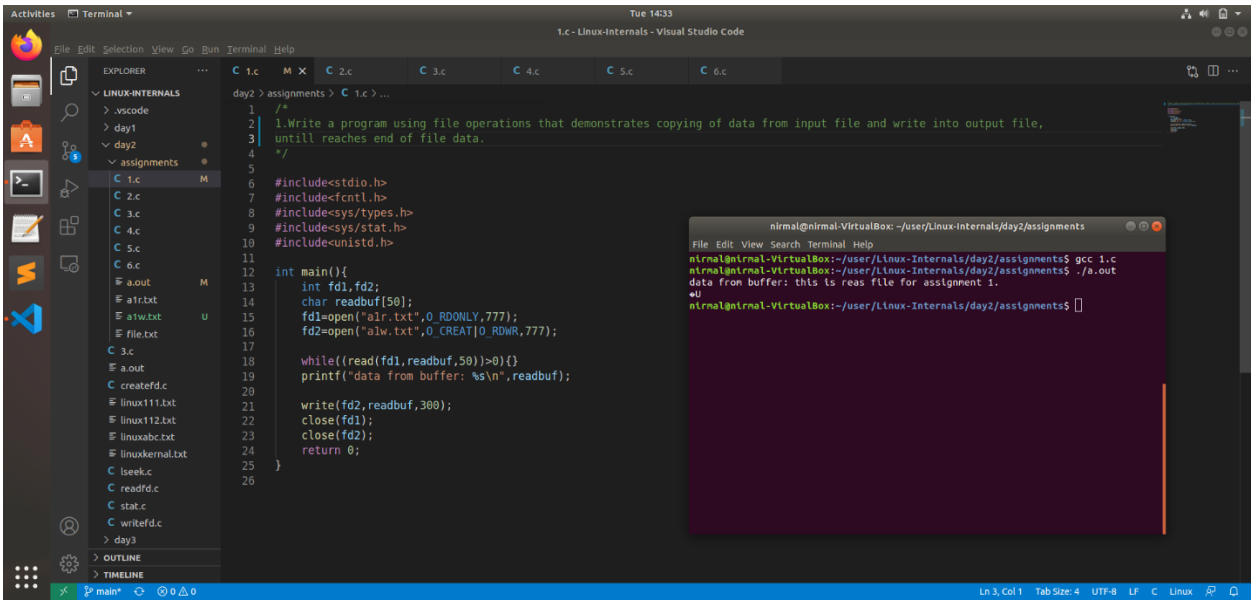
```
/*
1.Write a program using file operations that demonstrates copying of data from input file and write into output file, untill reaches end of file data.
*/
#include<stdio.h>
#include<fcntl.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<unistd.h>
int main(){
	int fd1,fd2;
	char readbuf[50];
	fd1=open("a1r.txt",O_RDONLY,777);
	fd2=open("a1w.txt",O_CREAT|O_RDWR,777);
	while((read(fd1,readbuf,50))>0){}
	printf("data from buffer: %s\n",readbuf)
	write(fd2,readbuf,300);
	close(fd1);
	close(fd2);
	return 0;
```

}



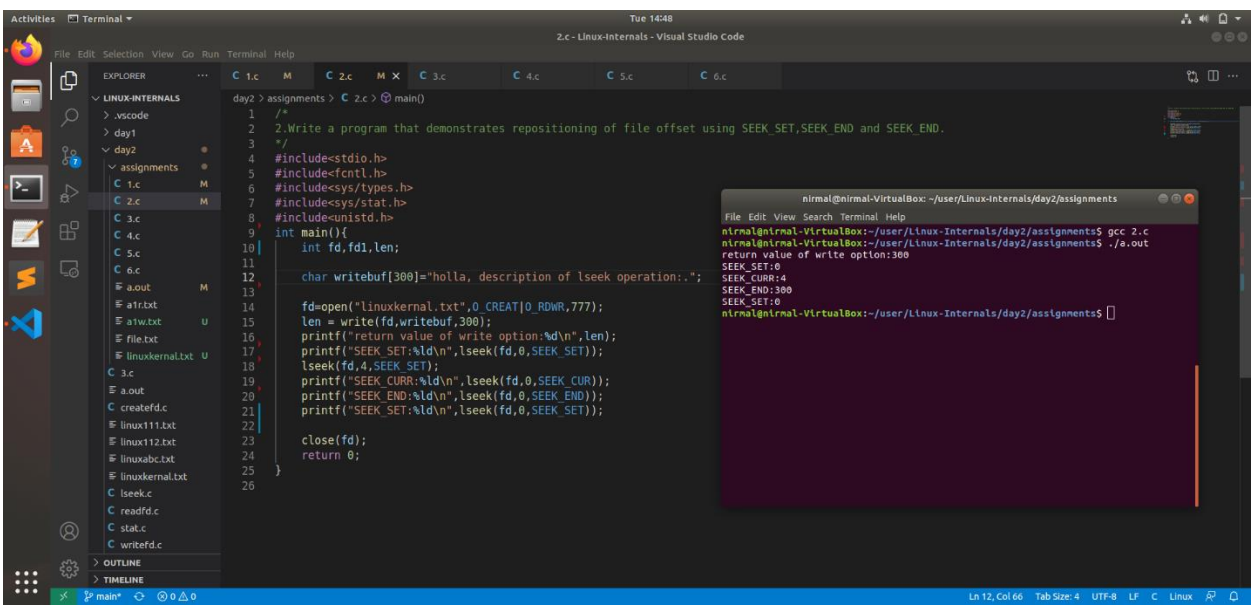
/*

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

*/

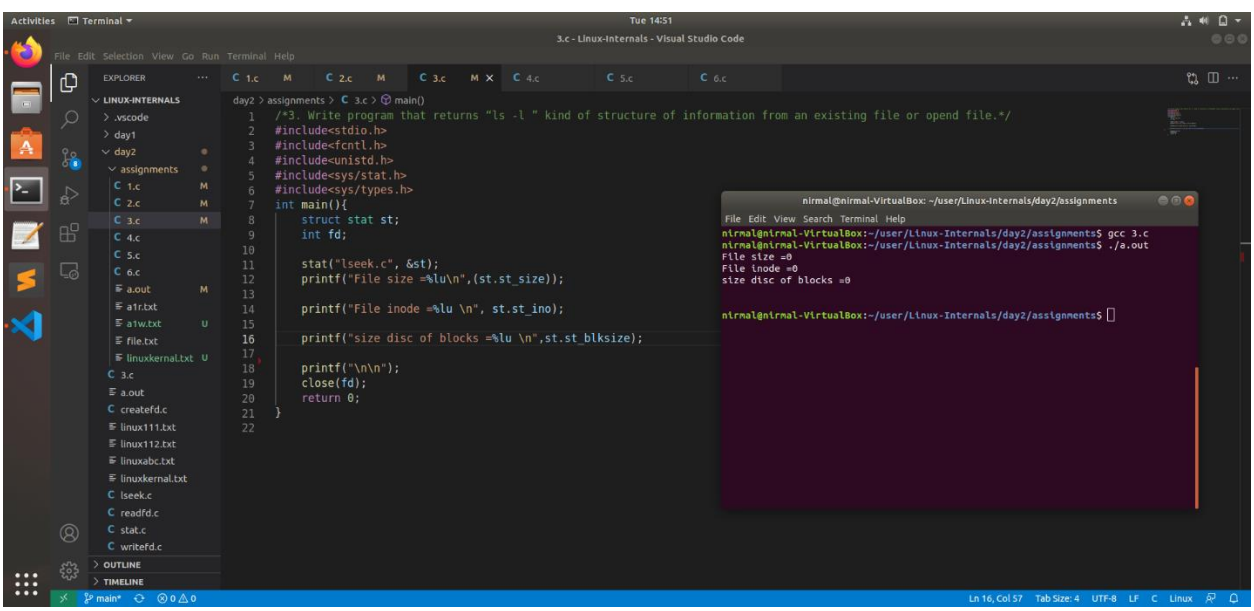
```
#include<stdio.h>

#include<fcntl.h>

#include<sys/types.h>

#include<sys/stat.h>

#include<unistd.h>

int main(){

        int fd,fd1,len;


        char writebuf[300]="holla, description of lseek operation:.";


        fd=open("linuxkernal.txt",O_CREAT|O_RDWR,777);
```

```
        len = write(fd,writebuf,300);
        printf("return value of write option:%d\n",len);
        printf("SEEK_SET:%ld\n",lseek(fd,0,SEEK_SET));
        lseek(fd,4,SEEK_SET);
        printf("SEEK_CURR:%ld\n",lseek(fd,0,SEEK_CUR));
        printf("SEEK_END:%ld\n",lseek(fd,0,SEEK_END));
        printf("SEEK_SET:%ld\n",lseek(fd,0,SEEK_SET));

        close(fd);
        return 0;
}
```



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

```
#include<stdio.h>

#include<fcntl.h>

#include<unistd.h>

#include<sys/stat.h>

#include<sys/types.h>

int main(){

        struct stat st;

        int fd;

        stat("lseek.c", &st);

        printf("File size =%lu\n",(st.st_size));

        printf("File inode =%lu \n", st.st_ino);

        printf("size disc of blocks =%lu \n",st.st_blksize);

        printf("\n\n");

        close(fd);

        return 0;

}
```



```
/*
5.Write a program that creates a file with a 4K bytes free space. (Such files are called files with holes.)
*/
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include<stdio.h>
char buf1[]="LAB ";
char buf2[]="OS Linux";
int main(){
        int fd;
        if ((fd=creat("file.txt", 0666)) < 0) {
                printf("Creation error");
                exit(1);
        }
        if (write(fd, buf1, sizeof(buf1)) < 0){
           printf("Writing error");
           exit(2);
        }
        if (lseek(fd, 4096, SEEK_SET) < 0){
           printf("Positioning error");
           exit(3);
        }
        if (write(fd, buf2, sizeof(buf2)) < 0){
           printf("Writing error");
           exit(2);
        }
}
```

/*

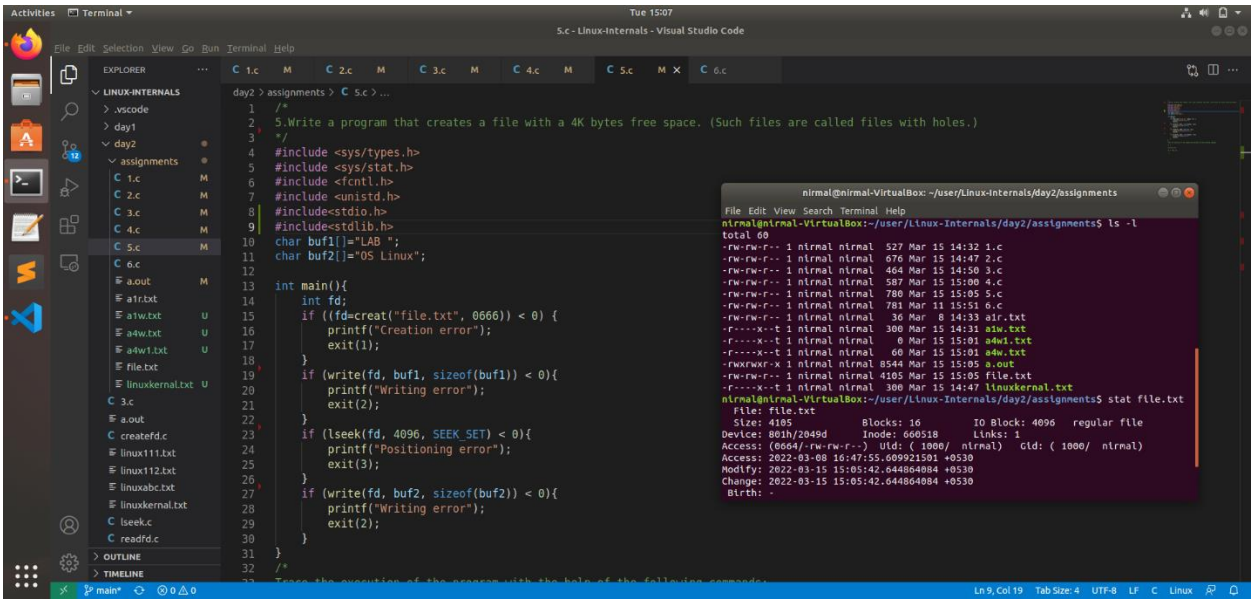
Trace the execution of the program with the help of the following commands:

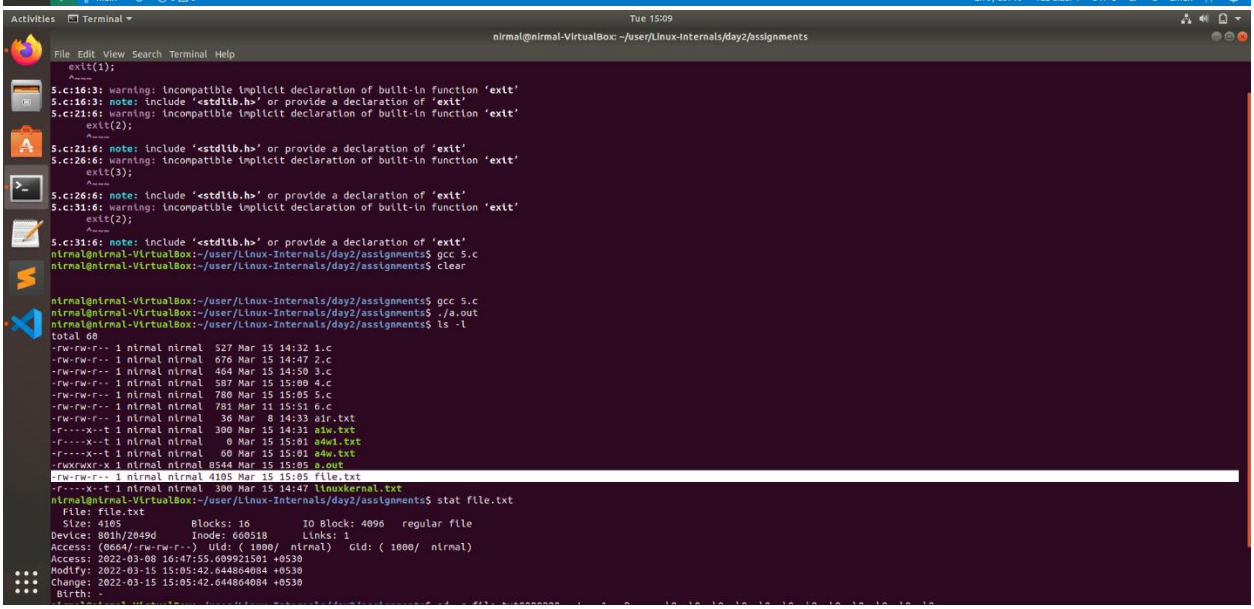
ls -l

stat file.txt

od -c file.txt

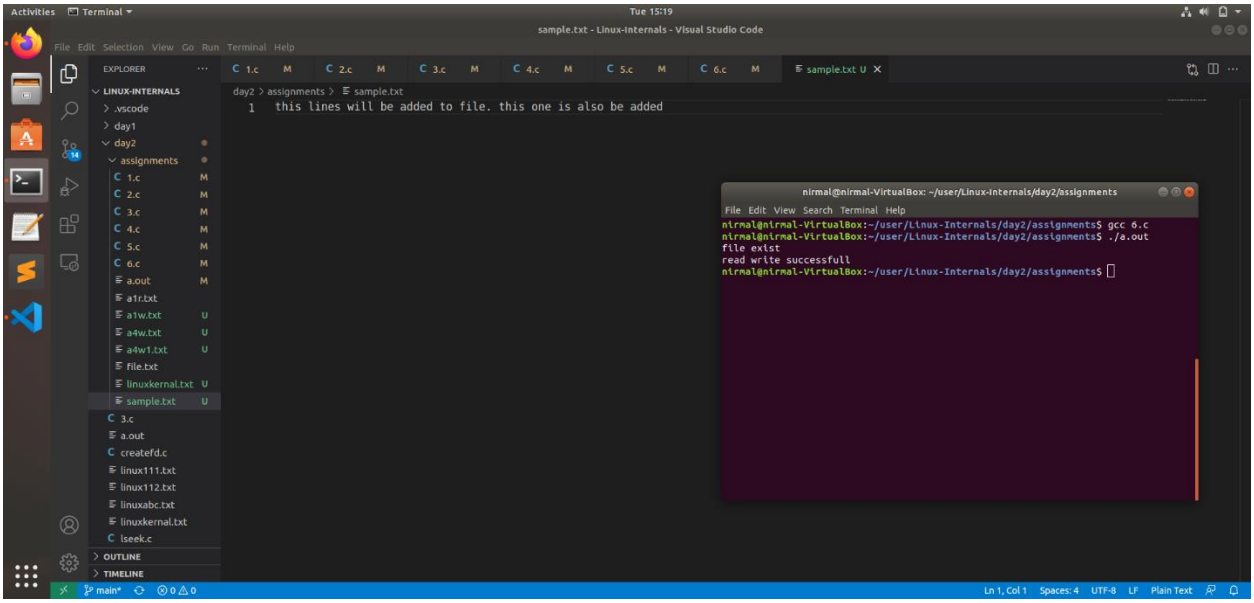
*/





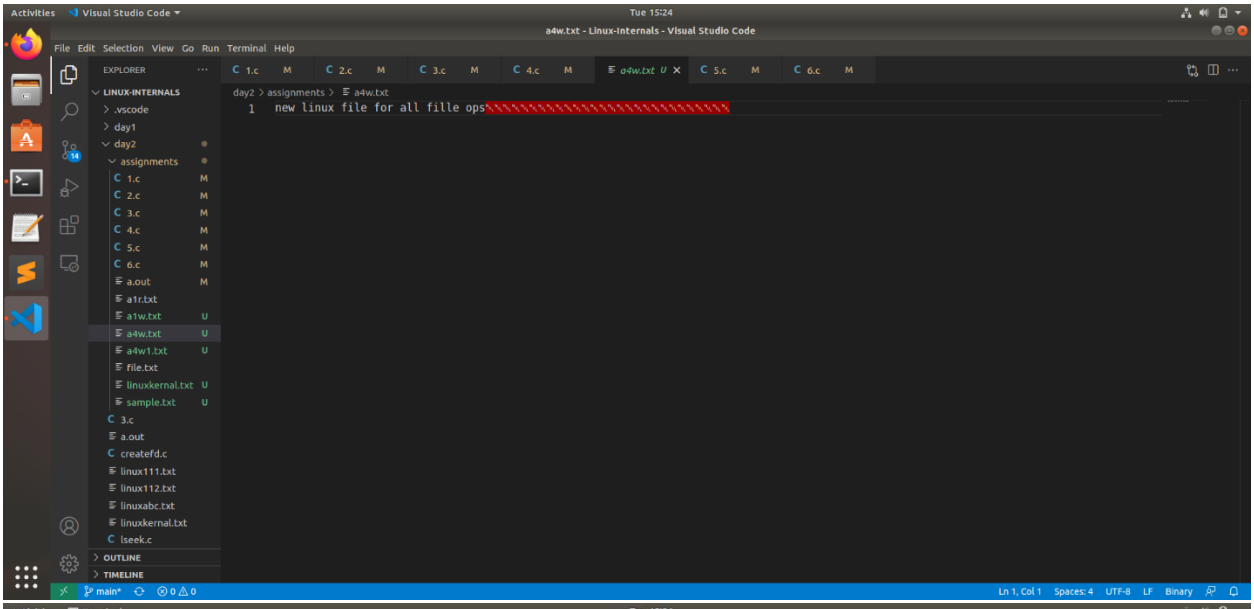
```
/*
6.write a program that used to check for the existance of the file and also check whether we can open file for read,write,execute or not.
*/
#include<stdio.h>
#include<fcntl.h>
#include<unistd.h>
#include<stdlib.h>
#include<string.h>
int main(){
   int fd4;
   fd4 = open("sample.txt", O_RDWR, 777); //to read the file(in readonly mode)
   if (fd4>0) { //checking file exist
      printf("file exist\n");
      char to_write[] = "this lines will be added to file. this one is also be added";
      int leng = strlen(to_write);
      char to_read[leng];
      if(write(fd4, to_write, leng)<0){ // writing permision in file
         printf("failed to write\n");
      }
      else if(read(fd4, to_read, leng)<0){ //reading permision in file
         printf("unable to read the file\n");
      }
      else{
         printf("read write successfull\n");
      }
   }
   else{
      printf("not exists\n");
```

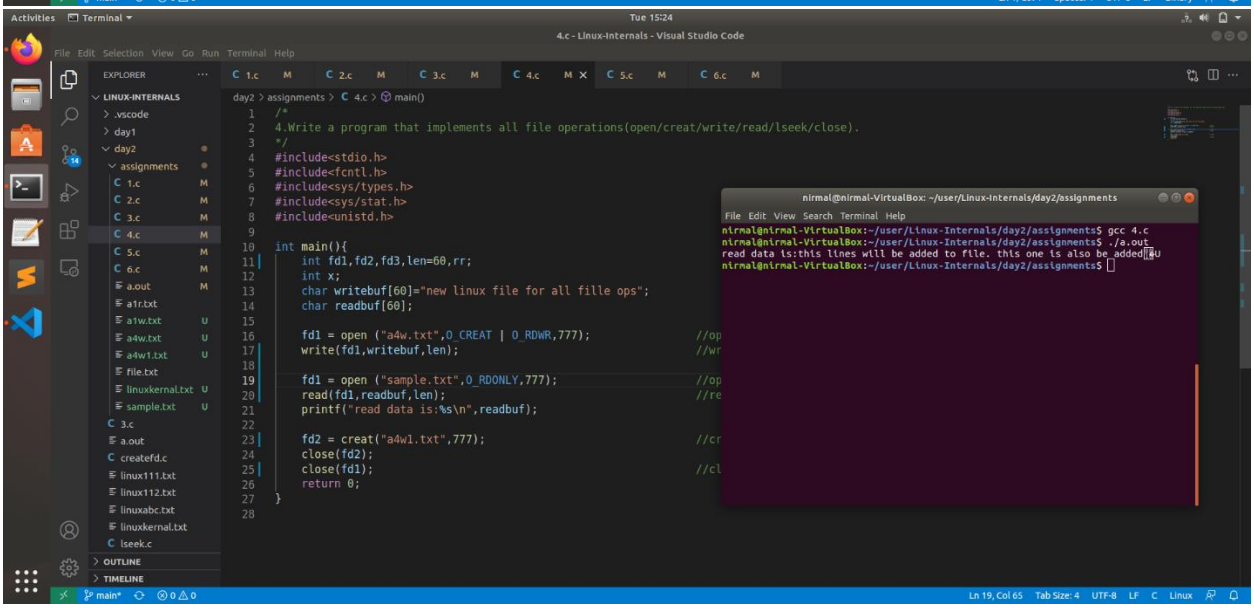
```
}
   int close(int fd4);
   return 0;
}
```

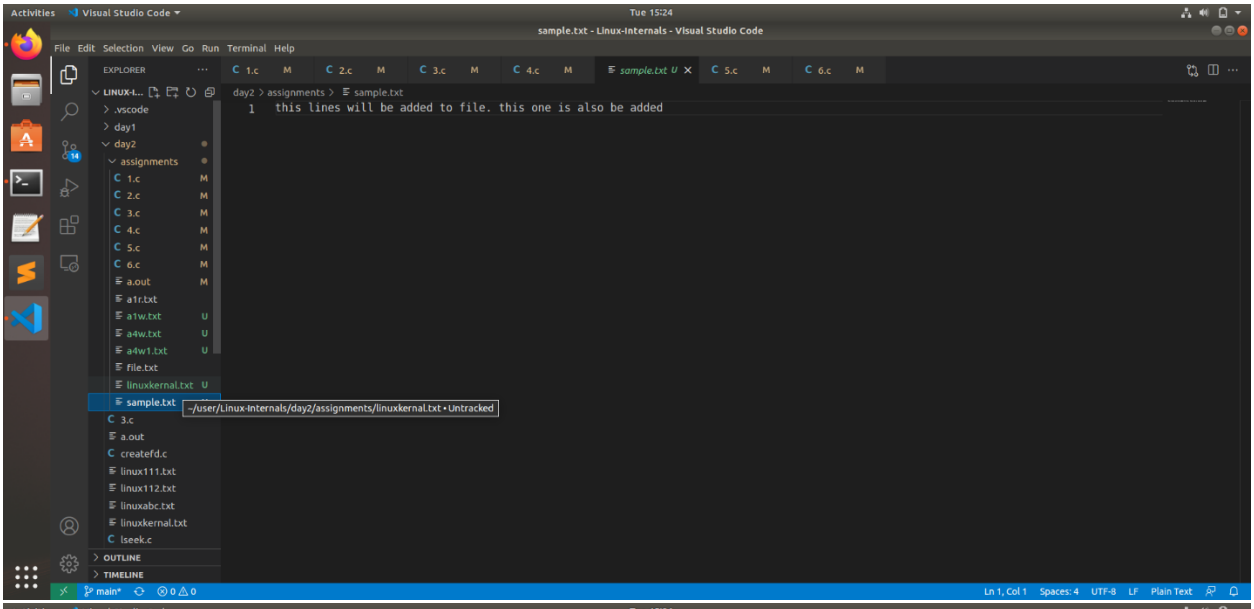


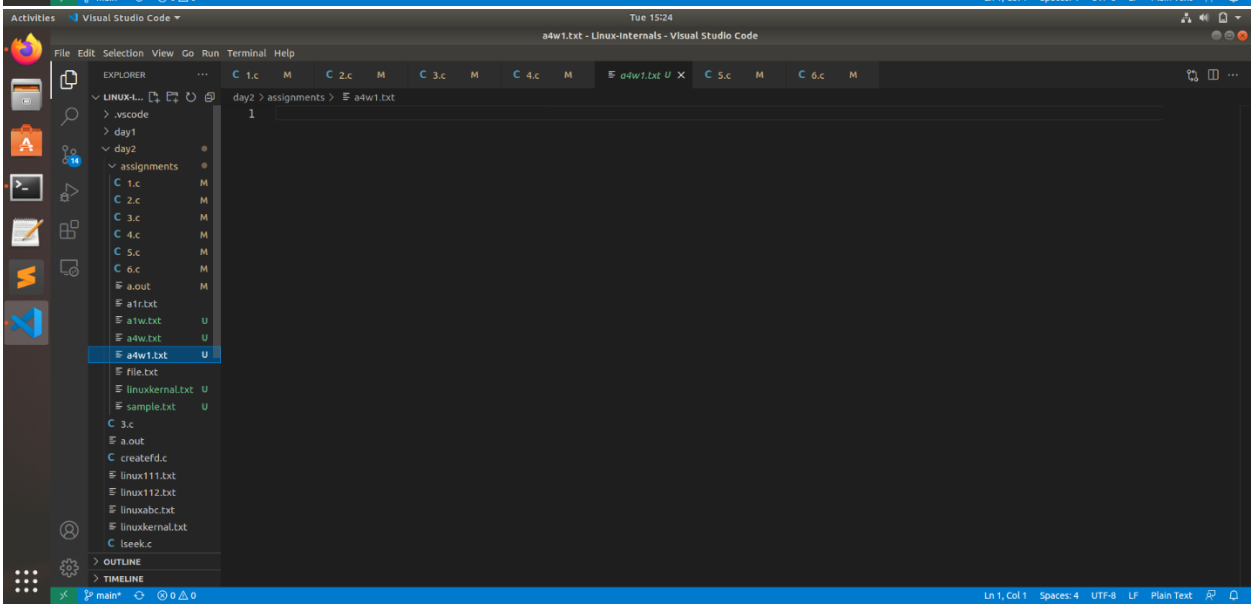

```
/*
4.Write a program that implements all file operations(open/creat/write/read/lseek/close).
*/
#include<stdio.h>
#include<fcntl.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<unistd.h>

int main(){
        int fd1,fd2,fd3,len=60,rr;
        int x;
        char writebuf[60]="new linux file for all fille ops";
```

```
        char readbuf[60];

        fd1 = open ("a4w.txt",O_CREAT | O_RDWR,777);                    //open
        write(fd1,writebuf,len);                                        //write

        fd1 = open ("sample.txt",O_RDONLY,777);                         //open
        read(fd1,readbuf,len);
        //read
        printf("read data is:%s\n",readbuf);

        fd2 = creat("a4w1.txt",777);                                    //creat
        close(fd2);
        close(fd1);
                //close
        return 0;
}
```


Activities
Visual Studio Code
Tue 15:24
a4w.txt - Linux-Internals - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
LINUX-INTERNALS
.vscode
day1
day2
assignments
1.c
2.c
3.c
4.c
5.c
6.c
a.out
a1r.txt
a1w.txt
a4w.txt
a4w1.txt
File.txt
linuxkernal.txt
sample.txt
3.c
a.out
createfd.c
linux111.txt
linux112.txt
linuxabc.txt
linuxkernal.txt
lseek.c
OUTLINE
TIMELINE
day2 > assignments > a4w.txt
new linux file for all fille ops%%%%%%%%%%%%%%%%%%%%%%%%%%%%
main*
Ln 1, Col 1
Spaces: 4
UTF-8
LF
Binary


Activities
Terminal
Tue 15:24
4.c - Linux-Internals - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
LINUX-INTERNALS
.vscode
day1
day2
assignments
1.c
2.c
3.c
4.c
5.c
6.c
a.out
a1r.txt
a1w.txt
a4w.txt
a4w1.txt
File.txt
linuxkernal.txt
sample.txt
3.c
a.out
createfd.c
linux111.txt
linux112.txt
linuxabc.txt
linuxkernal.txt
lseek.c
OUTLINE
TIMELINE
day2 > assignments > 4.c > main()
/*
4.Write a program that implements all file operations(open/creat/write/read/lseek/close).
*/
#include<stdio.h>
#include<fcntl.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<unistd.h>
int main(){
int fd1,fd2,fd3,len=60,rr;
int x;
char writebuf[60]="new linux file for all fille ops";
char readbuf[60];
fd1 = open ("a4w.txt",O_CREAT | O_RDWR,777);
write(fd1,writebuf,len);
fd1 = open ("sample.txt",O_RDONLY,777);
read(fd1,readbuf,len);
printf("read data is:%s\n",readbuf);
fd2 = creat("a4w1.txt",777);
close(fd2);
close(fd1);
return 0;
}
nirmal@nirmal-VirtualBox: ~/user/Linux-Internals/day2/assignments
File Edit View Search Terminal Help
nirmal@nirmal-VirtualBox:~/user/Linux-Internals/day2/assignments$ gcc 4.c
nirmal@nirmal-VirtualBox:~/user/Linux-Internals/day2/assignments$ ./a.out
read data is:this lines will be added to file. this one is also be added%%
nirmal@nirmal-VirtualBox:~/user/Linux-Internals/day2/assignments$
main*
Ln 19, Col 65
Tab Size: 4
UTF-8
LF
C
Linux


Activities
Visual Studio Code
Tue 15:24
sample.txt - Linux-Internals - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
1.c M
2.c M
3.c M
4.c M
sample.txt U
5.c M
6.c M
day2 > assignments > sample.txt
1 this lines will be added to file. this one is also be added
LINUX-I...
.vscode
day1
day2
assignments
1.c M
2.c M
3.c M
4.c M
5.c M
6.c M
a.out M
a1r.txt
a1w.txt U
a4w.txt U
a4w1.txt U
File.txt
linuxkernal.txt U
sample.txt U
~/user/Linux-Internals/day2/assignments/linuxkernal.txt • Untracked
3.c
a.out
createfd.c
linux111.txt
linux112.txt
linuxabc.txt
linuxkernal.txt
lseek.c
OUTLINE
TIMELINE
main*
0 0
Ln 1, Col 1
Spaces: 4
UTF-8
LF
Plain Text


Activities
Visual Studio Code
Tue 15:24
a4w1.txt - Linux-Internals - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
1.c M
2.c M
3.c M
4.c M
a4w1.txt U
5.c M
6.c M
day2 > assignments > a4w1.txt
1
LINUX-I...
.vscode
day1
day2
assignments
1.c M
2.c M
3.c M
4.c M
5.c M
6.c M
a.out M
a1r.txt
a1w.txt U
a4w.txt U
a4w1.txt U
File.txt
linuxkernal.txt U
sample.txt U
3.c
a.out
createfd.c
linux111.txt
linux112.txt
linuxabc.txt
linuxkernal.txt
lseek.c
OUTLINE
TIMELINE
main*
0 0
Ln 1, Col 1
Spaces: 4
UTF-8
LF
Plain Text