lseek () System Call

Name of the system Call:

lseek() – lseek system call is used to read and write the files randomly (not sequential).

Description:

lseek() system call is used to change the location of read and write pointers of the file descriptor. The location can be set either in relative or absolute terms. It is useful to be able to read or write to a specific offset with in the file.

Details of lseek system call():

lseek() system call is called by passing the file descriptor and offset as a parameter.
File descriptor is an abstract used to access a file or other input and output resources.
Offset is used to position the file offset within a particular location within the file.
Both the file descriptor and the offset is of type integer.

lseek(Filedescriptor,Offset);

we can have a third parameter (whence) in the lseek() system call mentioning from where we need to do the offset.

If whence is SEEK_SET, the offset is set to offset bytes.

If whence is SEEK_CUR, the offset is set to its current location plus offset bytes.

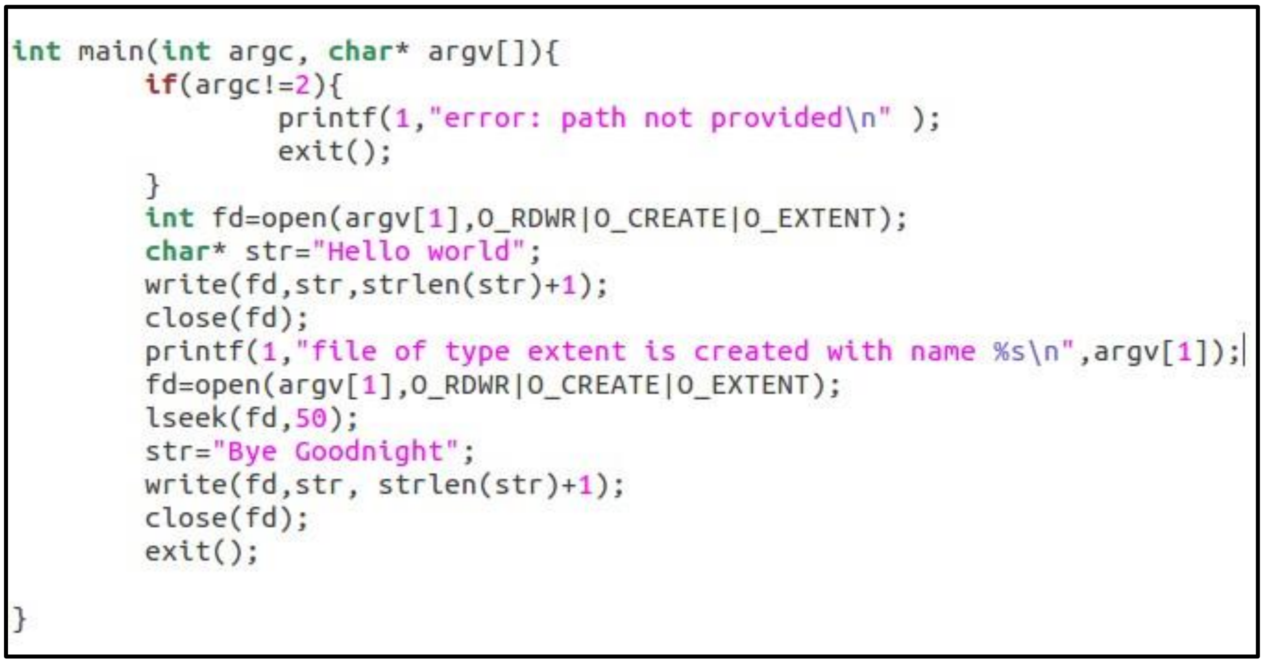
If whence is SEEK_END, the offset is set to the size of the file plus offset bytes.

Steps to perform lseek() system call:

- Open a file using the open () system call.
- It returns a pointer to the file (File descriptor).
- Using file descriptor write something in the file by calling write system call.
- Then close the file by using close system call and passing file descriptor as parameter.
- Then again open the same file.

- Call lseek() system call with the file descriptor and offset.
- Write something again into the file.
- Now close the file.

□ Check the output by using the cat system call.



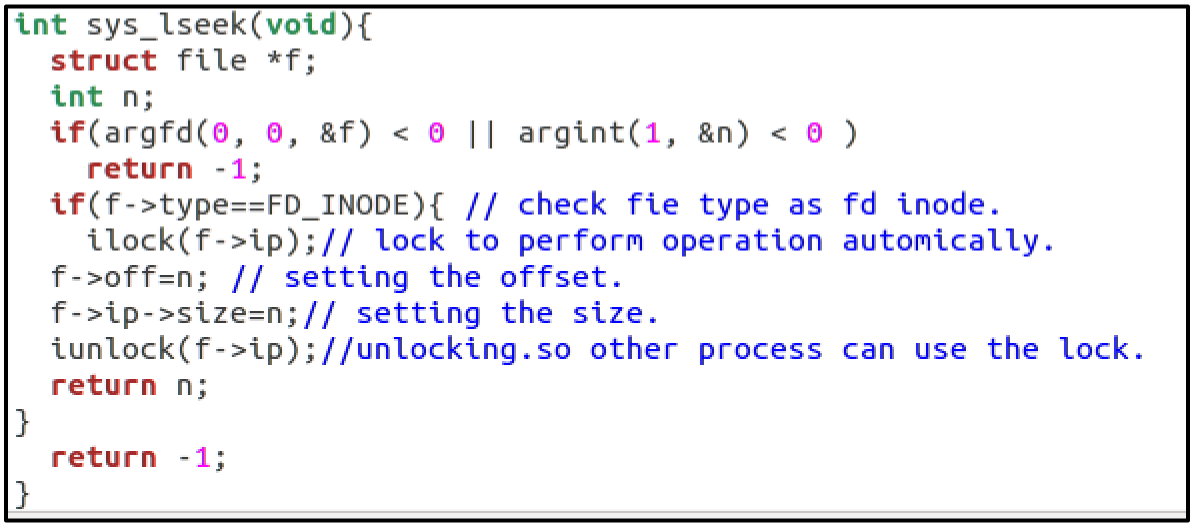
```
int main(int argc, char* argv[]){
        if(argc!=2){
                printf(1,"error: path not provided\n" );
                exit();
        }
        int fd=open(argv[1],O_RDWR|O_CREATE|O_EXTENT);
        char* str="Hello world";
        write(fd,str,strlen(str)+1);
        close(fd);
        printf(1,"file of type extent is created with name %s\n",argv[1]);
        fd=open(argv[1],O_RDWR|O_CREATE|O_EXTENT);
        lseek(fd,50);
        str="Bye Goodnight";
        write(fd,str, strlen(str)+1);
        close(fd);
        exit();

}
```

Calling from the user level:

We created a .c file name lseek.c inside the main program. This file when run with the filename to lseek it will automatically call lseek() system call with file descriptor and offset.

- When the system call is invoked the below function in the sysfile.c file is called.This function helps in shifting the position the pointer by an offset.It return -1 when it is not able to lseek.



```
int sys_lseek(void){
  struct file *f;
  int n;
  if(argfd(0, 0, &f) < 0 || argint(1, &n) < 0 )
    return -1;
  if(f->type==FD_INODE){ // check fie type as fd inode.
    ilock(f->ip);// lock to perform operation automically.
  f->off=n; // setting the offset.
  f->ip->size=n;// setting the size.
  iunlock(f->ip);//unlocking.so other process can use the lock.
  return n;
}
  return -1;
}
```