

How can implement the system call lseek(): Moves the file offset for a given file descriptor

lseek (C System Call): lseek is a system call that is used to change the location of the read/write pointer of a file descriptor. The location can be set either in absolute or relative terms.

Function Definition

```
off_t lseek(int fildes, off_t offset, int whence);
```

Field Description

int fildes : The file descriptor of the pointer that is going to be moved

off_t offset : The offset of the pointer (measured in bytes).

int whence : The method in which the offset is to be interpreted (rela, absolute, etc.). Legal value r this variable are provided at the end.

Return value : Returns the offset of the pointer (in bytes) from the beginning of the file. If the return value is -1, then there was an error moving the pointer.

For example, say our Input file is as follows:

A screenshot of a gedit text editor window. The title bar reads 'start.txt (~/) - gedit'. The window has a menu bar with 'Open' and 'Save' options. Below the menu bar, there are two tabs: 'start.txt' and 'end.txt'. The 'start.txt' tab is active, and its content is displayed in the editor area. The text in the editor is as follows:

```
We encounter various problems like Maximum length palindrome in a string,  
number of palindromic substrings and many more interesting problems on  
palindromic substrings . Mostly of these palindromic substring problems  
have some DP O(n2) solution (n is length of the given string) or then we  
have a complex algorithm like Manacher's algorithm which solves the  
Palindromic problems in linear time. In this article, we will study an  
interesting Data Structure, which will solve all the above similar  
problems in much more simpler way. This data structure is invented by  
Mikhail Rubinchik.
```

```

#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <fcntl.h>

void func(char arr[], int n)
{
    int f_write = open("start.txt", O_RDONLY);

    int f_read = open("end.txt", O_WRONLY);

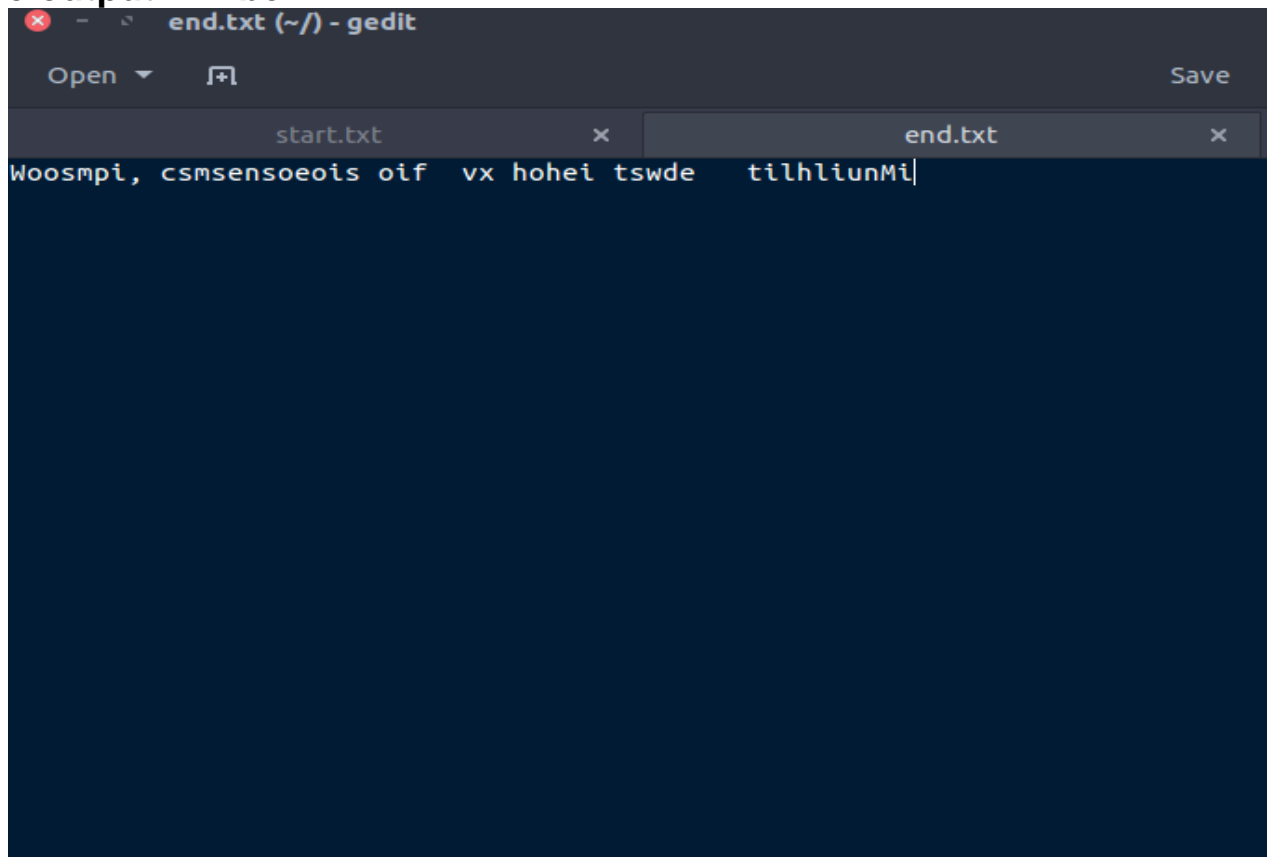
    int count = 0;
    while (read(f_write, arr, 1))
    {
        // to write the 1st byte of the input file in
        // the output file
        if (count < n)
        {
            // SEEK_CUR specifies that
            // the offset provided is relative to the
            // current file position
            lseek (f_write, n, SEEK_CUR);
            write (f_read, arr, 1);
            count = n;
        }

        // After the nth byte (now taking the alternate
        // nth byte)
        else
        {
            count = (2*n);
            lseek(f_write, count, SEEK_CUR);
            write(f_read, arr, 1);
        }
    }
    close(f_write);
    close(f_read);
}

int main()
{
    char arr[100];
    int n;
    n = 5;
    // Calling for the function
    func(arr, n);
    return 0;
}

```

The output will be



```
end.txt (~/) - gedit
Open  Save
start.txt  end.txt
Woosmpi, csmsensoeois oif vx hohei tswde tilhliunMi
```