

Assignment 5 – Buffered I/O

Description:

This assignment is to simulate how buffers work when a user is requesting data from a file using functions like open, read or close.

Approach:

An overall easier assignment compared to other assignments. Mainly test us on using functions written by someone else.

Although this assignment was one of the easier ones. I still spent a considerable amount of time doing this. The `b_open` function is straightforward to implement, just need to call `b_getFCB`, to get a file descriptor; and call the `GetFileInfo` to get the “low-level file information”, which are the `fileName`, `fileSize` and the location of the file.

Once a user requests to open a file, then `b_open` will create a file descriptor of this file, and init the struct. Return the file descriptor to the user.

Then jumping to the `b_read` function, this is the “main part” of this assignment. After the user requests to get a certain byte of data from the file, we need to read the file and fill our buffer first, then pass in the users’ required amount of data to the user’s buffer. The trick in this part is that the system will only read a block(512 byte in this case) of data, so we have to read a block of data to our own buffer, then pass it to the user accordingly.

I have 3 steps in my `b_read`:

1. Check if there are any remaining bytes in our own buffer, if the buffer is empty then do nothing, if there are bytes in the buffer:
 - a. Check if our buffer has enough bytes to give to the user, if there is, copy to the user, and then return
 - b. If our buffer is not enough, then give the user everything in our buffer.
2. Now that we have emptied our buffer, we should check if the remaining count is larger than a block. Then directly `LBRead` a block of data to the user's buffer. This step ends when the remaining count is smaller than a block.
3. Our buffer should be empty now, so we `LBRead` a block to our buffer, and then copy the remaining count to the user.

The 3 steps are the overall structure of the read function. But there are a lot of small things to keep track of, which was confusing to me, and I spent a lot of time getting them right:

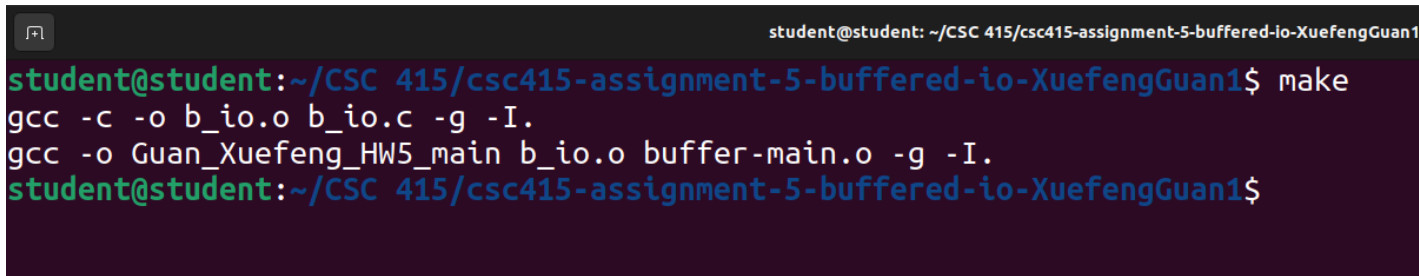
- I gotta make sure that when EOF is encountered, the program will remember that, and return 0 at the next iteration directly, without going deep into the program.
- I gotta make sure that when using `memcpy`, the third parameter which is the number of bytes to copy is not greater than our buffer’s length, otherwise this would cause overflow.

- I gotta make sure to keep track of which index of the user's buffer and my buffer is on.

Issues and Resolutions:

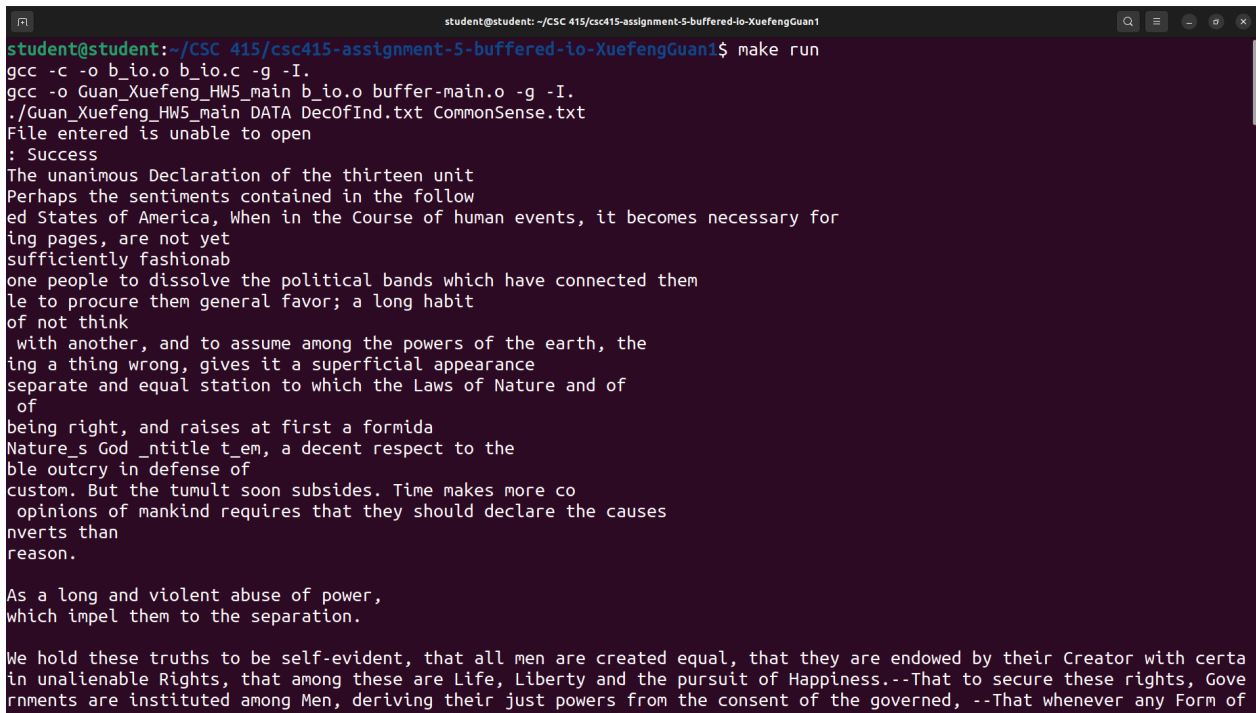
- The whole assignment I was trying to get the algorithm for the `b_read` straight, I kept getting weird answers. One thing bothered me the most was that I didn't know for example, if there are only 5 chars left in the file, and the count is 10, this means the user wants us to give them 10 more chars. When we do `LBAread(myBuffer, 1, location);` myBuffer will only have 5 chars. And then if I do `memcpy(userBuffer, myBuffer, count)`. This would cause overflow, and there were no errors indicating me that. I spent hours trying to figure out why I'm getting weird results, then I realized it was just `memcpy` doing buffer overflow.

Screenshot of compilation:

A terminal window with a dark background and light green text. The prompt is 'student@student: ~/CSC 415/csc415-assignment-5-buffered-io-XuefengGuan1'. The user enters 'make', followed by two gcc commands: 'gcc -c -o b_io.o b_io.c -g -I.' and 'gcc -o Guan_Xuefeng_HW5_main b_io.o buffer-main.o -g -I.'. The prompt returns after each command.

```
student@student:~/CSC 415/csc415-assignment-5-buffered-io-XuefengGuan1$ make
gcc -c -o b_io.o b_io.c -g -I.
gcc -o Guan_Xuefeng_HW5_main b_io.o buffer-main.o -g -I.
student@student:~/CSC 415/csc415-assignment-5-buffered-io-XuefengGuan1$
```

Screen shot(s) of the execution of the program:

A terminal window showing the execution of the program. The prompt is 'student@student: ~/CSC 415/csc415-assignment-5-buffered-io-XuefengGuan1'. The user enters 'make run'. The output shows the compilation of 'run.c' and the execution of 'Guan_Xuefeng_HW5_main'. The program reads from 'DATA/DecOfInd.txt' and 'CommonSense.txt'. It prints a success message and then displays a large block of text, which is a excerpt from the Declaration of Independence.

```
student@student:~/CSC 415/csc415-assignment-5-buffered-io-XuefengGuan1$ make run
gcc -c -o b_io.o b_io.c -g -I.
gcc -o Guan_Xuefeng_HW5_main b_io.o buffer-main.o -g -I.
./Guan_Xuefeng_HW5_main DATA/DecOfInd.txt CommonSense.txt
File entered is unable to open
: Success
The unanimous Declaration of the thirteen unit
Perhaps the sentiments contained in the follow
ed States of America, When in the Course of human events, it becomes necessary for
ing pages, are not yet
sufficiently fashionab
one people to dissolve the political bands which have connected them
le to procure them general favor; a long habit
of not think
with another, and to assume among the powers of the earth, the
ing a thing wrong, gives it a superficial appearance
separate and equal station to which the Laws of Nature and of
of
being right, and raises at first a formida
Nature_s God _ntitle t_em, a decent respect to the
ble outcry in defense of
custom. But the tumult soon subsides. Time makes more co
opinions of mankind requires that they should declare the causes
nverts than
reason.

As a long and violent abuse of power,
which impel them to the separation.

We hold these truths to be self-evident, that all men are created equal, that they are endowed by their Creator with certa
in unalienable Rights, that among these are Life, Liberty and the pursuit of Happiness.--That to secure these rights, Gove
rnments are instituted among Men, deriving their just powers from the consent of the governed, --That whenever any Form of
```

```
student@student: ~/CSC 415/csc415-assignment-5-buffered-io-XuefengGuan1

Nor have We been wanting in attentions to our Brittish brethren. We have warned them
from time to time of attempts by their legislature
to extend an unwarrantable jurisdiction over us. We
have reminded them of the circumstances of our em
igration and settlement here. We have appealed to their native justice and
magnanimity, and we have conjured them by the ties of our common kindred to disavow the
se usurpations, which, would inevitably inter
rupt our connections and correspondence. They too have been deaf
to the voice of justice and of consanguinity. We must, therefore, acquiesce in
the necessity, which denounces our Separation, and hold
them, as we hold the rest of mankind, Enemies in War, in Peace Frie
nds.

We, therefore, the Representatives of the united Sta
tes of America, in General Congress, Assembled, appealin
g to the Supreme Judge of the world for the rectitude of our
intentions, do, in the Name, and by Authority of th
e good People of these Colonies, solemnly publish and declare,
That these United Colonies are, and of Right ought to be Free and Independent
States; that they are Absolved from all
Allegiance to the British Crown, and that all polit
ical connection between them and the State of Great Britain, is and ought to be
totally dissolved; and that as Free and Independent
States, they have full Power to levy War, conclude Peace, contract All
iances, establish Commerce, and to do all other Acts and Things
which Independent States may of right do. And for the
support of this Declaration, with a firm reliance on the protection of divine Providence,
we mutually pledge to each other our Lives, our Fortunes and our sacred Honor.
We have read 8120 characters from file DecOfInd.txt
We have read 1877 characters from file CommonSense.txt
student@student:~/CSC 415/csc415-assignment-5-buffered-io-XuefengGuan1$
```

The output of this program is 8120 for DecOfInd.txt and 1877 for CommonSense.txt