

Assignment 5 – Buffered I/O

Assignment description:

This assignment is about Buffered I/O, meaning our role in this assignment is to act or be the system call functions such as `open()`, `read()`, and `close()`. Thus our job in this assignment is to handle input and output buffering by opening a given file, reading certain bytes from a file, as well as making the routine that would deallocate dynamically allocated resources that we utilized in the program.

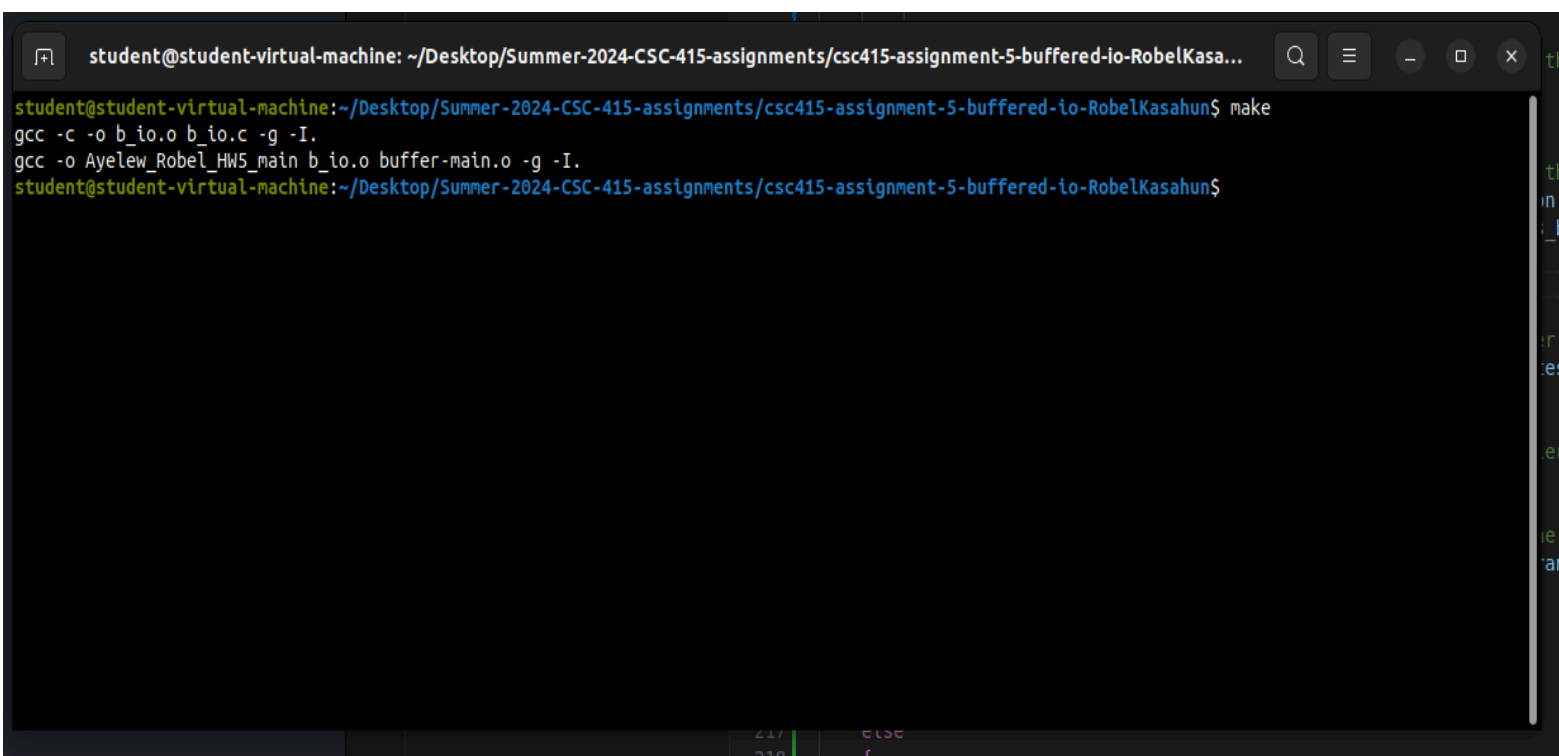
Assignment approach:

The assignment we were given initially was implemented therefore my duty is to add my logic to the routines or functions such as `b_open()`, `b_read()`, and `b_close()` so that these functions work as the system call functions that Linux provided. My approach to this assignment would be to first add variables such as `current_bytes_buffer_holding` and `buffer` to `b_fcb` structure because according to the assignment instruction each volume control block has to have its own buffer. After adding variables to `b_fcb`, I would head to the `b_open()` function to initialize the file descriptor by calling the `getFCB()`. What that would do for me is that, it will give me or return a free file control block or index of the file control block from the array. Next, I would call the `GetFileInfo()` routine by passing the file name to give me a file structure that would give me the name of the file, size in bytes, and the starting block of the file. Most importantly, the file control block that we receive from the `getFCB()` must be initialized, thus what I would do is that, I would get the free file control block from the `fcbArray` and initialize its variables. At this point, the `b_open()` will successfully initialize the free file control block and its buffer will only store 512 bytes. Then we return the index of the free file control block as a file descriptor and this file descriptor will be utilized by the `b_read()` to read a given number of bytes from a file opened by the `b_open()`. Next would be to implement the `b_read()` function which reads some bytes of data from a given file descriptor to a caller's buffer. Thus, my initial approach in this function would be to return `count` when the `count` is less than or equal to zero because at this stage there are no bytes to read since the `count` is either zero or negative number, but if `count` is greater than zero, I would read 1 block at a time from a certain position from the file into my buffer and transfer appropriate bytes into the caller's buffer. While doing this, I would keep track of the number of bytes transferred from my buffer to the caller's buffer because that's what I would do after finishing transferring.

Issues and Resolutions:

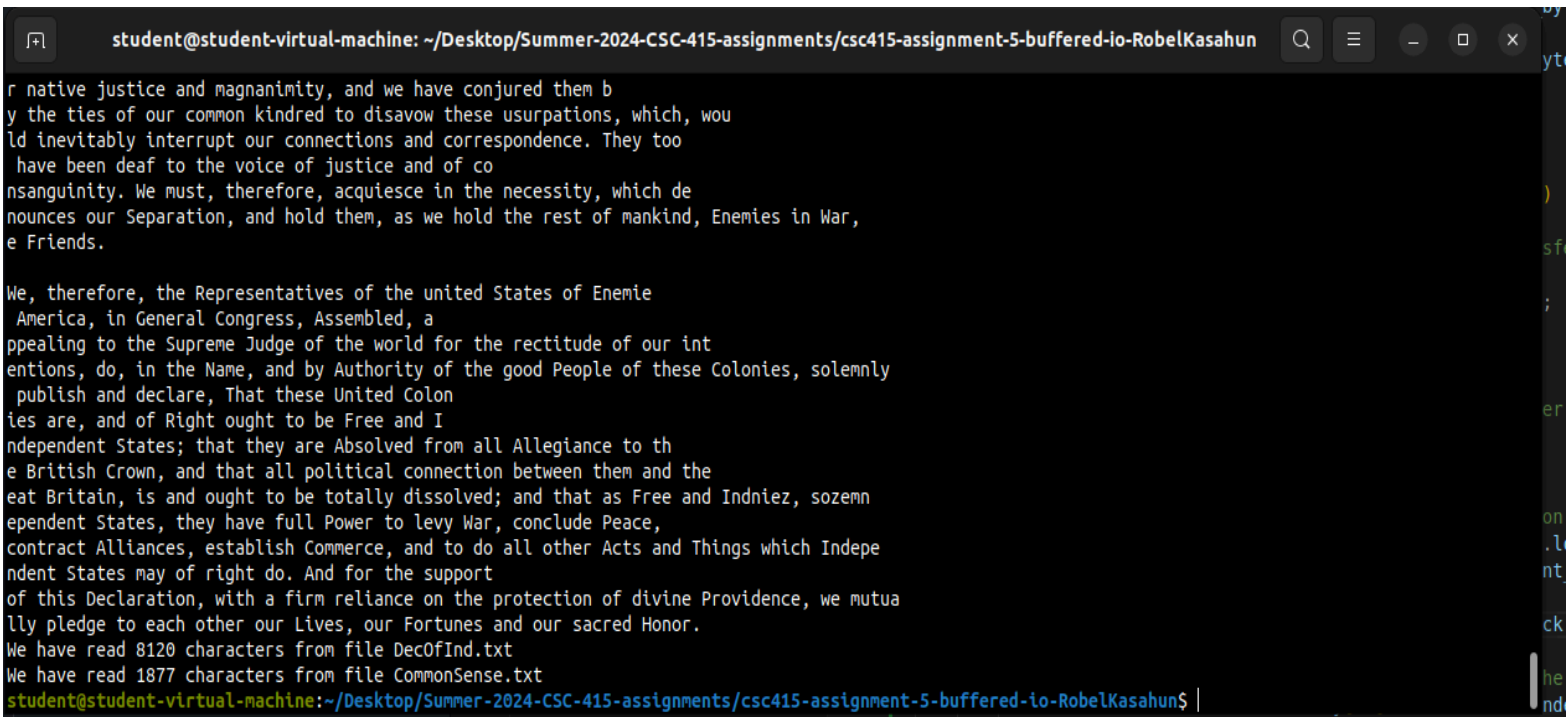
I had a lot of issues with this assignment, but the issue that took me longer than expected was getting the reading or transferring certain data work from the LBAread to my buffer and using memcpy inside the b_read() function. Fortunately, after trying more even though the due date was more than a day, I was able to get the solution. The problem with LBAread was that I was passing zero for some reason for the LBA block position, but after I changed the zero to (file location + bytes the buffer of the current file descriptor holding) divided by B_CHINK_SIZE, and was able to get the right result. Last but not least, I have a problem that I still cannot fix. The problem is that GetFileInfo() is returning a NULL value even though I passed the file name. Therefore, I am submitting the assignment without fixing the bug.

Screenshot of compilation:



```
student@student-virtual-machine: ~/Desktop/Summer-2024-CSC-415-assignments/csc415-assignment-5-buffered-io-RobelKasa...
student@student-virtual-machine:~/Desktop/Summer-2024-CSC-415-assignments/csc415-assignment-5-buffered-io-RobelKasahun$ make
gcc -c -o b_io.o b_io.c -g -I.
gcc -o Ayelew_Robel_HWS_main b_io.o buffer-main.o -g -I.
student@student-virtual-machine:~/Desktop/Summer-2024-CSC-415-assignments/csc415-assignment-5-buffered-io-RobelKasahun$
```

Screenshot(s) of the execution of the program:

A terminal window titled 'student@student-virtual-machine: ~/Desktop/Summer-2024-CSC-415-assignments/csc415-assignment-5-buffered-io-RobelKasahun'. The terminal displays the output of a program that reads two text files. The first file, 'DecOfInd.txt', contains the first part of the Declaration of Independence, and the second file, 'CommonSense.txt', contains the second part. The program outputs the text from both files and reports the number of characters read from each. The terminal window has a dark background and standard window controls (minimize, maximize, close) in the top right corner.

```
student@student-virtual-machine: ~/Desktop/Summer-2024-CSC-415-assignments/csc415-assignment-5-buffered-io-RobelKasahun$  
r native justice and magnanimity, and we have conjured them b  
y the ties of our common kindred to disavow these usurpations, which, wou  
ld inevitably interrupt our connections and correspondence. They too  
have been deaf to the voice of justice and of co  
nsanguinity. We must, therefore, acquiesce in the necessity, which de  
nounces our Separation, and hold them, as we hold the rest of mankind, Enemies in War,  
e Friends.  
  
We, therefore, the Representatives of the united States of Enemie  
America, in General Congress, Assembled, a  
ppealing to the Supreme Judge of the world for the rectitude of our int  
entions, do, in the Name, and by Authority of the good People of these Colonies, solemnly  
publish and declare, That these United Colon  
ies are, and of Right ought to be Free and I  
ndependent States; that they are Absolved from all Allegiance to th  
e British Crown, and that all political connection between them and the  
eat Britain, is and ought to be totally dissolved; and that as Free and Indniez, sozemn  
ependent States, they have full Power to levy War, conclude Peace,  
contract Alliances, establish Commerce, and to do all other Acts and Things which Indepe  
ndent States may of right do. And for the support  
of this Declaration, with a firm reliance on the protection of divine Providence, we mutua  
lly 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-virtual-machine:~/Desktop/Summer-2024-CSC-415-assignments/csc415-assignment-5-buffered-io-RobelKasahun$ |
```