Haley Park ID: 923812276
Github: jung-hyeon CSC415 Operating Systems

Assignment 5 - Buffered I/O

Description:

This assignment involves the implementation of buffered I/O operations in C, where the buffering is managed manually. Three functions need to be written: b_open, b_read, and b_close, as specified in the provided b_io.h file. b_open is responsible for returning a file descriptor and allocating a buffer. b_read is tasked with reading data in chunks of size B_CHUNK_SIZE and copying appropriate bytes to the caller's buffer. b_close should handle freeing any allocated resources. Low-level APIs like LBAread and GetFileInfo will be utilized for file operations. The b_read function must address cases where the read request exceeds the buffer size, and it should keep track of the file size, returning 0 upon reaching the end of the file.

Approach:

Assumption

- File is contiguous in this assignment
- In b_open function, Ignore flags in this assignment
- 1. b open function
 - a. Get file information
 - i. By using GetFileInfo function, get file information
 - ii. If file is not found, return -1 (ERROR)
 - b. Get file control block
 - i. By using b getFCB function, get file descriptor
 - ii. If there is no available FCB, return -1 (ERROR)
 - c. Allocate memory for buffer
 - i. By using malloc, allocate memory size of B CHUNK SIZE
 - ii. If memory allocation is failed, return -1 (ERROR)
 - d. Populate the FCB with necessary information
 - Initialize variables like fi, buffer, buf_index, buf_remain, block_count, and file_index
 - e. Return the file descriptor
- 2. b_read function
 - a. If the user requested bytes is bigger than file size, change the user requested bytes to remained file bytes
 - b. If bytes that remain in own buffer is bigger than user requested bytes,
 - Directly buffer from own buffer to caller's buffer (not need to LBAread) because own buffer already has bytes
 - ii. ⇒ Copy memory as many bytes as user wants directly from own buffer
 - c. Else, Need to use LBAread to get data (need to get new data)
 - i. Copy memory from own buffer if it is not empty (Before get new data)
 - ii. ⇒ Copy memory as many bytes as bytes remained in own buffer

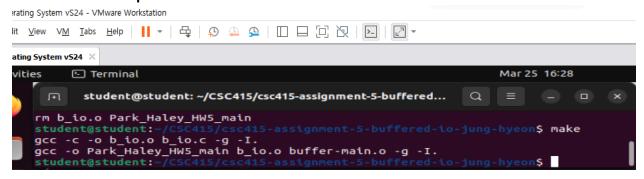
- iii. If bytes wanted by user is bigger than B_CHUNK_SIZE, buffer directly from file to user buffer
 - 1. Calculate needed lba_count
 - 2. LBAread from file to buffer as many blocks as needed
- iv. Read B_CHUNK_SIZE a byte chunks from LBAread into the buffer
- v. Buffer the remaining bytes that user wants from own buffer to user buffer
- d. Track file index
- e. Return read bytes count
- 3. b close function
 - a. Free the buffer associated with the file
 - b. Reset the file control block

Issues and Resolutions:

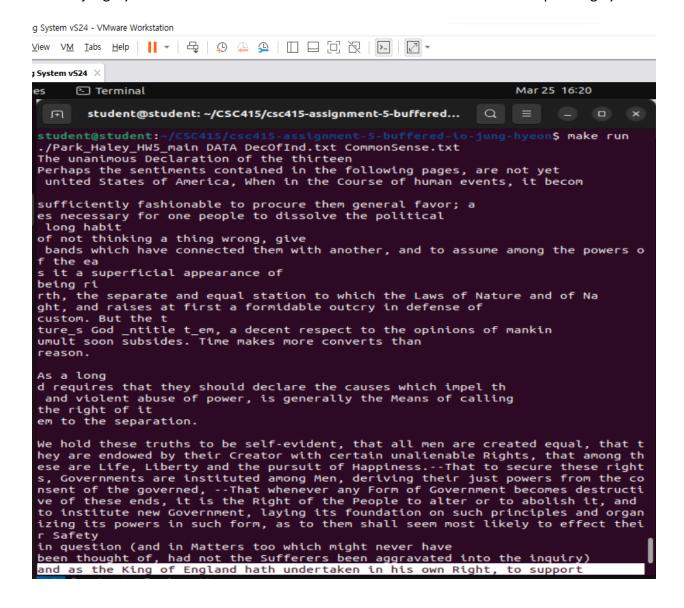
- 1. Using LBAread
 - a. First, I assign 0 for LBAread's last parameter. But It's not right. I always got the same strings. So I changed it by using the file's location.
 - b. When the bytes wanted by the user are bigger than B_CHUNK_SIZE, I used a
 while loop for the first time. But I can get multiple B_CHUNK_SIZE at one time. So
 I changed LBAread's second parameter to Iba count
 - i. EX) int lba count = (count read count) / B CHUNK SIZE;

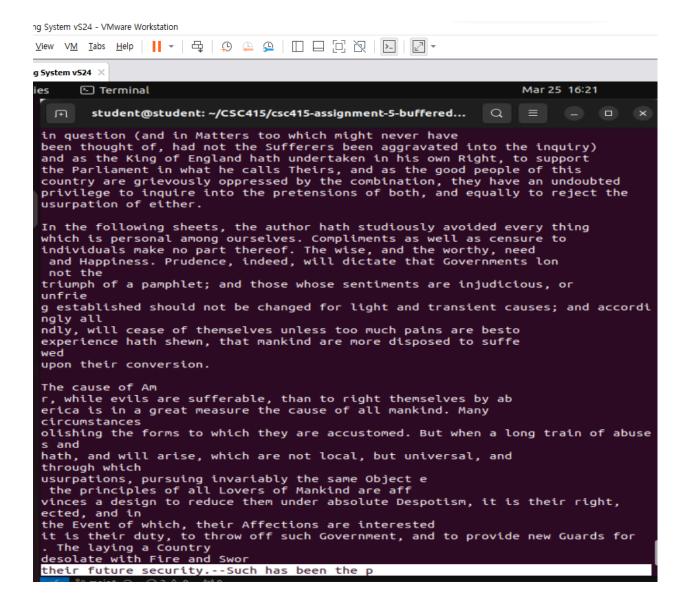
Analysis: N/A

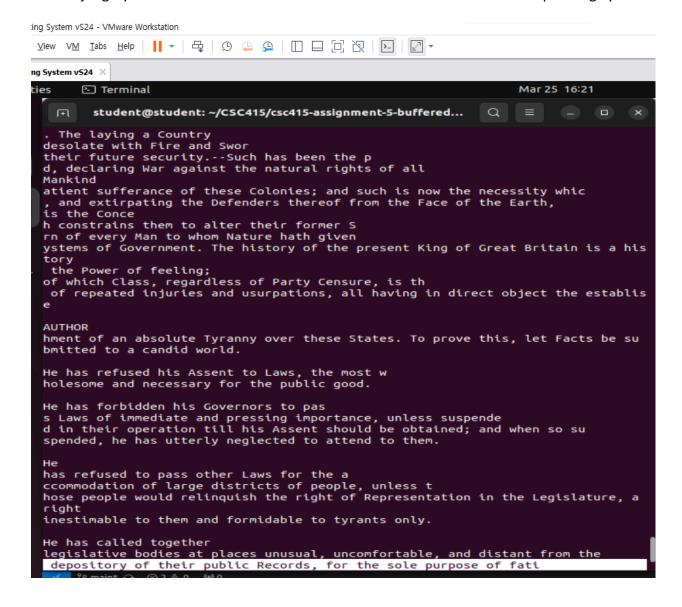
Screen shot of compilation:

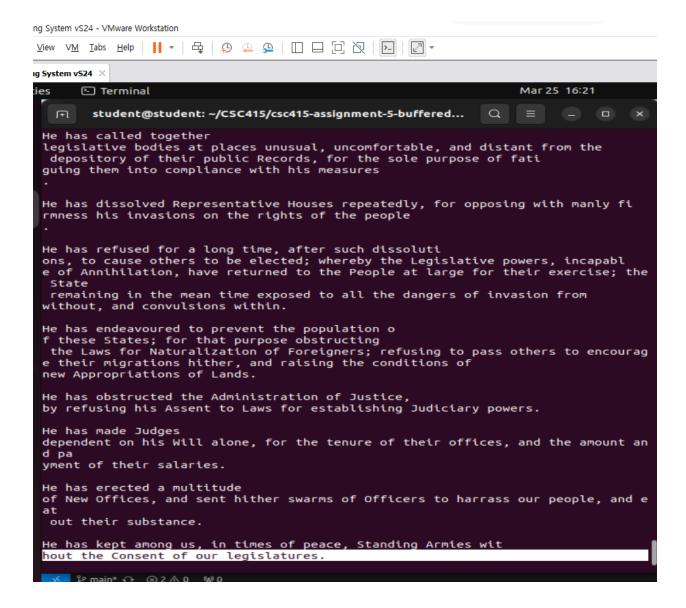


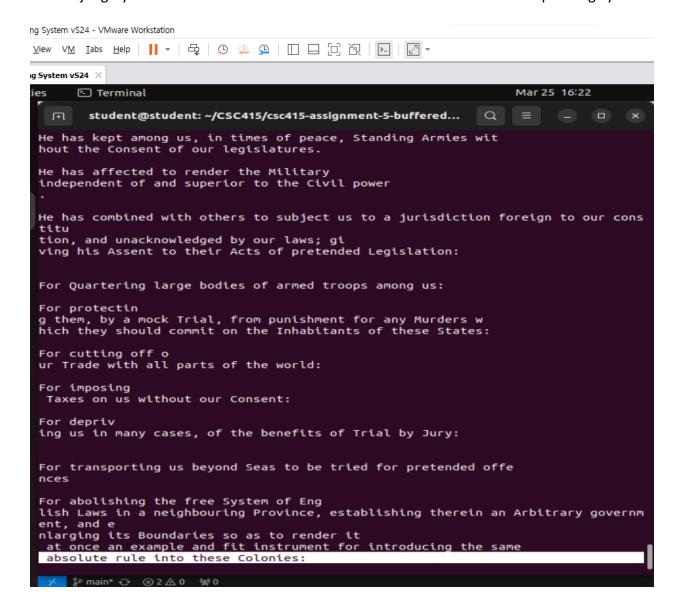
Screen shot(s) of the execution of the program:

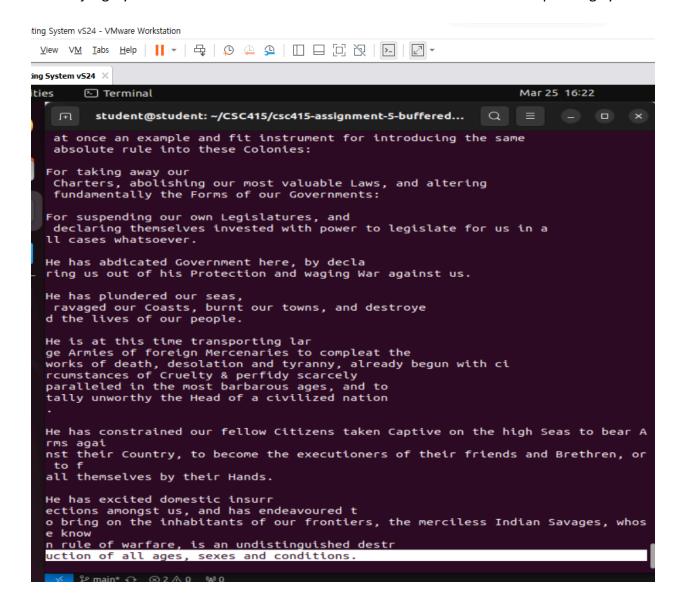


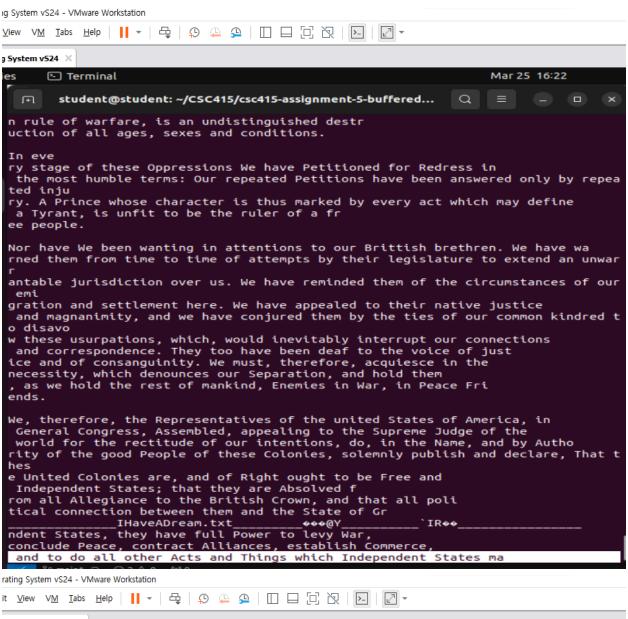












ating System vS24 X Terminal Mar 25 16:23 vities student@student: ~/CSC415/csc415-assignment-5-buffered... Q IHaveADream.txt `IR** ndent States, they have full Power to levy War, conclude Peace, contract Alliances, establish Commerce, and to do all other Acts and Things which Independent States ma y of right do. And for the support of this Declaratio n, with a firm reliance on the protection of divine Providence, we mutually ple dg e to each other our Lives, our Fortunes and our sacre d Honor. We have read 8120 characters from file DecOfInd.txt We have read 1877 characters from file CommonSense.txt student@student:~/CSC415/csc415-assignment-5-buffered-io-jung-hyeon\$