Daryl Stronge ID: 917844673
Github: DSNoNintendo CSC415.01 Operating Systems

Assignment 5 – Buffered I/O read

Description:
This assignment is to read data from read data from memory blocks using a file control block, and copy the data to a user buffer.

Approach:
My first step in this assignment was figuring out what member variables the file control block would require. I decided on were:

- *buffer* to store the data from the memory blocks (malloced to *B_CHUNK_SIZE*)
- *curr_block* to store what memory block to read from. I initialized this variable with the value of *location* from the *fileInfo* object
- *curr_byte* to store the current index of the buffer for copies to the caller buffer
- *remaining_bytes* to store the number of bytes left to be copied
- *bytes_copied* to track the total number of bytes copied to the caller buffer

After this, I wanted to create an efficient way to copy bytes to the caller buffer while keeping track of the necessary member variables listed above. To do this, I created a function *copy_to_caller_buffer* to copy bytes from the file control block buffer to the caller buffer while incrementing the necessary file control block variables to keep track of the moving parts of the program. Whenever bytes were copied to the caller buffer buffer, *curr_byte*, *remaining_bytes*, and *bytes_copied* all needed to be incremented or decremented by the number bytes copied from the file control block to the caller buffer.

I also needed an efficient way to reset these variables when the next memory block needed to be read. I created a function *reset_fcb* to do this. Whenever the time comes for the file control block to read from the next block, the *remaining_bytes*, *curr_byte*, and *buffer* are reset.

The last function I created is *all_bytes_read*. This function simply cross-references the fileSize and bytes_copied member variables to check if all bytes of a file have been read.

After implementing these functions, my last step was to implement the logic for my code to run correctly.

Issues and Resolutions:

I didn't face many issues working on this assignment, but I did use this assignment as an opportunity to write more efficient code and clean up my logic.

For example, this function:

```
int copy_to_caller_buffer(char *dest, char *source, int count, b_fcb *fcb)
{
	memcpy(dest, source, count);
	fcb->curr_byte += count;
	fcb->bytes_copied += count;
	fcb->remaining_bytes -= count;
	return count;
}
```

Was written to mimic the *memcpy* function. It accepts the same exact arguments as *memcpy*, plus the file control block argument. Instead of incrementing the file control blocks member variables every time *memcpy* was called, I wrote a function to do it in order to keep my code cleaner.

For this assignment I also used ternary operators to avoid using too many if-else statements. In this code snippet:

```
bytes_copied = copy_to_caller_buffer(buffer + total_bytes_copied,
                                     fcb->buffer + fcb->curr_byte,
                                     (fcb->fi->fileSize - fcb->bytes_copied < count)
                                         ? fcb->fi->fileSize - fcb->bytes_copied :
                                         count,
                                     fcb);
```

I could have written another if-else statement here in order to figure out whether to copy the number of bytes requested by the caller or the number of bytes remaining in the file, but the third argument of the *copy_to_caller_buffer* function was derived using a ternary operator, making the code faster to read.

Screen shot of compilation:

```
student@student:~/csc415-assignment-5-buffered-io-DSnoNintendo$ make
gcc -c -o b_io.o b_io.c -g -I.
gcc -o Stronge_Daryl_HW5_main b_io.o buffer-mainM1.o -g -I.
student@student:~/csc415-assignment-5-buffered-io-DSnoNintendo$
```

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

```
nd hold them, as we hold the rest of mankind, Enemies in War, in Peac
e Friends.

We, therefore, the Representatives of the uni
ted States of America, in General Congress, Assembled, appealing to the S
upreme Judge of the world for the rectitude of our intentions, do, in the Name,
and by Au
thority of the 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 political connection between them and the Sta
te 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
Pea
ce, contract Alliances, establish Commerce, and to do all
other Acts and Things which Independent St
ates may of right do. And for the support of this Declarati
on, with a firm reliance on the protection of divine Providence, we mutually ple
dg
e 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:~/csc415-assignment-5-buffered-io-DSnoNintendo$
```