Junyoung Kim ID: 920303420
Github: coder-j-0090 CSC415 Operating Systems

Assignment 2 – Struct and Buffer

Description:
This assignment requires students to program that use structures, pointers, character strings, enumerated types, bitmap fields,and buffering data into blocks

Approach:
First of all, I read the README file, and checked if the .h file matches the instruction. Then, I used the if statement to check if argc has 3 components(first name, last name, and message). If not, the program stopped immediately to save resources.

Then, I called struct, which was in .h file, then allocated memory with malloc, named it student, then set the first and second argument as firstName and lastName from struct. Following instruction, I typed in my student ID and grade level, which was named level in .h file, and I entered one of the values in enum. For student->languages, considering each KNOWLEDGE_OF_NameOfProgram has assigned number, I used | as bitwiser, and it worked. I also made sure that the message is limited to 100 characters.

Instruction said return 0 if writePersonalInfo function from struct work, so I used if statement to check if it works properly. After this process, I used free to make sure every malloc has its own corresponding free.

After setting up malloc for buffer and initializing the buffer size and position, I ran a while statement that will work until it goes over everything in Buffer ; if everything is gone over, getNext() will return NULL. It retrieves data and checks the end of data.

I declared remainder and stringPos to check how much input(currentStr) needs to be copied and the position of the string to find out where to start with. Then, I used another while statement. Unless the remaining string is less than BufferSize, it copied the remaining portion of currentStr to stringBuffer and updated the position and the size of the buffer. Then, it broke out the loop as the entire string has been copied.

If the buffer isn't big enough, then copy as much as possible. 'commitBlock(stringBuffer)' processes the buffer, then updates the remaining space and position of the string. Then, the program resets the 'sizeOfBuffer' and 'stringPos' to 0 so the next chunk of data could work.

After that, if statement(if(sizeOfBuffer < BLOCK_SIZE)) check if there's any leftover data after the loop, and if so, use 'commitBlock' to process the remaining data. Then, free allocated memory for stringBuffer and call function 'checkIt()' to verify.

Issues and Resolutions:
I was trying if(argc!=3) at the very beginning of the program to see if it could work, since argc should be firstName, lastName, and message, so I thought using argc!=3 would be more efficient, but it didn't work, so I tried something similar that means same, and argc < 4 worked.

'student->languages' also caused some issues. It was integer in .h file, so I was confused, but since all the values for KNOWLEDGE_OF_NameOfLanguage was 1,2,4, etc so I tried ||(or), but it didn't work. Considering these numbers are multiplying 2 starting from 1, so I tried bitwiser, and it worked.

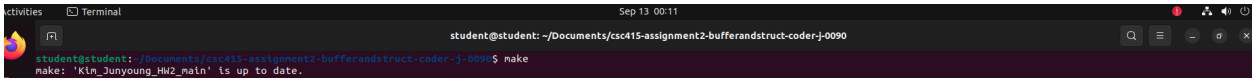
Another issue i had was order of parameters in memcpy, which was ordered in (defining a pointer(void), defining source of data to be copied(const void *), and number of bytes(size_t) to be copied). I found this information from tutorialspoint(https://www.tutorialspoint.com/c_standard_library/c_function_memcpy.htm), and it also influenced my datatype choice for stringBuffer, sizeOfBuffer, and bufferPos.

Analysis:

It shows that Languages that I know are C, Java, JavaScript, Python, C++, SQL, and HTML. 3103 is decimal, and convert it to binary, it's 0000 1100 0001 1111. Starting from the right(Least significant bit), so bits 0, 1, 2, 3, 10, and 11 are included, which are C, Java, Javascript, Python, SQL, and HTML.

It shows my other information correctly such as Student Level, name, and student ID.

Screenshot of compilation:



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