Hajime Miyazaki ID: 922402751
Github: hMiyazaki95 CSC415 Operating Systems

Assignment 2 - Buffer And Struct

Description:
purpose of this assignment is to show how to use structures, pointers, character strings, enumerated types, bitmap fields, and buffering data into blocks.

Approach / what I Did:

After I cloned the repository of Assignment 2, First, I included a header file and three libraries. Then I started with the main function that takes two parameters. In the main function, I first created a pointer “hm” to a structure called “personalInfo” and used the malloc() to allocate memory based on the structure size. For the variable naming convention “hm” is the initial in my first name and last name since the function is about personal information. After the memory allocation, I checked if the memory was successfully allocated into the structure. Otherwise, it prints an error message and exits the program.

Next, I set the attribute values of the personalInfo structure using command-line arguments and constants. It sets the first and last names using argv[1] and argv[2], respectively. It sets studentID to a constant value and level to a constant value of the enum type "level_t". It sets the "languages" variable to a bitwise OR operation of the constants KNOWLEDGE_OF_C, KNOWLEDGE_OF_CPLUSPLUS, and KNOWLEDGE_OF_JAVASCRIPT. It copies the message from the command-line argument to the "message" field of the personalInfo structure using the "strcpy" function. Then I called the "writePersonalInfo" function to write the personal information to a file. If the function returns 0, the program prints "Success" to the console. Otherwise, it prints "Failure".

Next, I made the program that enters a loop that reads strings using the "getNext" function until there are no more strings. For each string, it computes its length and copies it into the "buffer" array using a for loop. If the buffer is full, it commits the block using the "commitBlock" function, sets the buffer to zero using the "memset" function, and sets the index to zero. If the buffer is not full, it continues to read and copy strings until it is full. After the loop, the program checks if the buffer is not empty and commits the remaining bytes to the block using the "commitBlock" function.

Finally, I called the "checkIt" function to obtain the hexdump of the personalInfo structure and store it in the "r" variable. It then frees the memory allocated for the personalInfo structure using the "free" function and returns the value of the "r".

Issues and Resolutions:
I didn’t encounter particular issues but one issue that I think could happen is a buffer overflow since I used a fixed-size buffer to read strings from the getNext() function. This could be resolved by allocating memory dynamically for the buffer as needed.

Hajime Miyazaki
Github: hMiyazaki95

ID: 922402751
CSC415 Operating Systems

Screenshot of compilation: