Robel Ayelew ID:922419937
Github: RobelKasahun CSC415 Operating Systems

Assignment 2 – Buffering and Structures

Assignment Description:

The main aim of this assignment is to learn about structures, buffering as well as dynamic memory allocation, therefore this assignment has two parts. The first part is to populate the given personalInfo structure with our personal information, meaning the personalInfo structure has firstName, lastName, studentID, gradeLevel, languages, and message properties, hence the plan is to populate or to assign my information to the above properties.
The second part of this assignment is to fill a limited size that is a 256-byte buffer using values that come from the getNext function. At this point, we do not know how big in bytes the values coming from the getNext function, hence the actual problem of the second part of this assignment is to write the big data that we do not know in size in the buffer which is 256 bytes big without overflowing the buffer.

Approach:

My approach to this assignment is that I would first finish the first part of this assignment by first allocating dynamic memory on the heap because we cannot populate NULL personalInfo, therefore the first step is going to be the allocation of memory on the heap for the personalInfo. After successfully allocating memory for personalInfo, it is time to populate the properties with information of mine.

My approach to the second part of this assignment which is buffering is going to be first to allocate memory of size 256 bytes, therefore this is going to be our buffer for storing data that comes from the getNext function. Furthermore, I would check if the memory allocation has been successful for the buffer. If so, I would use some kind of looping to capture each string that comes from the getNext() function. Inside the while loop, I would start filling the buffer while the current string size is less than the current buffer size, if not we know that either the buffer is full or the current string size is greater than the remaining buffer size. What I would do is I would get the remaining bytes by subtracting the remaining buffer size from the current string size since the current string size is greater than the remaining buffer size and fill the buffer. At this point, the buffer is full. If there are data that come from the getNext function, I would advance the position pointer to the very beginning of the buffer and overwrite the buffer. Most importantly, I would commit the block or flush if I knew that the buffer was full. Finally, I would do the above instructions as long as there are data that come from the getNext function.

Issues and Resolutions:

I had a lot of issues, but the main issue I had was with the second part of this assignment which was buffering. For this part, I had *No such file or directory* fault that persisted for longer hours because I did not know where it came from, but the problem was that I made a mistake when allocating dynamic memory for personalInfo. What I did is that I allocated personalInfo with the size of BLOCK_SIZE, meaning I did personalInfo personal_info = malloc(BLOCK_SIZE) instead of personal_info = malloc(sizeof(personalInfo)), therefore this was the main issue I had and made it work by changing from malloc(BLOCK_SIZE) to malloc(sizeof(personalInfo)).

Analysis:

```
000000: (1) → E6 42 91 B1 FC 7F 00 00  (2) → EC 42 91 B1 FC 7F 00 00.
000010: (4) → E1 02 FB 36 (5) → 14 00 00 00  (3) →  1E 0C 01 00 (6) → 46 6F 75 72
000020: 20 73 63 6F 72 65 20 61  6E 64 20 73 65 76 65 6E
000030: 20 79 65 61 72 73 20 61  67 6F 20 6F 75 72 20 66
000040: 61 74 68 65 72 73 20 62  72 6F 75 67 68 74 20 66
000050: 6F 72 74 68 20 6F 6E 20  74 68 69 73 20 63 6F 6E
000060: 74 69 6E 65 6E 74 2C 20  61 20 6E 65 77 20 6E 61
000070: 74 69 6F 6E 2C 20 63 6F  6E 63 65 69 76 65 64 20
```

E6 42 91 B1 FC 7F 00 00 represents the ***firstName*** from the personalInfo structure because when I printed out the memory address of the firstName, I found that they are the same memory address.

EC 42 91 B1 FC 7F 00 00 represents the ***lastName*** of the personalInfo structure.

1E 0C 01 00 represents the property named ***languages*** from the personalInfo structure.

E1 02 FB 36 represents the property named ***studentID*** from the personalInfo structure.

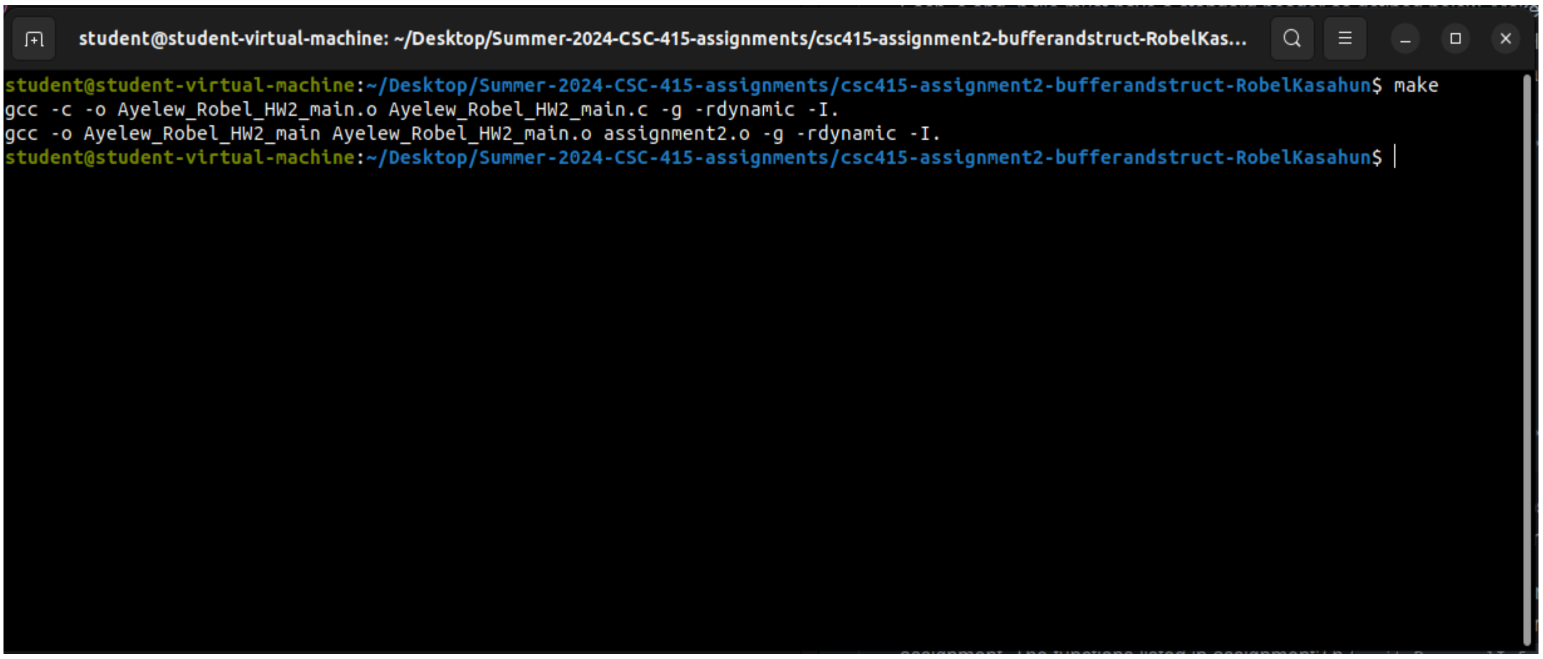
14 00 00 00 represents the property named ***level*** from the personalInfo structure.

Last but not least #6 represents represents to the values of the ***message*** property in the personalInfo structure.

Robel Ayelew
Github: RobelKasahun

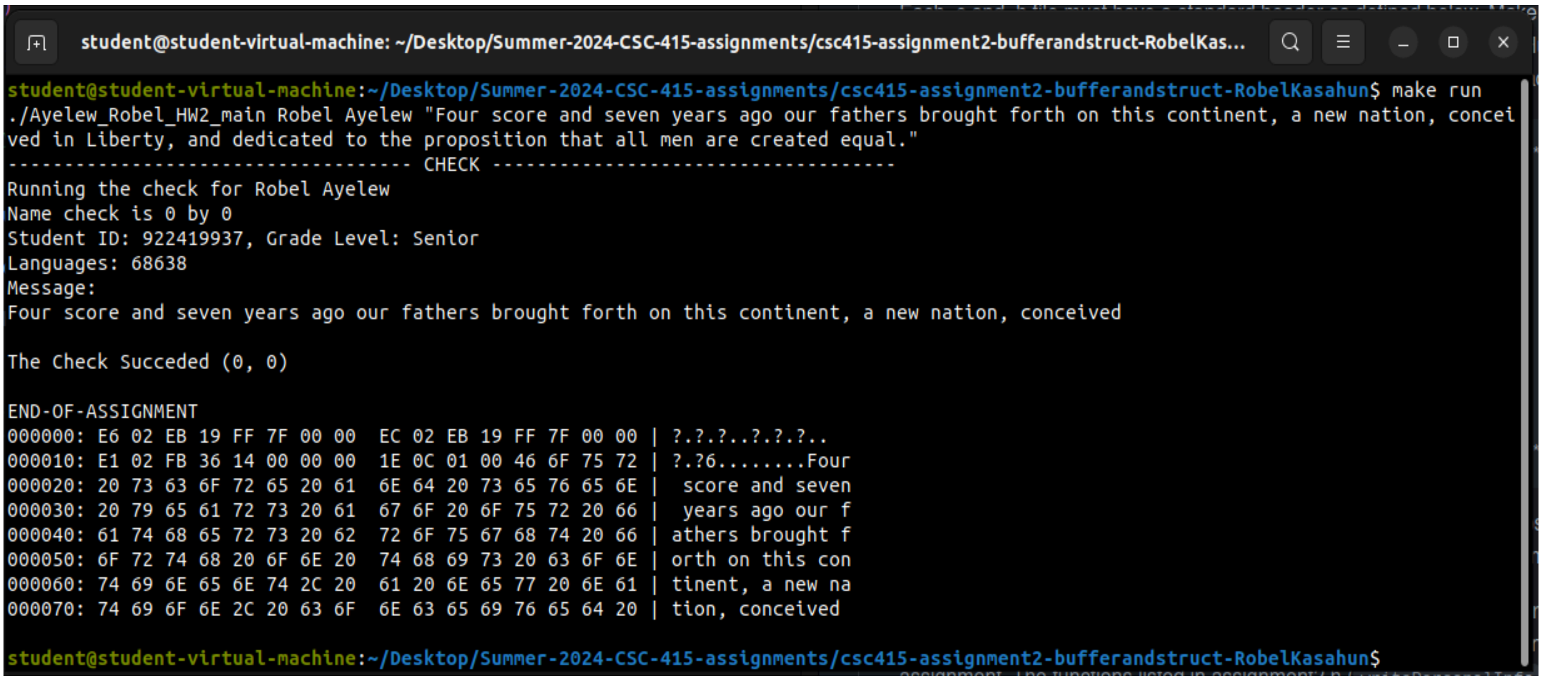
ID:922419937
CSC415 Operating Systems

Screenshot of compilation:



```
student@student-virtual-machine: ~/Desktop/Summer-2024-CSC-415-assignments/csc415-assignment2-bufferandstruct-RobelKas...
student@student-virtual-machine:~/Desktop/Summer-2024-CSC-415-assignments/csc415-assignment2-bufferandstruct-RobelKasahun$ make
gcc -c -o Ayelew_Robel_HW2_main.o Ayelew_Robel_HW2_main.c -g -rdynamic -I.
gcc -o Ayelew_Robel_HW2_main Ayelew_Robel_HW2_main.o assignment2.o -g -rdynamic -I.
student@student-virtual-machine:~/Desktop/Summer-2024-CSC-415-assignments/csc415-assignment2-bufferandstruct-RobelKasahun$
```

Screenshot(s) of the execution of the program:



```
student@student-virtual-machine: ~/Desktop/Summer-2024-CSC-415-assignments/csc415-assignment2-bufferandstruct-RobelKas...
student@student-virtual-machine:~/Desktop/Summer-2024-CSC-415-assignments/csc415-assignment2-bufferandstruct-RobelKasahun$ make run
./Ayelew_Robel_HW2_main Robel Ayelew "Four score and seven years ago our fathers brought forth on this continent, a new nation, concei
ved in Liberty, and dedicated to the proposition that all men are created equal."
------------------------------------- CHECK -----------------------------------
Running the check for Robel Ayelew
Name check is 0 by 0
Student ID: 922419937, Grade Level: Senior
Languages: 68638
Message:
Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived

The Check Succeded (0, 0)

END-OF-ASSIGNMENT
000000: E6 02 EB 19 FF 7F 00 00  EC 02 EB 19 FF 7F 00 00 | ?.?.?..?.?.?..
000010: E1 02 FB 36 14 00 00 00  1E 0C 01 00 46 6F 75 72 | ?.?6........Four
000020: 20 73 63 6F 72 65 20 61  6E 64 20 73 65 76 65 6E |  score and seven
000030: 20 79 65 61 72 73 20 61  67 6F 20 6F 75 72 20 66 |  years ago our f
000040: 61 74 68 65 72 73 20 62  72 6F 75 67 68 74 20 66 | athers brought f
000050: 6F 72 74 68 20 6F 6E 20  74 68 69 73 20 63 6F 6E | orth on this con
000060: 74 69 6E 65 6E 74 2C 20  61 20 6E 65 77 20 6E 61 | tinent, a new na
000070: 74 69 6F 6E 2C 20 63 6F  6E 63 65 69 76 65 64 20 | tion, conceived

student@student-virtual-machine:~/Desktop/Summer-2024-CSC-415-assignments/csc415-assignment2-bufferandstruct-RobelKasahun$
```