Nathan Rennacker ID: 921348958
Github: nlrennacker CSC415 Operating Systems

Assignment 2 - Struct and Buffer

Description:

This assignment touched on a lot of different topics: from populating structs, to pointers, filling buffered memory blocks from a set of larger data, finishing off with understanding how little endian addressing can be converted to hex. Firstly, the personalInfo struct is filled with my own information, and then a partial string from the command line. It's then written to a data structure which we don't have access to. Similarly, a string is made available through getNext() – from a data structure that we don't have access to, and is written to another data structure that we don't have access to via a series of buffer commits.

Approach / What I Did:

I started off this assignment by touching up on malloc and the way it interacts with sizeof in relation to struct memory sizes. Then I started populating the data within the personalInfo structure. All of that was straightforward, then I added some checks for the message portion of the personalInfo structure that would ensure, even if the 4th argument was less than the size of the message array, there wouldn't be any data written into message that was *not* supposed to be there. I then moved on to the buffer section of the assignment where I first created a nested loop that did the job, but ended up refining my code so that I could use memcpy instead. (I was debating between using strncpy and memcpy but I decided to use memcpy because I'm already calculating the length of what I need to copy over to the bufferblock, so the extra null terminator checking that strncpy has didn't seem necessary to use)

Issues and Resolutions:

Problem
I struggled to understand how little endian memory relates to hexadecimal in the analysis part of the assignment.
Solution
I really had to play around with a hexadecimal to int/text converter for a long time before the ordering of how data was stored in addresses and how to convert that to hexadecimal actually clicked.

(cont.)

Problem
I wanted to make the solution to populating and committing the buffer more efficient than my initial solution which involved a nested loop.
Solution
I wrote out what I thought would work, and then tried to implement it, it didn't function as expected at first, but I realized I wasn't updating the bytePos for where in the bufferBlock I left off, after I did that it worked like a charm!

Analysis:

```
END-OF-ASSIGNMENT
000000: 76 32 14 EA FC 7F 00 00  7D 32 14 EA FC 7F 00 00     | v2.??..}2.??..
000010: 5E AB EA 36 03 00 00 00  1F 0C 08 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
```

Values in the table are addressed in little endian so, for example, in the case of the first 8 bytes, `76 32 14 EA FC 7F 00 00` would translate to hexadecimal: `0x00007FFCEA143276`

🟧 Is the 8 byte *pointer* value which corresponds with the first name in the personalInfo struct (which is why it shows up as gibberish on the right, pointer ≠ readable string)

🟨 Is the 8 byte *pointer* value which corresponds with the last name in the personalInfo struct

🟦 Is the 4 byte value of the id in the personalInfo struct
Hex: `0x36EAAB5E` Decimal: `921348958`

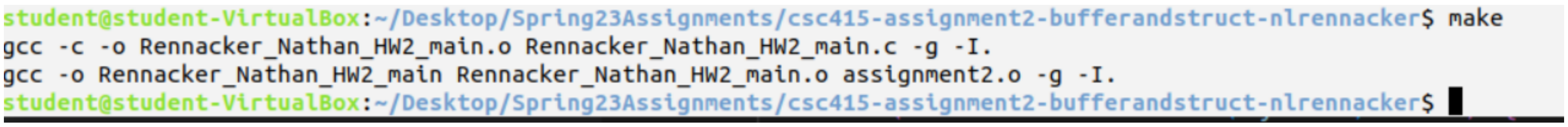
🟦 Is the 4 byte value which is the int that corresponds with the enum gradeLevel in the personalInfo struct (enums in C are stored as an int which take 4 bytes of storage)
Hex: `0x00000003` Decimal: `3 - corresponds to SENIOR`

Nathan Rennacker ID: 921348958
Github: nlrennacker CSC415 Operating Systems

🟦 Is the 4 byte value which corresponds with the languages in the personalInfo struct
Hex: `0x00080C1F` Binary: `00000000000010000000110000011111`
This means I know C, Java, Javascript, Python, C++, SQL, HTML, Arm Assembly

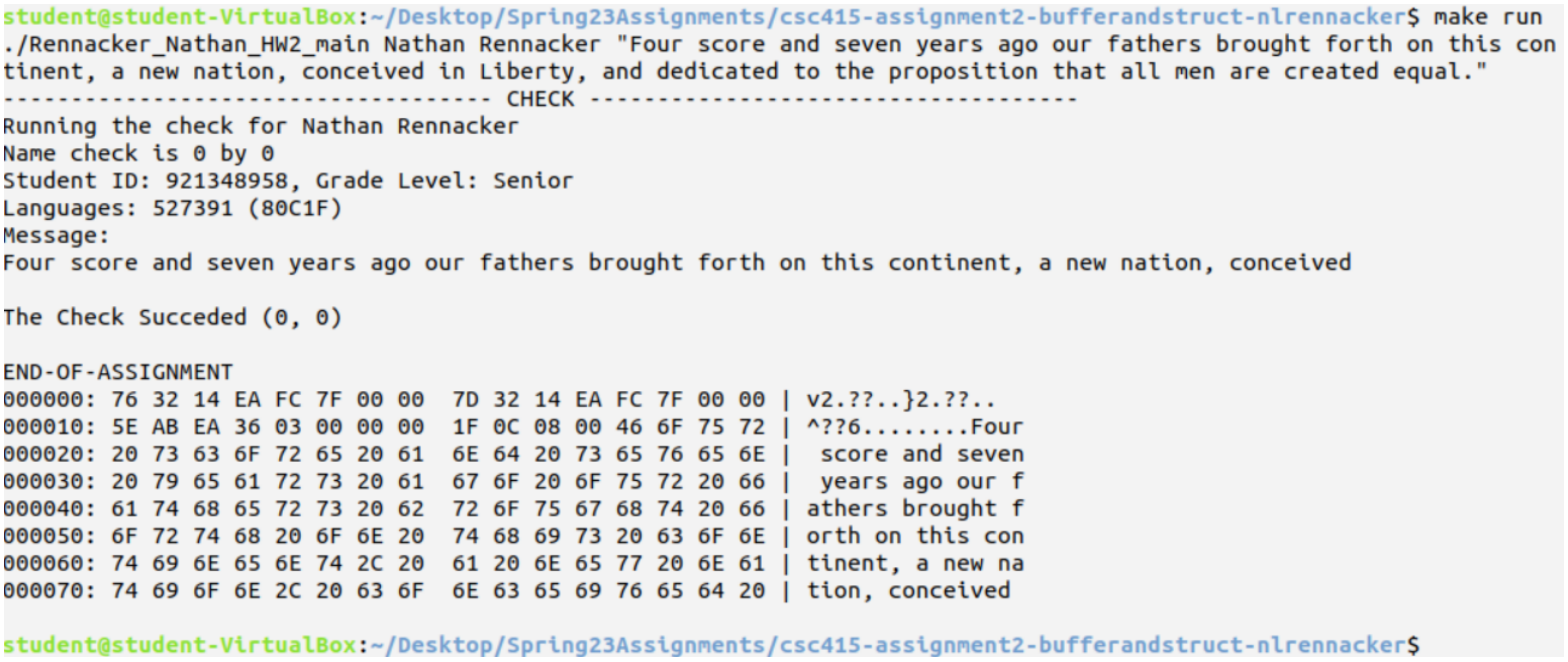
🟪 Is the 100 bytes which correspond with the message in the personalInfo struct
Hex value → `Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived`

Compilation:



```
student@student-VirtualBox:~/Desktop/Spring23Assignments/csc415-assignment2-bufferandstruct-nlrennacker$ make
gcc -c -o Rennacker_Nathan_HW2_main.o Rennacker_Nathan_HW2_main.c -g -I.
gcc -o Rennacker_Nathan_HW2_main Rennacker_Nathan_HW2_main.o assignment2.o -g -I.
student@student-VirtualBox:~/Desktop/Spring23Assignments/csc415-assignment2-bufferandstruct-nlrennacker$ 
```

Execution:



```
student@student-VirtualBox:~/Desktop/Spring23Assignments/csc415-assignment2-bufferandstruct-nlrennacker$ make run
./Rennacker_Nathan_HW2_main Nathan Rennacker "Four score and seven years ago our fathers brought forth on this con
tinent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal."
----------------------------------- CHECK -----------------------------------
Running the check for Nathan Rennacker
Name check is 0 by 0
Student ID: 921348958, Grade Level: Senior
Languages: 527391 (80C1F)
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: 76 32 14 EA FC 7F 00 00  7D 32 14 EA FC 7F 00 00 | v2.??..}2.??..
000010: 5E AB EA 36 03 00 00 00  1F 0C 08 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-VirtualBox:~/Desktop/Spring23Assignments/csc415-assignment2-bufferandstruct-nlrennacker$
```