Juan Estrada
Github: jjestrada2
ID: 923058731
CSC415 Operating Systems

Assignment 2 – Struct and Buffer

Description:
The main function, command-line arguments are used to populate a personalInfo structure with personal details, such as the first and last name, student ID, grade level, knowledge of various programming languages, and a message. These details are then written using the writePersonalInfo function. The program proceeds to allocate a buffer and processes a series of strings obtained from the getNext function. It copies these strings into the buffer, ensuring that the buffer does not exceed a specified block size. When the buffer is full or all strings have been processed, it commits the data using the commitBlock function. The final step is to call the checkIt function to validate the stored personal information. After freeing allocated memory, the function returns an exit code based on the success of the checkIt function, providing feedback on the execution of the program.

Approach / What I Did:

1. Allocate Memory: Initially, the program allocates memory for a personalInfo structure using malloc. It checks if the allocation was successful and exits with an error message if not.

2. Populate Personal Info: The first and second command-line arguments (first name and last name) are copied into the firstName and lastName fields of the personalInfo structure. The student ID is set to a predefined value, and the grade level is set to "SENIOR." The languages field is populated using bitwise OR operations with predefined language constants. The third command-line argument is copied into the message field.

3. Write Personal Info: The writePersonalInfo function is called to write the populated personal information to some internal storage. The result is checked, and success or failure is printed accordingly.

4. Allocate Buffer: A buffer of size BLOCK_SIZE is allocated using malloc. This buffer is used to process and commit strings.

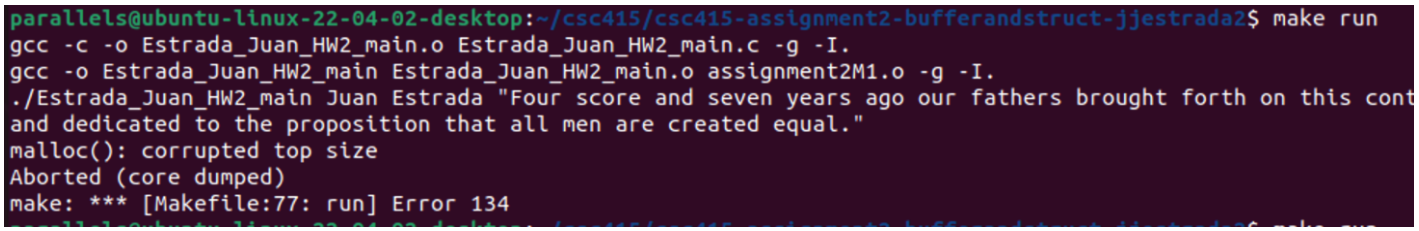
5. Process and Commit Strings: The program enters a loop that calls the getNext function to retrieve strings one by one until there are no more strings. For each string, it checks if the current buffer has enough space to accommodate it. If yes, the string is copied into the buffer, and the index is updated. If the buffer is full, it is committed using the commitBlock function, and the remaining part of the current string is copied to the buffer.

6. Final Commit and Validation: After the loop, the remaining data in the buffer is committed. The checkIt function is called to validate the stored personal information.

7. Free Memory: Finally, memory allocated for myInfo and the buffer is freed to avoid memory leaks.

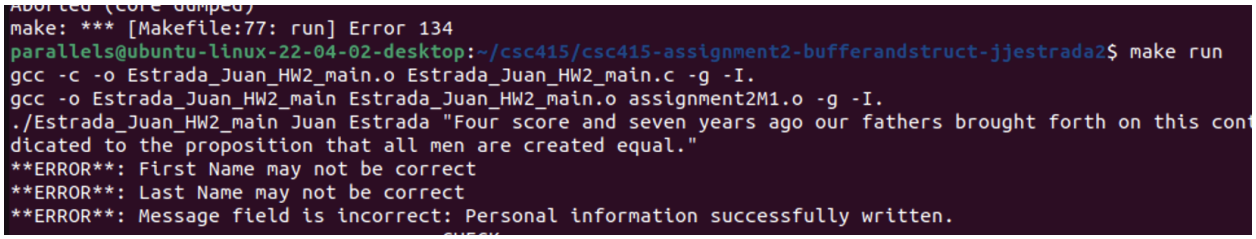
Issues and Resolutions:

1.My first issue was "malloc() corrupted top size" followed by "Aborted (core dumped)"



typically indicates a memory corruption issue in your program that has led to a segmentation fault (core dump). This error occurs when the memory allocation data structures used by malloc have been corrupted, which is a severe issue and can result in undefined behavior.
I resolved it by reviewing my code carefully, paying attention to memory allocation and deallocation. And I found that I was using uninitialized memory.

2.I encountered two additional problems. The first one was an error message indicating that the First Name might not be correct, and the second one was a similar error message suggesting that the Last Name might also be incorrect. Furthermore, there was an issue with the Message field being incorrect.



I addressed this issue by modifying the code. Instead of assigning strings using myInfo->firstName = strdup(argv[1]) and myInfo->lastName = strdup(argv[2]), I updated it to directly assign the values like firstName = argv[1] and lastName = argv[2].

3.I encountered two issues during the second part of the assignment, which involved constructing a buffer. Initially, I received an error message marked as #1. This signaled a problem in the way I was passing data to my buffer. In response, I added several print statements within my while loop to diagnose the issue. Eventually, I realized that the problem stemmed from my memcpy calls. I noticed that I had omitted the index variable that should be added to the buffer. Originally, my call looked like this: memcpy(buffer, nextString, stringLength - spaceLeft);. However, after debugging, I corrected it to: memcpy(buffer + indx, nextString, stringLength - spaceLeft);.

```
---------------------------------- CHECK ----------------------------------
Running the check for Juan Estrada
Name check is 0 by 0
Student ID: 923058731, Grade Level: Senior
Languages: 68639 (10C1F)
Message:
Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived

The Check Failed (1, 0)

END-OF-ASSIGNMENT
000000: 58 03 CB F2 FF FF 00 00  5D 03 CB F2 FF FF 00 00 | X.????..].????..
000010: 2B C2 04 37 03 00 00 00  1F 0C 01 00 46 6F 75 72 | +?.7........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
```

Analysis:
Fourth Section (000030 - 00003F):

000000: 58 A3 95 E1 FF FF 00 00 5D A3 95 E1 FF FF 00 00 | X?????..]?????..
000010: 2B C2 04 37 03 00 00 00 1F 0C 01 00 46 6F 75 72 | +?.7........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

Byte 0 to Byte 3 (4 bytes): Might be the address of that store my name in the struct

Hexadecimal: 58 A3 95 E1
In little-endian format: E1 95 A3 58
Converted to decimal: 3,040,428,536

Byte 4 to Byte 7 (4 bytes): Might be the address of that store my last name

Byte 8 to Byte 11 (4 bytes): Might be the address of that store my ID

Byte 12 to Byte 15 (4 bytes): Might be the address of that store my Languages

73 63 6F 72 65 20 61 6E 64 20 73 65 76 65 6E : This seems to be the message field with ASCII characters. The message reads "Four score and seven."

79 65 61 72 73 20 61 67 6F 20 6F 75 72 20 66: This continues the message field with the text " years ago our f."

61 74 68 65 72 73 20 62 72 6F 75 67 68 74 20 66: This part of the message continues with "thers brought f."

6F 72 74 68 20 6F 6E 20 74 68 69 73 20 63 6F 6E: This part of the message continues with "orth on this con."

74 69 6E 65 6E 74 2C 20 61 20 6E 65 77 20 6E 61: This part of the message ends with "tinent, a new na." The bytes before this are part of the message field.

Screenshot of compilation:

```
parallels@ubuntu-linux-22-04-02-desktop:~/csc415/csc415-assignment2-bufferandstruct-jjestrada2$ make
gcc -c -o Estrada_Juan_HW2_main.o Estrada_Juan_HW2_main.c -g -I.
gcc -o Estrada_Juan_HW2_main Estrada_Juan_HW2_main.o assignment2M1.o -g -I.
```

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

```
Personal information successfully written.
---------------------------------- CHECK ----------------------------------
Running the check for Juan Estrada
Name check is 0 by 0
Student ID: 923058731, Grade Level: Senior
Languages: 68639 (10C1F)
Message:
Four score and seven years ago our fathers brought forth on this continent, a new na
tion, conceived

The Check Succeded (0, 0)

END-OF-ASSIGNMENT
000000: 58 A3 95 E1 FF FF 00 00  5D A3 95 E1 FF FF 00 00 | X?????..]?????..
000010: 2B C2 04 37 03 00 00 00  1F 0C 01 00 46 6F 75 72 | +?.7........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
```