Siarhei Pushkin
Github: spushkin

Assignment 2 – Buffering and Structures

Description:

This assignment focuses on enhancing understanding of C programming, particularly with structures, pointers, buffering, and block operations. The task involves creating a program that captures personal information from command-line arguments, including name, student ID, grade level, known programming languages, and a custom message. The program then writes this data using a provided function, reads additional strings into a fixed-size buffer, processes them in chunks, and ensures all data is correctly handled and committed. The final step involves performing a check to confirm successful execution.

Approach:

My approach to the assignment started with carefully reading the requirements and understanding the provided resources, like the header file (assignment2.h) and function prototypes. I followed the assignment instructions step by step, starting with how to fill out the personalInfo structure with data from the command-line arguments. I thought about properly managing memory, allocating and freeing it as needed. Per instructions, to handle additional text data, I devised a plan to read it into a fixed-size buffer, copy data in chunks, and commit the buffer when it was full. I also thought about how to handle any leftover data in the buffer. I aimed to keep my code clean, well-organized, and well-documented throughout the process. This careful planning helped me stay organized and focused during the implementation.

Issues and Resolutions:

During the assignment, I encountered issues with memory allocation and buffer management. Initially, I struggled with correctly copying strings into the buffer without exceeding its limits. I resolved this by carefully calculating the available space and using **memcpy** to manage data chunks correctly. These steps ensured smooth execution and correct data handling throughout the program.

Analysis:

```
END-OF-ASSIGNMENT
000000: 1D 23 04 CD FF FF 00 00  25 23 04 CD FF FF 00 00 | .#.???..%#.???..
000010: 2D 73 02 37 14 00 00 00  0F 00 00 00 46 6F 75 72 | -s.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 00 | tion, conceived.
```

First Name (0x000000 to 0x000007) - pointer to the first name string (stored as 8 bytes);

Last Name (0x000008 to 0x00000F) - pointer to the last name string (stored as 8 bytes);

Student ID (0x000010 to 0x000013):

Hexadecimal value: 0x3702732D, Decimal value: 922907437 (my student ID, stored as 4 bytes);

Grade Level (0x000014 to 0x000017):

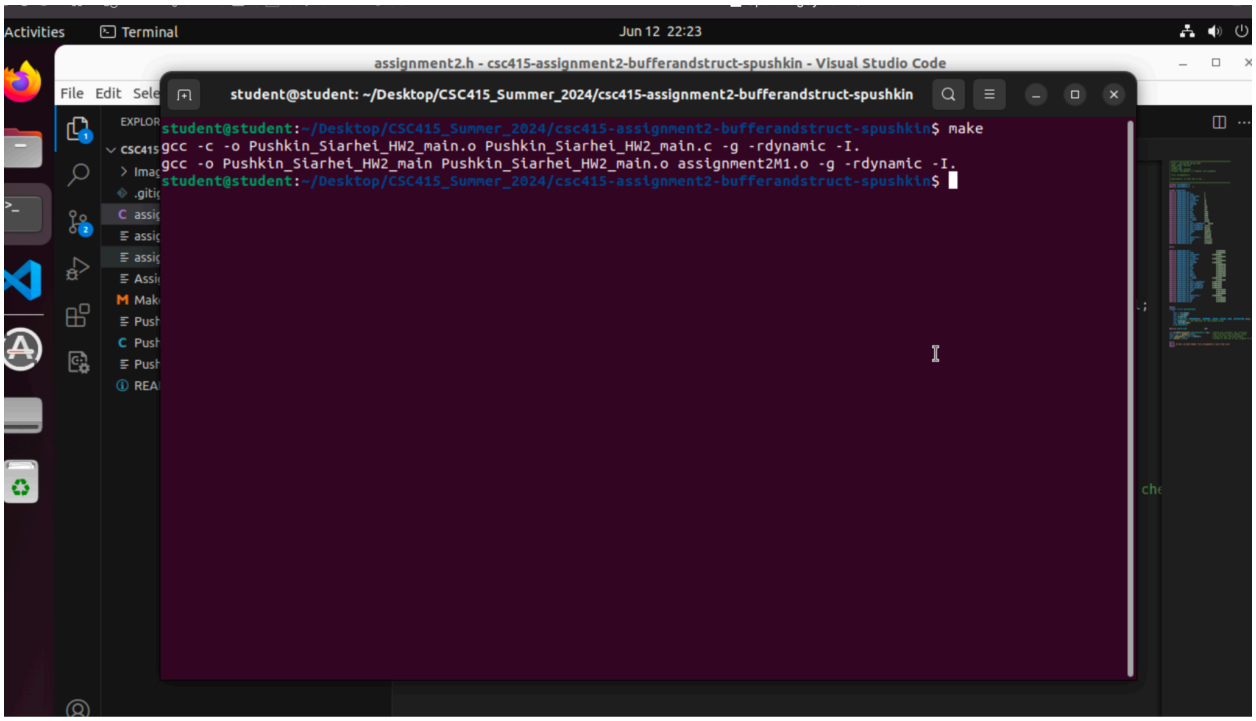
Hexadecimal value: 0x00000014, Decimal value: 20 (enum SENIOR, stored as 4 bytes);

Languages (0x000018 to 0x00001B):

Hexadecimal value: 0x0000000F, Decimal value: 15 (KNOWLEDGE_OF_C (1) + KNOWLEDGE_OF_JAVA (2) + KNOWLEDGE_OF_PYTHON (8) + KNOWLEDGE_OF_JAVASCRIPT (4) = 15, stored as 4 bytes)

Message (stored as 100-byte character array) (0x00001C to the end): ASCII value: "Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived."

Screen shot of compilation:



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