Yuvraj Gupta
Id : 922933190
Github : YuvrajGupta1808
CSC415 Operating Systems

Assignment 2 – Buffer And Struct

Description:

This assignment involves writing a C program to understand and utilize buffering and structures. The tasks include allocating and populating a personalInfo structure, writing this information, and managing a series of C strings by buffering data into blocks and committing them. The primary goal is to gain a better understanding of C programming, especially pointers, memory allocation, buffering, and binary data representation.

Approach:

1. Reading the Assignment: Carefully read and understood each step of the assignment. Reviewed the provided assignment2.h header file to understand the structure personalInfo and the required functions.
2. Review Header File: Examined assignment2.h to understand the defined constants for languages and the structure personalInfo. Noted the prototypes for the provided functions: writePersonalInfo, getNext, commitBlock, and checkIt.
3. Setup the Environment: Named my C file Gupta_Yuvraj_HW2_main.c as per the standard. Edited the Makefile to include my first and last name in the appropriate variables. Ensured all comments were meaningful and code was neatly formatted.
4. Allocate and Populate personalInfo Structure: Allocated memory for the personalInfo structure. Populated the firstName and lastName fields from the command line arguments. Set the studentID field with my actual student ID. Set the level field appropriately. Populated the languages field with at least three known languages. Copied the third command line parameter to the message field.
5. Writing Personal Information: Called the writePersonalInfo function with the populated personalInfo structure and checked the return value for success.
6. Buffering Data into Blocks: Allocated a buffer of size BLOCK_SIZE. Used the getNext function to retrieve strings and copied them into the buffer

using memcpy. Committed the buffer when it was filled or at the end of data input.

7. Final Step - Check Results: Called the checkIt function to validate the results and returned its value.
8. Analysis - Carefully analyzed the hex dump output to ensure it corresponded to the expected values in the personalInfo structure. Verified the correctness of the student ID, level, languages, and message fields in the hex dump output. Used the hex dump to confirm that the binary data matched the expected values, ensuring the correct implementation of memory allocation and data copying.

Issues:

1. **Issue 1:** The primary issue in the first while loop is that it does not properly handle cases where the data being copied might exceed the remaining space in the buffer. This can lead to incorrect memory copying and ultimately cause the checkIt function to fail.

   The first loop checks if offset + nextLen exceeds BLOCK_SIZE. If it does, it commits the block and resets the offset to 0. However, after committing the block, it doesn't handle the remaining part of data that wasn't copied. The loop starts over, losing the remaining part of the data. This means only part of the data is copied to the block, while the rest is discarded.

   When the block is full, the remaining data should be copied to the start of a new block. The first loop fails to do this.

```
while ((data = getNext()) != NULL)
{
    size_t nextLen = strlen(data);
    if (offset + nextLen >= BLOCK_SIZE) {
        commitBlock(block);
        offset = 0;
    }
    memcpy(block + offset, data, nextLen);
    offset += nextLen;
}
```

```
student@student:~/Desktop/csc415-assignment2-bufferandstruct-YuvrajGupta1808$ make run
./Gupta_Yuvraj_HW2_main Yuvraj Gupta "Four score and seven years ago our fathers brought forth on this continent, a new
 nation, conceived in Liberty, and dedicated to the proposition that all men are created equal."
---------------------------------- CHECK ----------------------------------
Running the check for Yuvraj Gupta
Name check is 0 by 0
Student ID: 922933190, Grade Level: Junior
Languages: 11
Message:
Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived

The Check Failed (-1, 192)

END-OF-ASSIGNMENT
000000: 30 D3 32 DE 45 AF 00 00  50 D3 32 DE 45 AF 00 00 | 0?2?E?..P?2?E?..
000010: C6 D7 02 37 13 00 00 00  0B 00 00 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 00 | tion, conceived.

Check result: -1
make: *** [Makefile:70: run] Error 255
student@student:~/Desktop/csc415-assignment2-bufferandstruct-YuvrajGupta1808$
```

Test Failed

Solution: The correct approach is shown in the code, which correctly handles partial data copying and ensures no data is lost.

The correct approach calculates how much space is left in the current block using size_t space_left = BLOCK_SIZE - offset;.

If len exceeds space_left, copy only space_left bytes to the block. Commit the filled block, reset offset to 0, and adjust the data pointer and len to handle the remaining data. After handling the part that fits, copy any remaining data into the new block.

2. **Issue:** Understanding Hex Dump Had difficulty understanding the hex dump output for analysis.

   Solution: Referred to structure definitions and cross-verified values to interpret the dump correctly.

Analysis:

```
student@student:~/Desktop/csc415-assignment2-bufferandstruct-YuvrajGupta1808$ make run
./Gupta_Yuvraj_HW2_main Yuvraj Gupta "Four score and seven years ago our fathers brought forth on this continent, a new
 nation, conceived in Liberty, and dedicated to the proposition that all men are created equal."
First Name Pointer: 0xc0f46e20d330, First Name: Yuvraj
Last Name Pointer: 0xc0f46e20d350, Last Name: Gupta
---------------------------------- CHECK ----------------------------------
Running the check for Yuvraj Gupta
Name check is 0 by 0
Student ID: 922933190, Grade Level: Junior
Languages: 11
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: 30 D3 20 6E F4 C0 00 00  50 D3 20 6E F4 C0 00 00 | 0? n??..P? n??..
000010: C6 D7 02 37 13 00 00 00  0B 00 00 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 00 | tion, conceived.

Check result: 0
student@student:~/Desktop/csc415-assignment2-bufferandstruct-YuvrajGupta1808$ make run
```

```
000000: 30 D3 20 6E F4 C0 00 00 50 D3 20 6E F4 C0 00 00  | 0.? n??..P.? n??..
000010: C6 D7 02 37 13 00 00 00 0B 00 00 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
```

Memory Address Interpretation:

In a 64-bit system, memory addresses, or pointers, are represented as 8 bytes (64 bits) in hexadecimal notation. Each byte consists of two hexadecimal digits, which represent 8 bits. These pointers are crucial for accessing data stored in different locations in memory. When we allocate memory for a variable or a structure in C, the memory address of the allocated space is stored in a pointer variable. This address can be dereferenced to access or manipulate the data stored at that location.

For instance, the first 8 bytes in our hex dump, 30 D3 20 6E F4 C0 00 00, represent a pointer to the memory location where the first name is stored. This value is in little-endian format, which means the least significant byte is stored first. To interpret this correctly, we need to read it as 0x0000C0F46E20D330. This address is where the actual data for the first name, "Yuvraj," is located in memory. The same logic applies to the next 8 bytes, 50 D3 20 6E F4 C0 00 00, which point to the memory address of the last name "Gupta," read as 0x0000C0F46E20D350.

To verify the correctness, we compare these addresses with the output of the pointers printed by the program. The printed pointers should match the interpreted addresses from the hex dump. If they do, this confirms that our pointers are correctly pointing to the intended memory locations of the strings.

First Line:

<u>First 4 Bytes - First Name Pointer (Red Color):</u>

- Hexadecimal in Dump: 30 D3 20 6E F4 C0 00 00
- Interpreted Address: 0x0000C0F46E20D330
- Printed Pointer: 0xc0f46e20d330
- Comparison: The printed pointer 0xc0f46e20d330 matches the interpreted address 0x0000C0F46E20D330, confirming it points to the correct memory location of the first name “Yuvraj."

<u>First 4 Bytes - Last Name Pointer (Blue Color):</u>

- Hexadecimal in Dump: 50 D3 20 6E F4 C0 00 00

- Interpreted Address: 0x0000C0F46E20D350
- Printed Pointer: 0xc0f46e20d350
- Comparison: The printed pointer 0xc0f46e20d350 matches the interpreted address 0x0000C0F46E20D350, confirming it points to the correct memory location of the last name "Gupta."

Second Line:

First 4 Bytes - Student ID (Magenta Color):

- Hex Value: C6 D7 02 37
- Interpretation: This is a little-endian representation of the student ID.
- Verification: Convert Student ID to Hexadecimal - Decimal 922933190 to Hexadecimal: 3702D7C6
- Little-endian Representation - Reverse the order: C6 D7 02 37
- Comparison: This confirms that the value C6 D7 02 37 in the hex dump correctly represents the student ID 922933190.

Next 4 Bytes - Level (Purple Color):

- Hex Value: 13 00 00 00
- Interpretation: Each enumerator, like FRESHMAN (17) and SOPHOMORE (18), represents an integer constant. The bytes 13 00 00 00 in the hex dump, reversed due to little-endian ordering, correspond to 19, representing JUNIOR.
- Verification: Convert Level to Hexadecimal - Decimal 19 (which corresponds to "Junior") to Hexadecimal: 13
- Little-endian Representation: 13 00 00 00 (since 13 is already in the least significant byte)
- Comparison: This confirms that the value 13 00 00 00 in the hex dump correctly represents the grade level “Junior".

Next 4 Bytes - Languages (Green Color):

- Hex Value: 0B 00 00 00

- Interpretation: This is a little-endian representation of the languages field.
- Verification: Binary Representation - 0B in Hexadecimal is 1011 in Binary. This indicates knowledge of C (1), Java (2), and Python (8), summing up to 11.
- Convert Languages to Hexadecimal: Decimal 11 to Hexadecimal: 0B
- Little-endian Representation: 0B 00 00 00
- Comparison: This confirms that the value 0B 00 00 00 in the hex dump correctly represents the languages field with knowledge of C, Java, and Python.

Last 4 Bytes - Message (Orange Color):

- Hex Value: 46 6F 75 72
- Interpretation: These are ASCII values representing the text.
- Verification: Convert Hexadecimal to ASCII:
  46 -> F 6F -> o 75 -> u 72 -> r
- This confirms that the value 46 6F 75 72 correctly represents the ASCII text “Four".
- These bytes represent ASCII characters that continue the message.

Third Line (Orange Color)**:**

- Hex Value: 20 73 63 6F 72 65 20 61 6E 64 20 73 65 76 65 6E
- Interpretation: These are ASCII values representing the text.
- Verification: 20 -> “ ”, 73 -> s, 63 -> c, 6F -> o, 72 -> r, 65 -> e, 20 -> “ ”, 61 -> a, 6E -> n, 64 -> d, 20 -> “ ”, 73 -> s, 65 -> e, 76 -> v, 65 -> e, 6E -> n
- This confirms that the hex dump correctly represents the part of the message " score and seven”.

Fourth Line (Orange Color)**:**

- Hex Value: 20 79 65 61 72 73 20 61 67 6F 20 6F 75 72 20 66
- Interpretation: These are ASCII values representing the text.
- Verification: 20 -> " ", 79 -> y, 65 -> e, 61 -> a, 72 -> r, 73 -> s, 20 -> " ", 61 -> a, 67 -> g, 6F -> o, 20 -> " ", 6F -> o, 75 -> u, 72 -> r, 20 -> " ", 66 -> f

- This confirms that the hex dump correctly represents the part of the message " years ago our f”.

Fifth Line (Orange Color)**:**

- Hex Value: 61 74 68 65 72 73 20 62 72 6F 75 67 68 74 20 66
- Interpretation: These are ASCII values representing the text.
- Verification: 61 -> a, 74 -> t, 68 -> h, 65 -> e, 72 -> r, 73 -> s, 20 -> " ", 62 -> b, 72 -> r, 6F -> o, 75 -> u, 67 -> g, 68 -> h, 74 -> t, 20 -> " ", 66 -> f
- This confirms that the hex dump correctly represents the part of the message "athers brought f”.

Sixth Line (Orange Color)**:**

- Hex Value: 6F 72 74 68 20 6F 6E 20 74 68 69 73 20 63 6F 6E
- Interpretation: These are ASCII values representing the text.
- Verification: 6F -> o, 72 -> r, 74 -> t, 68 -> h, 20 -> " ", 6F -> o, 6E -> n, 20 -> " ", 74 -> t, 68 -> h, 69 -> i, 73 -> s, 20 -> " ", 63 -> c, 6F -> o, 6E -> n
- This confirms that the hex dump correctly represents the part of the message "orth on this con”.

Seventh Line (Orange Color)**:**

- Hex Value: 74 69 6E 65 6E 74 2C 20 61 20 6E 65 77 20 6E 61
- Interpretation: These are ASCII values representing the text.
- Verification: 74 -> t, 69 -> i, 6E -> n, 65 -> e, 6E -> n, 74 -> t, 2C -> ,, 20 -> " ", 61 -> a, 20 -> " ", 6E -> n, 65 -> e, 77 -> w, 20 -> " ", 6E -> n, 61 -> a
- This confirms that the hex dump correctly represents the part of the message "tinent, a new na”.

Eight Line (Orange Color)**:**

- Hex Value: 74 69 6F 6E 2C 20 63 6F 6E 63 65 69 76 65 64 20
- Interpretation: These are ASCII values representing the text.

- Verification: 74 -> t, 69 -> i, 6F -> o, 6E -> n, 2C -> ,, 20 -> " ", 63 -> c, 6F -> o, 6E -> n, 63 -> c, 65 -> e, 69 -> i, 76 -> v, 65 -> e, 64 -> d, 20 -> " "
- This confirms that the hex dump correctly represents the part of the message "tion, conceived ".

Screen shot of compilation:

```
student@student:~/Desktop/csc415-assignment2-bufferandstruct-YuvrajGupta1808$ make
gcc -c -o Gupta_Yuvraj_HW2_main.o Gupta_Yuvraj_HW2_main.c -g -rdynamic -I.
gcc -o Gupta_Yuvraj_HW2_main Gupta_Yuvraj_HW2_main.o assignment2M1.o -g -rdynamic -I.
student@student:~/Desktop/csc415-assignment2-bufferandstruct-YuvrajGupta1808$
```

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

Test 1: Base Case

```
student@student:~/Desktop/csc415-assignment2-bufferandstruct-YuvrajGupta1808$ make run
./Gupta_Yuvraj_HW2_main Yuvraj Gupta "Four score and seven years ago our fathers brought forth on this continent, a new
 nation, conceived in Liberty, and dedicated to the proposition that all men are created equal."
---------------------------------- CHECK ----------------------------------
Running the check for Yuvraj Gupta
Name check is 0 by 0
Student ID: 922933190, Grade Level: Junior
Languages: 11
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: 30 93 75 60 0B BF 00 00  50 93 75 60 0B BF 00 00 | 0?u`.?..P?u`.?..
000010: C6 D7 02 37 13 00 00 00  0B 00 00 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 00 | tion, conceived.

Check result: 0
student@student:~/Desktop/csc415-assignment2-bufferandstruct-YuvrajGupta1808$
```

Test 2: When RUNOPTIONS="Yuvraj Gupta \”There will be a message printed\”"

```
student@student:~/Desktop/csc415-assignment2-bufferandstruct-YuvrajGupta1808$ make run RUNOPTIONS="Yuvraj Gupta \"There
 will be a message printed here\""
./Gupta_Yuvraj_HW2_main Yuvraj Gupta "There will be a message printed here"
---------------------------------- CHECK ----------------------------------
Running the check for Yuvraj Gupta
Name check is 0 by 0
Student ID: 922933190, Grade Level: Junior
Languages: 11
Message:
There will be a message printed here

The Check Succeded (0, 0)

END-OF-ASSIGNMENT
000000: 30 73 53 7E B2 B5 00 00  50 73 53 7E B2 B5 00 00 | 0sS~??..PsS~??..
000010: C6 D7 02 37 13 00 00 00  0B 00 00 00 54 68 65 72 | ??.7........Ther
000020: 65 20 77 69 6C 6C 20 62  65 20 61 20 6D 65 73 73 | e will be a mess
000030: 61 67 65 20 70 72 69 6E  74 65 64 20 68 65 72 65 | age printed here
000040: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000050: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000060: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000070: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................

Check result: 0
student@student:~/Desktop/csc415-assignment2-bufferandstruct-YuvrajGupta1808$
```

Test 3: When RUNOPTIONS="Yuvraj Gupta \”Buffering in computer systems is crucial for managing data flow between devices or processes, ensuring smooth and efficient operations.\”"

```
student@student:~/Desktop/csc415-assignment2-bufferandstruct-YuvrajGupta1808$ make run RUNOPTIONS="Yuvraj Gupta \"Buffe
ring in computer systems is crucial for managing data flow between devices or processes, ensuring smooth and efficient
operations.\""
./Gupta_Yuvraj_HW2_main Yuvraj Gupta "Buffering in computer systems is crucial for managing data flow between devices o
r processes, ensuring smooth and efficient operations."
---------------------------------- CHECK ----------------------------------
Running the check for Yuvraj Gupta
Name check is 0 by 0
Student ID: 922933190, Grade Level: Junior
Languages: 11
Message:
Buffering in computer systems is crucial for managing data flow between devices or processes, ensur

The Check Succeded (0, 0)

END-OF-ASSIGNMENT
000000: 30 E3 EB F2 56 C0 00 00  50 E3 EB F2 56 C0 00 00 | 0???V?..P???V?..
000010: C6 D7 02 37 13 00 00 00  0B 00 00 00 42 75 66 66 | ??.7........Buff
000020: 65 72 69 6E 67 20 69 6E  20 63 6F 6D 70 75 74 65 | ering in compute
000030: 72 20 73 79 73 74 65 6D  73 20 69 73 20 63 72 75 | r systems is cru
000040: 63 69 61 6C 20 66 6F 72  20 6D 61 6E 61 67 69 6E | cial for managin
000050: 67 20 64 61 74 61 20 66  6C 6F 77 20 62 65 74 77 | g data flow betw
000060: 65 65 6E 20 64 65 76 69  63 65 73 20 6F 72 20 70 | een devices or p
000070: 72 6F 63 65 73 73 65 73  2C 20 65 6E 73 75 72 00 | rocesses, ensur.

Check result: 0
student@student:~/Desktop/csc415-assignment2-bufferandstruct-YuvrajGupta1808$
```