

Assignment 2 – Buffer and Struct

Description:

Structures, pointers, character strings, enumerated types, bitmap fields, and data buffering into blocks will all be included in this assignment. The code component is separated into four main steps. The first step is to create memory for `personallInfo` and assign each value, some of this by using command line options. The second step is to use `writepersonallInfo` to see whether the process was successful. The third step is to establish a buffer to hold the string returned by `getNext()` until it returns `NULL`. Because the buffer size is 256, we should reuse it. When the buffer is filled, we call the `commitBlock()` function to commit the data.

Approach / What I Did:

I used `malloc` to allocate a memory area for `personallInfo` using `sizeof(personallInfo)`. I chose `sizeof(personallInfo)` since it is the object for which we are allocating space and should be created with that size. Then, assign `firstName` and `lastName` to `argv[1]` and `argv[2]`, which will retrieve my first and last name from the command line parameter.

I just assign `studentID` and `level` directly with actual numbers. To prevent extra hardcoding, I utilized the name of each compile time constant for language. I used `strncpy()` instead of `strcpy()` for message because the message array only has 100 bytes of memory, and `strncpy()` allows me to ensure that it copies the correct amount of char into it. When troubleshooting, I used `printf()` to print the return result from `writePersonallInfo()` to see whether I accomplished step four accurately.

The core algorithm is included in the while loop, which controls the looping logic until I receive `NULL` from `getNext()`. Because we must reuse the same buffer, there are two scenarios that occur when I attempt to store extra characters into the buffer. The first is that the buffer has enough capacity, which is the length of the string + the current buffer size is less than `BLOCK SIZE`. In this scenario, I add the string with `strncat` so that the new data does not overwrite the existing data. Another issue is that the buffer does not have enough space to accommodate the string. In this scenario, I maintain track of the remaining string by storing it (using `strncat`) until the buffer is filled.

Issues and Resolutions:

My issue was a misunderstanding of how to use `strcpy` and `strcat`. I used `strcpy` to copy everything into the buffer, even if it wasn't full. I discovered that when I use `strcpy`, it replaces the old characters in the buffer, causing me to lose data before `commitBlock()`. When the buffer is not full, I use `strcat` instead of `strcpy`, and the characters add to the preceding characters.

Screen shot of compilation:

```
parallels@ubuntu-linux-20-04-desktop:~/Documents/csc415-assignment2-bufferandstruct-pragati-e$ make
gcc -c -o makani_pragati_HW2_main.o makani_pragati_HW2_main.c -g -I.
gcc -o makani_pragati_HW2_main makani_pragati_HW2_main.o assignment2M1.o -g -I.
```

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

```
parallels@ubuntu-linux-20-04-desktop:~/Documents/csc415-assignment2-bufferandstruct-pragati-e$ make run
./makani_pragati_HW2_main pragati makani "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 pragati makani
Name check is 0 by 0
Student ID: 920821704, Grade Level: Junior
Languages: 2071 (817)
Message:
Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived

The Check Succeeded (0, 0)

END-OF-ASSIGNMENT
000000: BC 02 D4 CF FF FF 00 00 C4 02 D4 CF FF FF 00 00 | ?.????..?..
000010: C8 9F E2 36 02 00 00 00 17 08 00 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
parallels@ubuntu-linux-20-04-desktop:~/Documents/csc415-assignment2-bufferandstruct-pragati-e$
parallels@ubuntu-linux-20-04-desktop:~/Documents/csc415-assignment2-bufferandstruct-pragati-e$
parallels@ubuntu-linux-20-04-desktop:~/Documents/csc415-assignment2-bufferandstruct-pragati-e$
parallels@ubuntu-linux-20-04-desktop:~/Documents/csc415-assignment2-bufferandstruct-pragati-e$
```