

PA0: Environment Setup, AddressSanitizer and GDB

CSCE 313 – 512

Due Date: August 28, 2020

**Debugging with DGB:**

After setting up the system and compiling the buggy code, I fixed the errors by adding the following code into the blanks:

Blank A:

#include <vector>

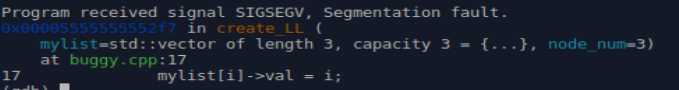
using namespace std;

Blank B:

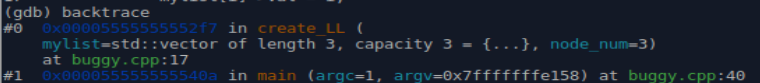
public:

After filling in the blankes in the code, I corrected the statements in likes 15, 16, 21, 28, and 29 to make a member variable of an object accesed through a pointer (ex. ret += ptr->val).

After this, I went on to fix the runtime errors. I ran my program as usual and launched it under gdb. The printed variable names or line numbers to locate the errors were in some internal address format, so I compiled my code with the -g option. The following error was caught on line 17:



After using “backtrace”, I found where the segmentation fault was occuring.



After setting the breakpoint and running the program from the beginning, I printed the contents of mylist.

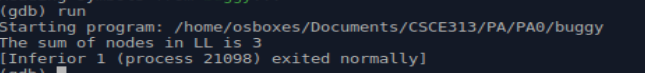


To fix the segmentation error, I filled in Blank C.

Blank C:

mylist[i] = new node;

After this, I got another segmentation fault in the sum\_LL function. This was due to going “out of range” in mylist on the last itteration. I fixed this fault by itterating the loop till the condition i < node\_num – 1 and got the following output:



Next, I filled in Blank D to free the dynamically allocated memory from the heap for elements of mylist to avoid memory leaks.

Blank D:

for (int i=o, I < NODE\_NUM; i++){

delete mylist[i];

mylist[i] = nullptr;

}

**Debugging with AddressSanitizes:**

After setting up the system and compiling the buggy code, I fixed the errors by adding the following code into the blanks:

Blank A:

#include <vector>

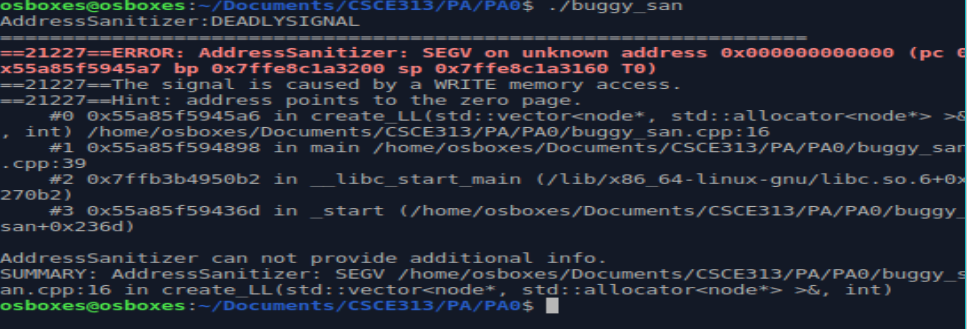
using namespace std;

Blank B:

public:

After filling in the blankes in the code, I corrected the statements in likes 15, 16, 21, 28, and 29 to make a member variable of an object accesed through a pointer (ex. ret += ptr->val).

After running the program, I obtained the following fault:

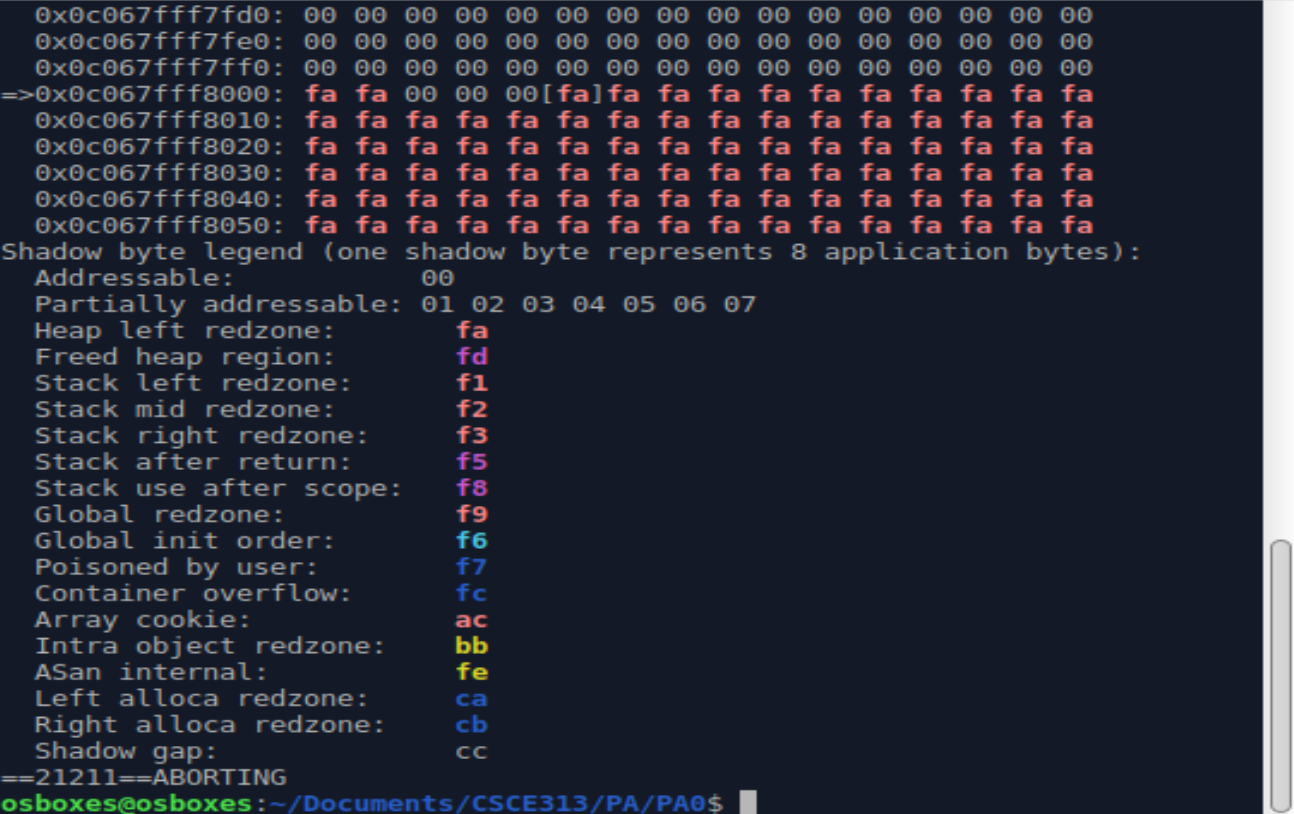


Thus, I filled in Blank C:

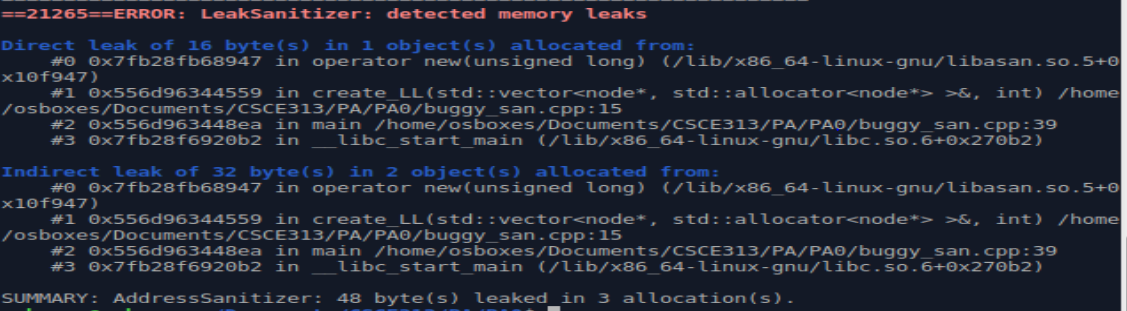
Blank C:

mylist[i] = new node;

I received the following output before I fixed the “out of range” error in mylist on the last itteration. I fixed this fault by itterating the loop till the condition i < node\_num – 1.



After fixing that error, I ran my code again and recived an erorr that detected memory leaks.



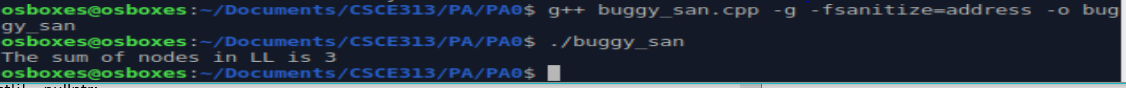
Thus, I addressed the deletion of dynamically allocated memory by filling in Blank D:

Blank D:

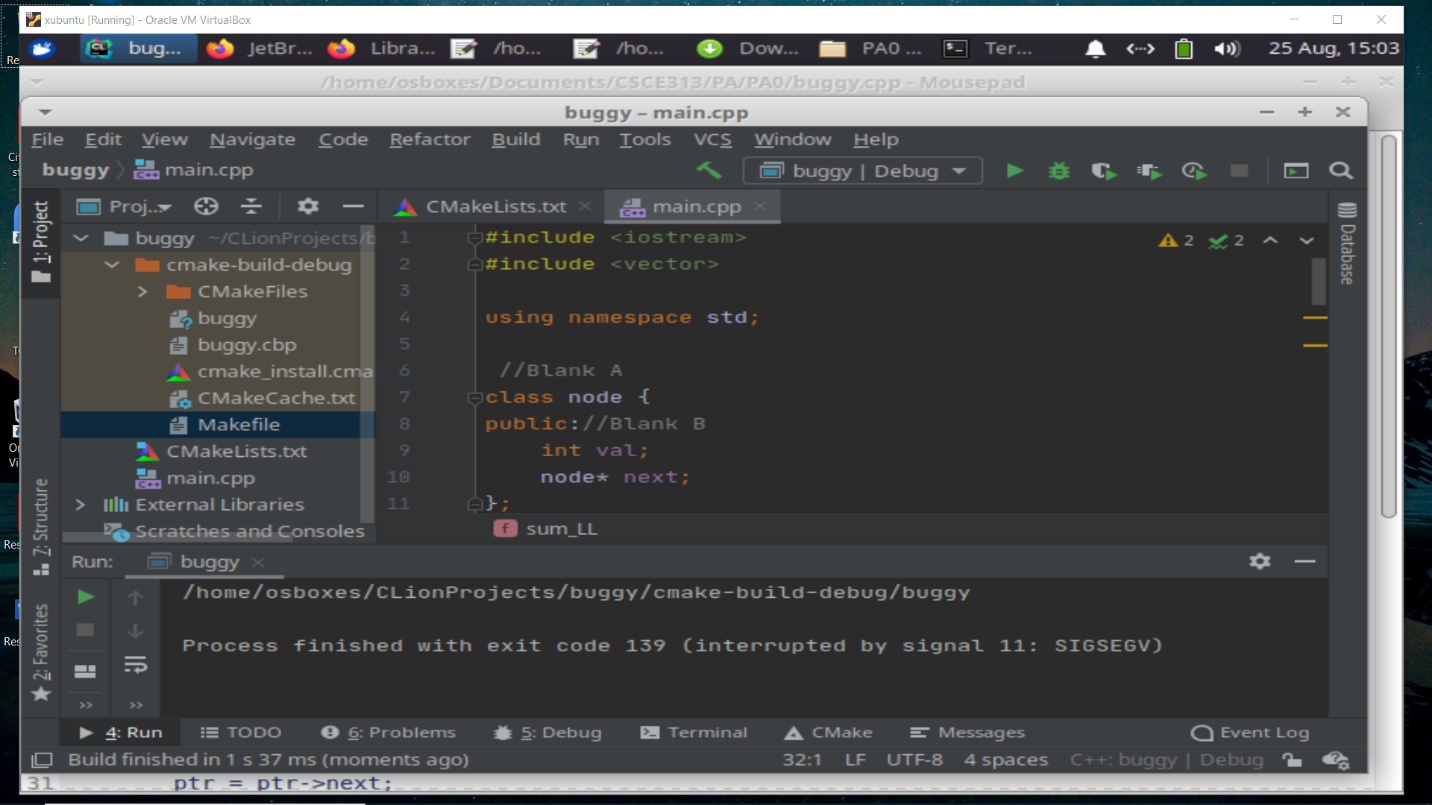
for (int i=o, I < NODE\_NUM; i++){

delete mylist[i];

mylist[i] = nullptr;}

The output of the code after fixing all of the errors is as follows:

**IDE:**

The following image shows the 10th step of the PA of repeating the process in the steps above.