LW: Pass By Reference

Due by Fri, 15 Feb 2019 08:00:00-0600

Objectives

- To understand that:
    - argument passing is similar to initialization,
    - when a function is called, each formal argument is initialized by its corresponding actual argument, and
    - when a reference is provided as a formal parameter, the corresponding actual argument will be referred to by the formal parameter.

Labwork

- There are questions throughout this document that you will need to respond to for credit.
- There are questions at the end of this document that you will respond to for credit.
- Using Putty (PC) or terminal (Mac), log-in to **`compute.cse.tamu.edu`**
- Create an empty directory for this labwork. In a terminal (e.g. putty) navigate to this directory.
- Download the source code from https://drive.google.com/open?id=0B_ouNNuWgNZCVm5udjlUaXpBV1E and move it into a directory in your h:drive
- Verify you copied the file with the following command:
```
ls
```
- Compile using the following command:
```
g++-7.2.0 -std=c++17 -Wall -Wextra -pedantic
-fsanitize=address,undefined *.cpp
```
- Inspect the source code; notice that the source code is identical to that in LW: Pass By Value, with the exception that the type of formal parameter to vint_half_sum is now a reference.
    a. How did the declaration of `vint_half_sum` change in order to pass the argument by reference instead of by value?

An ‘&’ sign was included after the data type.

    b. How did the definition of `vint_half_sum` change in order to pass the argument by reference instead of by value?

An ‘&’ sign was included before the variable name.

c. With this in mind, understand that you can view the initialization of the formal parameter `vector<int>& v` in the function call to `vint_half_sum` on line 34 with `vint` as:
    i. vector<int>& v = vint;

- Compile and run the source code.
- Carefully, observe the output printed to the screen and then answer the following questions in the fields provided:
    a. Explain why the modification of `v` in `vint_half_sum` *does* mutate the actual argument `vint:`

    In the pass-by-reference, a new memory address is not allocated and we are modifying the original vector at its location in memory.

    b. What information included in the output produced by the calls to `vis::print` in `main` with `vint` and in `vint_half_sum` supports your response to 8a?

    The values of the vector change, but the memory address stay the same.

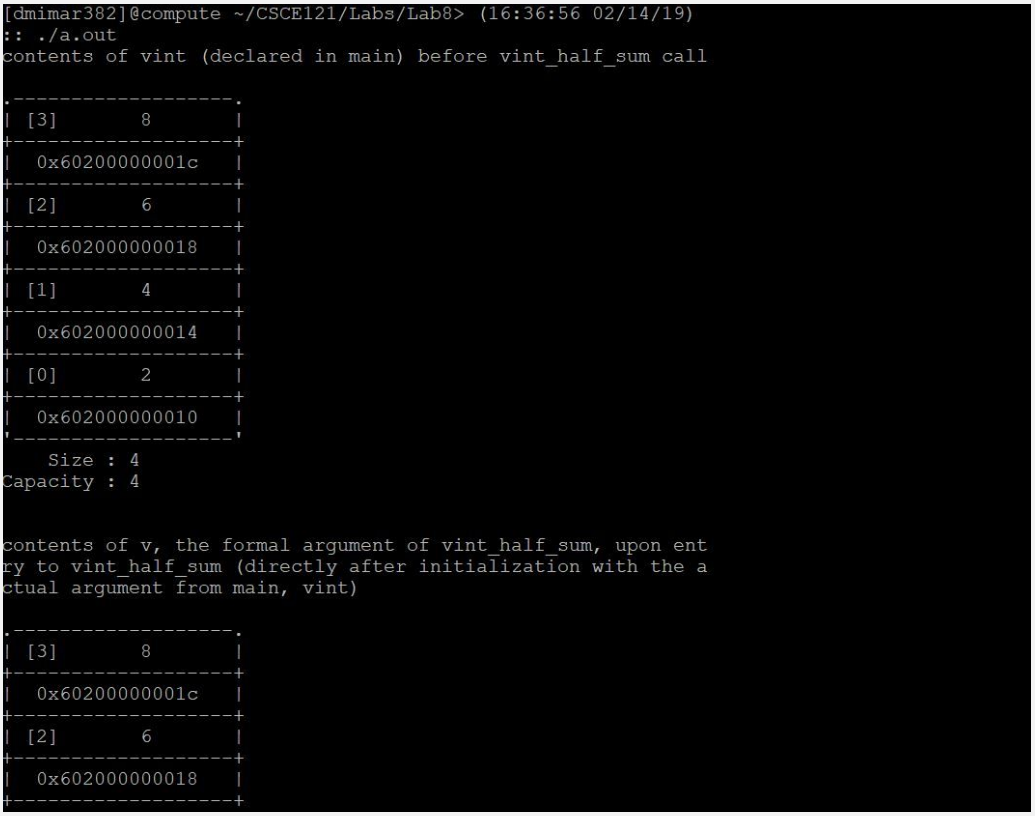
    c. What is the difference between pass-by-value and pass-by-reference? When might you use one over the other? Do you see any potential pitfalls in using either method?

Pass by value creates a copy of the vector at a new memory address and modifies the vector values at that new memory address. Pass by reference directly modifies the stored values at the original memory address. We will use pass-by-reference when we want to modify the original data being stored. Otherwise,if we are only manipulating the data, we will use pass-by-value.

d. It is recommended that when passing a parameter by reference to a function you should denote in the function's identifier if it modifies the object in which a formal argument refers. Why do you think that this is a good idea? For instance, why might we decide to update the name of `vint_half_sum` to `half_elems_of_vint_ret_sum`?

By updating the name, we ensure that anyone looking at the code knows that the original information stored at that memory address has been modified.

e. Capture the output written to the terminal window by this program in the form of a screenshot; if you cannot include everything, that's okay. Drag and drop or paste your screenshot into the box below:

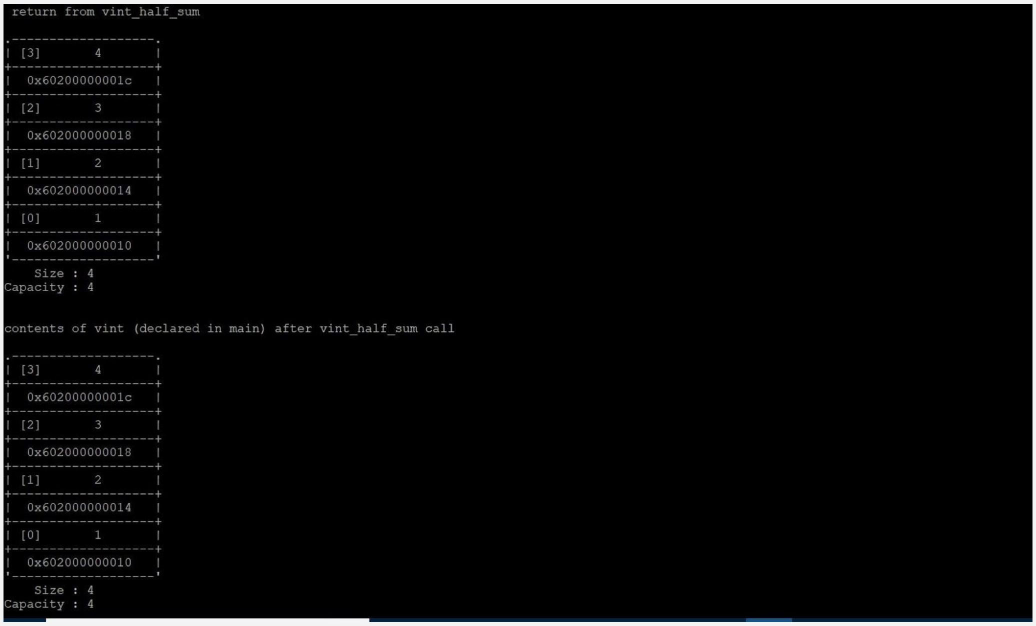


```
[dmimar382]@compute ~/CSCE121/Labs/Lab8> (16:36:56 02/14/19)
:: ./a.out
contents of vint (declared in main) before vint_half_sum call

.--------------------.
| [3]       8        |
+--------------------+
|  0x60200000001c    |
+--------------------+
| [2]       6        |
+--------------------+
|  0x602000000018    |
+--------------------+
| [1]       4        |
+--------------------+
|  0x602000000014    |
+--------------------+
| [0]       2        |
+--------------------+
|  0x602000000010    |
'--------------------'
    Size : 4
Capacity : 4


contents of v, the formal argument of vint_half_sum, upon ent
ry to vint_half_sum (directly after initialization with the a
ctual argument from main, vint)

.--------------------.
| [3]       8        |
+--------------------+
|  0x60200000001c    |
+--------------------+
| [2]       6        |
+--------------------+
|  0x602000000018    |
+--------------------+
```



```
 return from vint_half_sum

.--------------------.
| [3]       4        |
+--------------------+
|  0x60200000001c    |
+--------------------+
| [2]       3        |
+--------------------+
|  0x602000000018    |
+--------------------+
| [1]       2        |
+--------------------+
|  0x602000000014    |
+--------------------+
| [0]       1        |
+--------------------+
|  0x602000000010    |
'--------------------'
    Size : 4
Capacity : 4


contents of vint (declared in main) after vint_half_sum call

.--------------------.
| [3]       4        |
+--------------------+
|  0x60200000001c    |
+--------------------+
| [2]       3        |
+--------------------+
|  0x602000000018    |
+--------------------+
| [1]       2        |
+--------------------+
|  0x602000000014    |
+--------------------+
| [0]       1        |
+--------------------+
|  0x602000000010    |
'--------------------'
    Size : 4
Capacity : 4
```

Submission

- Save this completed labwork as a PDF [File -> Download As -> PDF Document (.pdf)] and submit to Gradescope for grading.