

2020 Fall CPSC 240

Assignment 3: Array Sort

Goals

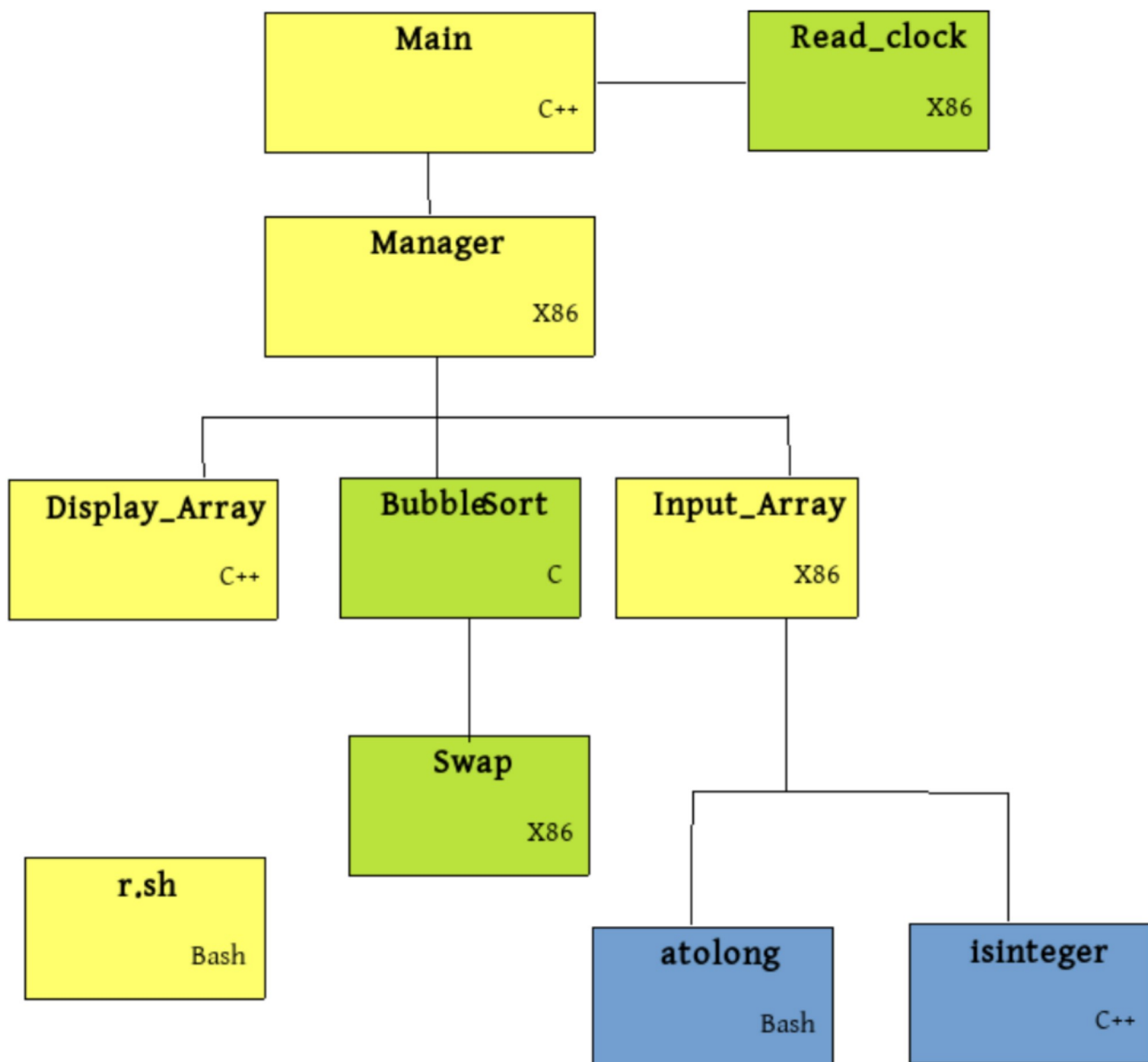
Make an assembly program that teaches all of the following:

- =pass by pointer

- =how to read the CPU's own clock

Programming requirements

This assignment 3 is all about adding a few new modules to the old assignment 2.



The modules in light beige were created by you for Assignment 2.

The blue modules are free library functions. They are free downloads. There is nothing for you to do here. Do not make your own version of `isinteger` or `atolong`.

The green modules are what assignment 3 is all about. You have to create the green ones. Well, you don't have to create 3 green ones you only have to create 2 green ones because one of the green modules is the bubble sort. You simply copy the bubble sort from one of your old textbooks and re-use it. That leaves two green modules for you to build as new ones.

Please copy Bubble Sort from a textbook or from the web. Do not ask a classmate for help debugging the sort function because you made it yourself.

If you don't like Bubble Sort you may substitute another sort. The requirement is that somewhere in the body of the sort function there must be a place where two elements of the array are swapped. Most sort algorithms meet this requirement.

The swap function is written in assembly. Swapping two values is such a common activity you would think that this function should be in a library. For now treat it as an application function, that is, a function (module) created specifically for use in one specific application.

You may use xor to perform the swap. Typically xor is a fast efficient way to perform a swap of two integers. This case is different.

There is a rule in the assembler that you can never have two memory operands in the same instruction. That is why you can add a number on top of the stack to a register as in:

```
add rcx, qword [rsp]
```

That example shows one register operator and one memory operator.

You cannot have two memory operators:

```
add qword [rsp], qword [rsp+8]
```

That is an attempt to add the second number in the stack to the first number in the stack. That is an illegal instruction.

In the case of swap the two numbers arrive as two pointers. The natural tendency is to do this:

```
xor [rdi], [rsi]
```

```
xor [rsi], [rdi]
```

```
xor [rdi], [rsi]
```

But, two memory operands in the same instruction is an error. Bad luck. You have to find an alternative strategy.

Sample run #1

The time on the CPU clock is now 2547892 tics
Welcome to Array Sorting program
Bought to you by Janice Ward

This program will sum your array of integers
Enter a sequence of long integers separated by white space.
After the last input press enter followed by Control+D:

```
3  
-12  
9  
21  
-4  
13 <enter> <cntl+D>
```

These number were received and placed into the array:
3 -12 9 21 -4 13

The array has been sorted by the bubble sort algorithm

This is the order of the values in the array now:
-12 -4 3 9 13 21

The largest number in the array will now be returned to the main function.

Main received 21, and plans to keep it.
Main will return 0 to the operating system. Bye.
The time on the CPU clock is now 5778930 tics.

Color codes:

Blue background is produced by module Input_Array

Yellow background is produced by module Manager

Green background is produced by module Display_Array

White background is produced by module Main

Module sum does not produce any visible output.

Sample run #2

The time on the CPU clock is now 16747300 tics.

Welcome to Arrays of Integers

Bought to you by Janice Ward

This program will sum your array of integers

Enter a sequence of long integers separated by white space.

After the last input press enter followed by Control+D:

3

-12

9

2R7

The last input was invalid and not entered into the array.

-4

5 <enter><cntl+D>

These numbers were received and placed into the array:

3 -12 9 -4 7

The array has been sorted by the bubble sort algorithm

This is the order of the values in the array now:

-12 -4 3 7 9

The largest number in the array will now be returned to the main function.

Main received 3, and is not sure what to do with it.

Main will return 0 to the operating system. Bye.

The time on the CPU clock is now 39749222 tics.

Other facts

The language used in each module is shown in the individual rectangles.

Obviously you change the name of the author in the example to be your own name.

Make a bash file that compiles and runs everything related to this program.

Make this program qualify for the Professional Software Certificate. That means your Assignment 3 is so good that it is suitable for demonstration and delivery to the Human Resource Job Interview team where you are applying for employment.

Do not change the comments, the author, the license, or any other part of a library functions if your only intent is to use it. If you plan to create a new derived function from an existing library function then, of course, you add new information about yourself as new author and new information about the enhancement you implemented. The derived software must carry the same license as the parent software, but now updated with the new author as copyright holder and new year of copyright.

The list of modules (files) should include the library modules and you should include library files when you ship the final product to the customer. I am the customer.

When you are done

This assignment can count for extra credit. It is not a requirement to do this programming exercise. Doing this assignment is strongly recommended.

When your program has functional correctness then make it qualify for Professional Status. Professional Software has been explained repeatedly this semester. You know what it is.

Send the program to me in an email message with 10 attachments. In the subject line place the string "CPS 240 Assignment 3 for Credit".

Don't email me broken code. It is better to discuss fixing code during the lab portion of the class meeting.

Email: holliday@fullerton.edu

November 1, 2020 is the last day I will accept Assignment 3 for credit.

Friendly reminder: Assignment 3 is valid for extra credit until the date given above, but it may also be valuable for its knowledge that may appear on the midterm. Just a reminder.

You have heard me say I want to avoid students who dump a lot of programs on me to grade during the last 3 days of the semester. This has happened in the past. I cannot physically manage that many programs arriving close to final exams.

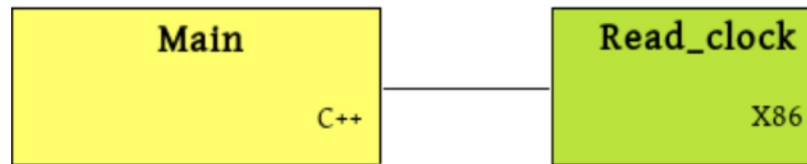
November 22-29 is fall break week.

Appendix: How to complete Assignment 3 without Assignment 2.

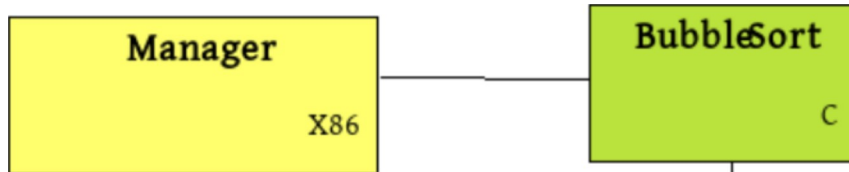
When I evaluate Assignment 3 I am interested in modules: read_clock, bubble_sort, and swap, and how those 3 modules connect with the rest of the system.

Here are the connections: of interest:

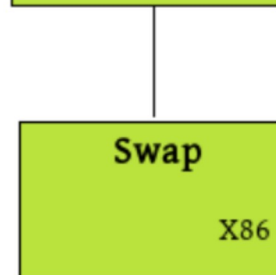
C++ calls X86:



X86 calls C:



C calls X86:



According to the email I've receive: the biggest issue is with input_array. For assignment 3 input_array is no longer being evaluated. Make an input_array in C++ and use it. I have posted an example of how to terminate a loop in C++ using control+D. Note that input_array can still call atolong and isinteger. You are free to substitute atol for atolong.

You can still complete Assignment 3 by using (mostly) C++ as the foundation.

Announcements

I have been asked numerous times who is Chris Sawyer?

You probably know the King of Rock.

You probably know the King of Techno?

You probably know the King of Pop?

You probably know the King of Hearts? (Alice in Wonderland)

You probably don't know the King of Swing. (Benny Goodman)

You probably need to know the King of Assembly.

<https://lifeandtimes.games/episodes/files/28>

<https://www.eurogamer.net/articles/2016-03-03-a-big-interview-with-chris-sawyer-the-creator-of-rollercoaster-tycoon>

How many different roller coasters have you ridden?