

How does a C compiler allocate space for static global data and function definitions with parameters on storage (memory)? Where will it be allocated to, the stack or the heap (or both)?

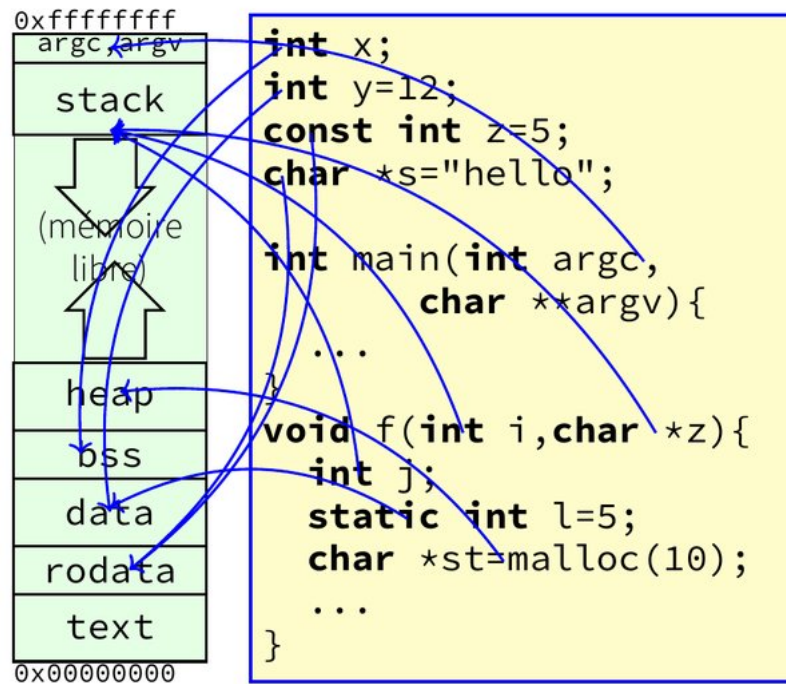


Alain Merigot

Updated Feb 2

Allocation for static data will be performed by the link editor, neither in the stack or on the heap, but on specific memory locations.

Actually the memory available for a process is divided in segments.



This represents a simple C programs on the right and the memory segments on the left.

This segments are (bottom to top)

- Text, that holds the program instructions
- the data segments: rodata, data and bss. They are all used for the static data (global variables), with the following differences: rodata is for read-only data (const, strings, etc). Any attempt to write anything will lead to a bus error. Data is for normal initialized data. Bss is for uninitialized (implicitly zeroed) data. While initialized data must be explicitly in the exe file, bss do not as the runtime fills bss by zeroes before launching the main. All these segments are managed by the linker (see below)
- Heap is for dynamically allocated data. It will grow upwards if extra space is required.
- Stack is for automatic variables, parameters, etc in functions. It grows downwards as required (but the total stack segment generally has a limited size).

When the compiler generate code in the object file, it write information on the static variables (what is their type, name, default value, where are they referenced).







When all the object files are compiled, they are fed to the linker. The linker collects all the information on the static variables and will choose a position for them in one of the data segments. It then patches the code generated by the compiler so that it can access these data with their actual address (that is only known after the link edition). Last, the link editor collects all the information for the different segments and write to the executable file.



To run the program, the launcher, will read the executable header, get the size of the different segments, ask the OS to create a process and assign it some memory space.

Messages




Quora








instruction of the program.


4.8K views · View 34 upvotes · View 3 shares

 34





 3


 Add a comment...


Add comm

View collapsed comments


View question >


About the Author


 **Alain Merigot**


 5.3K content views

5.3K this month


 Joined March 2017

 Following · 5

 [Notify me](#)


 Ask question

Related Spaces

 **C Programmers**


43.6K followers · 9 posts today

A space for C programmers && everything about the C Programming Language!

 **Programming & Programmers**

1.2K followers · 33 posts in the last week

Information on programming and programmers

 **PythonProgrammerGirls**

3.2K followers · 30 posts in the last week

programming, python, pythonprogramming, pythonprogrammergirl

View more Spaces >

More answers from Alain Merigot

View more >

Can two processes running on separate CPUs access the same memory locations (i.e., shared memory)? If so, how does that work?

6 views

What is the purpose of pointers when dealing with dynamic memory allocation in the C programming language?

22 views

Why does a processor have a limited amount of memory space, when it can access more random-access memory (RAM) than that?

3 views

What is the difference between ROM and RAM? Which type of memory do you think will be more useful in the future?

4 views

What is the difference between the write-through memory cache and the copy back memory cache?


4 views


No messages

Connect with others on Quora by beginning a new conversation.

New message

Messages





https://www.quora.com/How-does-a-C-compiler-allocate-space-for-static-global-data-and-function-definitions-with-parameters-on-storage-memory... 2/2