C++ (programming language) C (programming language) Programming Languages Computer Programming

Does the address of a pointer lie in a stack or heap (in C++)?

6 Answers

Federico Mengozzi, studied Computer Science at University of California, Davis
Answered Oct 5 2017 · Author has **78** answers and **47.8k** answer views

Pointer are variables, they lie in the stack.

You can use pointer if you want to access elements on the heap, in order to do that you declare a variable of type pointer that you can use to store the address, in the heap, of such element.

```
1  int *p = new int;
```

After this instruction you have reserved a int-size portion of memory in the heap and its address it's stored in the variable pointer, that variable like every other non-dynamic variables is store in the stack.

```
1  int **p = new int*;
```

In this case you reserve a portion of memory for a pointer to int in the heap, but the pointer that can reference...(more)

Related Questions More Answers Below

- Where do pointers point to the stack or heap in C/C++?
- Is C++ valarray as effective as plain heap memory operation with a pointer?
- What gets written in the stack and what gets written in the heap in C++?
- Are classes and structures in C++ on the stack or in the heap?
- Where does the object reside in C++, stack or heap?

Vivek Nagarajan, Programmer for 25 years
Answered Oct 5, 2017 · Upvoted by Richard Conto, Linux, FreeBSD, Solaris, Mac OS/X, Ultrix developer · Author has **5.4k** answers and **10.2m** answer views

If you declare a pointer on the stack:

```
1  int *i = NULL;
```

i resides on the stack...

If you declare it on the heap:

```
1  int **pi = new int*;
```

***pi** resides on the heap

Notice that you can refer to the stack allocated variable by name, but the heap allocated variable only via indirection - unless you create an alias reference or else if you have a pointer member variable in an object that was allocated on the heap.

What **i** and ***pi** point to is once again arbitrary - they can store heap, stack or invalid addresses.

Related Questions

- Where do pointers point to the stack or heap in C/C++?
- Is C++ valarray as effective as plain heap memory operation with a pointer?
- What gets written in the stack and what gets written in the heap in C++?
- Are classes and structures in C++ on the stack or in the heap?
- Where does the object reside in C++, stack or heap?
- Can I access heap memory without a pointer in C++?
- How is heap memory allocated in c++?
- What how do you allocate memory for a pointer to a pointer on stack, and how's on heap?
- Does the heap in C++ contain classes and structure objects or does it just contain pointers?
- What is the difference between stack and heap memory in C++? What type of variables are stored in heap and can heap ever overflow?

BTW it is not ideal to refer to these as “stack” and “heap” - these are implementation details - th...(more)

Eduard - Gabriel Munteanu, fluent in Haskell, bash, C, asm and knows something about PL theory

Answered Oct 5, 2017 · Author has **1.5k** answers and **1.4m** answer views

It’s not entirely clear to me what you’re asking. We can distinguish:

- the address *of* a pointer, which is a pointer to a pointer
- the address *in* a pointer
- the value pointed to by the pointer

These are all different kinds of *values*, which may be stored anywhere or might not be stored at all. Values may be formally held in variables, although that doesn’t necessarily mean they’re stored anywhere.

Automatic variables in a local scope (i.e. “normal” variables in functions) are usually held on the stack *if* the variable isn’t eliminated by the compiler or assigned to a register. A variable may be a poin...(more)

Related Questions

Can I access heap memory without a pointer in C++?

How is heap memory allocated in c++?

What how do you allocate memory for a pointer to a pointer on stack, and how’s on heap?

Does the heap in C++ contain classes and structure objects or does it just contain pointers?

What is the difference between stack and heap memory in C++? What type of variables are stored in heap and can heap ever overflow?

Where are the arrays in C stored, in a stack or a heap?

Is the "heap" a library that abstracts objects assigned by a pointer in C/C++?

Why doesn't class pointer need heap allocation in C++?

What's the meaning of stack and heap in C programming?

What is the necessity for a C/C++ compiler to maintain stack memory variables? Why not manipulate all variables on heap with smart pointers?

Is reference in C++ a pointer?

How are the stack and heap divided in memory address-wise in C?

What is the use of a shared pointer in C++?

What is the function of stack and heap in programming?

Does C# use a stack or heap for memory allocation?