

## Re: Best way to store postfix data?

---

*Source:* <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2007-07/msg00373.html>

---

- *From:* Scoots <bssalmons@xxxxxxxxxxxxxx>
  - *Date:* Wed, 04 Jul 2007 06:29:30 -0700
- 

On Jul 2, 10:57 am, Joseph M. Newcomer <newco...@xxxxxxxxxxxxxx> wrote:

Look into based pointers. You can create a structure where you keep the symbol table as a based-pointer table (relative offsets) and the executable tree as a based pointer system, and consequently you can write the data out directly. Even if the data is not originally contiguous, you can write it out maintaining the relative pointer positions and therefore you can perfectly well save the relative pointers to the symbol table, then read it back in. Based pointers allow the pain-free conversion from relative pointers to absolute pointers.

You can also use based pointers to address an array of proxies to the actual symbol values.  
joe

On Mon, 02 Jul 2007 06:35:03 -0700, Scoots <bssalm...@xxxxxxxxxxxxxx> wrote:

I agree with you. In fact, I have a tree based logic that is quite fast (can read, parse, error check, and execute complex statements with control logic in the same time as the old read one line, execute one line (only simplest statements) engine.

But unfortunately, my boss wants me to try this approach and then compare them. I haven't optimized my tree approach (at the moment, it is a memory hog. 20,000 lines of code was spiking it from ~15mbs usage to ~60 mbs usage. There's some obvious things I can remove to drop that to ~40mbs, but there's some memory hog I've missed in there. Probably my method for building my assign logic. It's fast, but it's creating objects for each element of an assign statement. Very fast, but inefficient. As for any memory fragmentation, I could use CFixedAlloc at the cost of slightly higher memory cost requirements, but I'm willing to try other solutions.

## Re: Best way to store postfix data?

The reason for the two maps is fairly simple. As I'm building the logic, it CAN grow. As of right now, my 'instructions' are the same size if I can move the logic and assignments to virtual 'heap' space. To me, that seems infinitely preferable to trying to store them inline and recording the size of the block. Maybe not. Storing references to variables is no longer an issue, as we're going to be saving the files so that if the saved preprocessed version and the file share the same timestamp, we just load the preprocessed. This means I can't store the variable locations.

I was storing the assignment logic as a string, not the variables. Those are already in the engine. Instead, I'm trying to come up with the most efficient way of taking these instructions in, and storing them. Since I allow control routines, I have to execute at runtime, and hence, have to build a logic for the program. Post-fix notation seems like it would help a lot here, as I have both single argument functions and double argument functions. Since I can't guarantee that the variable addresses will stay the same between runs, I don't think I can store them (also, erm, they aren't simple variables. I can't create a simple reference to it. It's an inherited problem from code written 15 years ago.).

In short, I'm not trying to store the symbol table as strings, just my assignment statements. Variables are stored entirely separate from the engine, and I can not change that.

An efficient way of doing this interpretation is just stumping me. I definately appreciate your help, and I'll be happy to answer any questions you have or try anything you suggest. Unfortunately, I'm only an intern, with less than a year of c++ experience (though I've been coding for much longer than that), so I might come up with some truly monumentally odd ways of solving things. I'm not opposed to learning new things. Thanks for the help.

Joseph M. Newcomer [MVP]  
email: newco...@xxxxxxxxxxxxx

Web:<http://www.flounder.com>

MVP Tips:[http://www.flounder.com/mvp\\_tips.htm](http://www.flounder.com/mvp_tips.htm)– Hide quoted text –

– Show quoted text –

Thanks, that was actually really helpful! I hadn't seen those before...

Re: Best way to store postfix data?

Re: Best way to store postfix data?