Subject: U ++ Vector vs STL vector
Posted by 281264 on Mon, 12 Jul 2010 20:02:41 GMT
View Forum Message <> Reply to Message

Hi,

I am interested in knowing the differences, advantages and disadvantages of the Vector and

I will be grateful if someone explains me theses topics.

Thanks and best wishes.

Javier

Subject: Re: U ++ Vector vs STL vector
Posted by Novo on Mon, 12 Jul 2010 22:13:18 GMT
View Forum Message <> Reply to Message

memory. You do not need to call a constructor and a destructor in a row. Such a situation happens very often when you want to resize a vector, which stores data in a continuous piece of memory. You just need to move old data into a newly allocated memory. If your data is "moveable" you can just call the memcpy() function instead of calling a copy constructor in a loop.

Hope this helps.

Subject: Re: U ++ Vector vs STL vector
Posted by mr_ped on Tue, 13 Jul 2010 06:49:08 GMT
View Forum Message <> Reply to Message

deep copy means the Vector data are really copied into another Vector. When you do a = b; with Vectors, b is just picked up and moved to a, leaving b "empty" in picked state. If you want true copy without picking b, you have to use deep copy. (a <<= b; is the short form I think?)

Subject: Re: U ++ Vector vs STL vector
Posted by dolik.rce on Tue, 13 Jul 2010 08:16:37 GMT
View Forum Message <> Reply to Message

The main difference between Vector and vector IMHO is the speed Since for Vector a=b just moves (pickes) the content, instead of copying it into the new container, it is way faster. In real life, most of the operations with containers doesn't care about the content of b after the operation.

Sometimes it requires the programmer to think bit more about what he's doing, but the performance gain really pays off (See comparisons). I think I could count on my fingers how many times I had to use the deep copy operator <<=.

Honza

Subject: Re: U ++ Vector vs STL vector
Posted by mrjt on Tue, 13 Jul 2010 11:08:48 GMT
View Forum Message <> Reply to Message

Another great feature is that picking be default allows you to use Vector/Array/other containers as a return type from functions. This greatly reduces the reliance on passing parameters by reference for complicated output.
