Subject: T* Detach() for ArrayMap
Posted by kohait00 on Tue, 02 Feb 2010 15:31:20 GMT
View Forum Message <> Reply to Message

hi there

i ran into the need of

T* ArrayMap<K, T>::Detach(const K & key)

function. normal Array<T> *does* have one
T* Array<T>::Detach(int i);

since one can do a

T & Add(const K & key, T * newt)

on both, Array<T> and ArrayMap<K, T>, it would be consistent to have a Detach() function on both cotainer types.

so maybe the following lines could be added to Map.h

T* Detach(int i) { T *t = &B::value[i]; B::key.Remove(i); return t; }

Subject: Re: T* Detach() for ArrayMap
Posted by mirek on Tue, 02 Feb 2010 21:52:37 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Tue, 02 February 2010 10:31hi there

i ran into the need of

T* ArrayMap<K, T>::Detach(const K & key)

function. normal Array<T> *does* have one
T* Array<T>::Detach(int i);

since one can do a

T & Add(const K & key, T * newt)

on both, Array<T> and ArrayMap<K, T>, it would be consistent to have a Detach() function on both cotainer types.

so maybe the following lines could be added to Map.h

T* Detach(int i) { T *t = &B::value[i]; B::key.Remove(i); return t; }

Hm, is that correct implementation?

I would say something like

T* Detach(int i) { B::key.Remove(i); return B::value.Detach(i); }

is the correct one?

Another problem there is that this kind of detach is quite slow. Maybe we need DetachUnlink too (?).

Mirek

Subject: Re: T* Detach() for ArrayMap
Posted by kohait00 on Wed, 03 Feb 2010 05:57:53 GMT
View Forum Message <> Reply to Message

you're absolutely right, this one's better. should have looked before..

the goal was just to be able to switch from Array to ArrayMap, when i.e one gets the impression, well i need to speed up things, or need to change stuff, so just to enhance it in one place, and supply the keys in the other places, without the need of doing lot of workarounds to replace missing behaviour.

Detach() slowness was not the main goal in that, though having it done faster is precious. one should at least be able to separate an obejct from a ArrayMap again (which maybe were constructed outside the map and added).

this kind of aproach is not possible for Vector and VectorMap though, the objects are stored on a consequtive mem space, which can only be Removed(). but Vector/VectorMap requires to have a Mouvable object (objects are reorganized on storage space during growth/shrink of vector), so why not having same Detach() also here? which would simply copy the stored object to a decent new place and return pointer? maybe because its no real "detach".. but it would make life esier, when exchanging vector to array, array to map or waht ever. dont know for sure if it makes sense.

DetachUnlink also makes sense, to say simply "stay in background until reactivated.."... but how would you reactivate such one?

Subject: Re: T* Detach() for ArrayMap
Posted by mirek on Wed, 03 Feb 2010 10:24:37 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Wed, 03 February 2010 00:57you're absolutely right, this one's better. should have looked before..

the goal was just to be able to switch from Array to ArrayMap, when i.e one gets the impression, well i need to speed up things, or need to change stuff, so just to enhance it in one place, and supply the keys in the other places, without the need of doing lot of workarounds to replace missing behaviour.

Detach() slowness was not the main goal in that, though having it done faster is precious. one should at least be able to separate an obejct from a ArrayMap again (which maybe were constructed outside the map and added).

this kind of aproach is not possible for Vector and VectorMap though, the objects are stored on a consequtive mem space, which can only be Removed(). but Vector/VectorMap requires to have a Mouvable object (objects are reorganized on storage space during growth/shrink of vector), so why not having same Detach() also here? which would simply copy the stored object to a decent new place and return pointer?

It is not quite typical situation. If you really need something like that, you can use Swap, which leads to similar results...

Quote:
DetachUnlink also makes sense, to say simply "stay in background until reactivated.."... but how would you reactivate such one?

Good point

Mirek

Subject: Re: T* Detach() for ArrayMap
Posted by mirek on Wed, 03 Feb 2010 10:26:37 GMT
View Forum Message <> Reply to Message

Detach added.

Mirek