Subject: valgrind problem, heap-leak debugging
Posted by slashupp on Fri, 13 Feb 2015 07:05:45 GMT
View Forum Message <> Reply to Message

Running my app ends with "Heap-leaks detected".

I don't do "new" anywhere, so there is no obvious place to start looking for the problem.

So I try [Debug/Test with Valgrind] and get a huge amount of output.
This is not helpful.

1. Is there a way I can force valgrind to just focus on my app and ignore upp-sources?

2. What other ways can I use to find the cause of the heap-leak?

PS: .. to maybe help others with same kind of problem ..
I've found the cause: needed a virtual dtor in a struct I was using, added it and the heap-leaks went away.
How I found it: Searched SO and found this answer [http://stackoverflow.com/a/2480641/15161]:
Quote:[..] code that creates your object on the heap. It would be bad practice for your class to assume that it is always going to be created on the stack...
My struct is defined within a class and instances are created, used and passed as parameters all over the place (without the SO-clue (and only with this dysfunctional valgrind-hook) it would have
been a real mission to find)

The two questions I think remains valid and any answers would be helpful for the future.

Subject: Re: valgrind problem, heap-leak debugging
Posted by mirek on Fri, 13 Feb 2015 09:09:05 GMT
View Forum Message <> Reply to Message

[quote title=slashupp wrote on Fri, 13 February 2015 08:05]Running my app ends with "Heap-leaks detected".

I don't do "new" anywhere, so there is no obvious place to start looking for the problem.

So I try [Debug/Test with Valgrind] and get a huge amount of output.
This is not helpful.

1. Is there a way I can force valgrind to just focus on my app and ignore upp-sources?

2. What other ways can I use to find the cause of the heap-leak?

PS: .. to maybe help others with same kind of problem ..
I've found the cause: needed a virtual dtor in a struct I was using, added it and the heap-leaks went away.
[/code]

Yeah, with U++, memory leaks are caused by missing virtual destructor in 95% of cases. Happens to me too.

BTW, U++ has a quite good way to resolve memory leaks:

http://www.ultimatepp.org/srcdoc$Core$Leaks$en-us.html
