

Bypassing Memory Leak in Modern C++ Realm

Dorottya Papp, Norbert Pataki

Dept. of Programming Languages and Compilers,
Fac. of Informatics, Eötvös Loránd University, Budapest
dorottypapp@yahoo.com, patakino@elte.hu

Abstract

Deallocation of dynamically allocated memory belongs to the responsibility of programmers in the C and C++ programming languages. However, compilers do not support the work of the programmers with error or warning diagnostics. Thus the result of this behaviour can be memory leak. Programs' memory consumption may be unreasonably big and even the operating system can be too slow because of the swapping.

We present some different scenarios when memory leak is occurred. We show the root cause of the scenarios. This paper presents existing tools for detecting or avoiding memory leak. These tools work in different ways. We analyze the smart pointers of C++11 standard, valgrind that is a run-time heap profiler, Hans Boehm's garbage collector and the Clang static analyzer. We present the pros and cons of the tools. We analyze how difficult to use these tools, how the efficiency is affected and how these tools can be enhanced for overcome unwanted memory leak. We present our proposals to make the tools more effective.

Keywords: C++, smart pointers, memory leak, garbage collection

MSC: 68N15 Programming languages