PARASOFT

Parasoft Insure++

THE ULTIMATE MEMORY DEBUGGER FOR C & C++

Parasoft Insure++

ELIMINATE DEFECTS AND MEMORY ISSUES

- Find memory errors before they become runtime problems.
- Find common errors during 64-bit porting.
- Optimize memory usage of applications.
- Reduce development and support cost.

TRY PARASOFT INSURE++

Schedule a demo to learn how Parasoft Insure++ can find and help your team handle memory management issues.

FIND REAL BUGS WITH RUNTIME MEMORY ANALYSIS & ERROR DETECTION

Errors such as memory corruption, memory leaks, access outside of array bounds, invalid pointers, and the like, often go undetected during normal testing, only to result in application crashes or security exploits in the field. Insure++ will help you find and eliminate defects quickly and easily.

Insure++ is a runtime memory analysis and error detection tool for C and C++ that automatically identifies a variety of difficult-to-find programming, memory access, and security errors, along with potential defects and inefficiencies in memory usage.

During testing, Insure++ checks all types of memory references, including those to static (global), stack, and shared memory — both in the user's code and in third-party libraries.

INSURE ++ MODES OF USE

Insure++'s memory analysis capabilities in are based on patented source instrumentation algorithms. Source code instrumentation enables Insure++ to detect more error types than other memory error detection technologies, and also provides complete information indicating the root causes of the errors found, using a full database of program elements and memory structures. There are two ways to use Insure++ for memory analysis and error detection:

SOURCE INSTRUMENTATION MODE

The first and most detailed analysis is achieved with full source code instrumentation. This requires that application sources be compiled and linked with Insure++, which generates its own instrumented files that are passed to the actual compiler.

LINK MODE

Without source code instrumentation, by linking your application object code and libraries with Insure++, the tool can "spy" on the kernel/application program interface to detect errors such as leaks, bad memory references, standard API usage errors, and so on.

"Problems that used to take us several days or even weeks to track down, we are able to routinely find and fix in a few hours with Insure++."

—Valued Customer

MEMORY DEBUGGING WITH PARASOFT INSURE++

ParaTarget Insure++'s patented instrumentation to pinpoint real memory issues and get immediate visibility at runtime. Identify memory issues both within your codebase and caused by external libraries, automatically tracking and monitoring all threads and processes within the application to quickly find algorithmic anomalies.

At compile time, use Insure++ to identify deviations from C/C++ standards that may lead to memory leaks or application instability.

RUNTIME MEMORY DEBUGGING

Errors detected include:

» Corrupted heap and stack memory

» Use of uninitialized variables and objects

» Array and string bounds errors on heap and stack

» Use of dangling, NULL, and uninitialized pointers

» All types of memory allocation and free errors or mismatches

» All types of memory leaks

» Type mismatches in global declarations, pointers, and function calls

COMPILE TIME ANALYSIS

Errors detected include:

» Cast of pointer loses precision

» Mismatch in format specification

» Mismatch in argument type

» Code is not evaluated, has no effect, or is unreachable

» Undefined identifier

» Variable declared, but never used

» Returning pointer to local variable

» Function returns inconsistent value

» Unused variables

SUPPORTED PLATFORMS

MICROSOFT WINDOWS

» Visual Studio 2012, 2013, 2015, and 2017

» Windows 7, 8, 8.1, 10

LINUX

» GNU GCC 5.4.x - 8.x

» Ubuntu 14, 16 & 17

» Fedora 18, 24, 25, & 26

» CentOS 7

TOTAL COVERAGE ANALYSIS (PARASOFT TCA®)

TCA analyzes and reports block code coverage and lets you get under the hood of your program to see which parts are actually testing and how often each block is executed.

» Identify the blocks of instrumented code exercised during memory debugging.

» Understand the relationships between memory defects and the code that was executed.

» See how many times a block of code has been executed to understand the impact of a memory defect on the overall stability of the application.

DYNAMIC MEMORY VISUALIZATION (PARASOFT INUSE®)

Insure++ visualizes how an application uses memory, providing a graphical view of all memory allocations over time with specific visibility into overall heap usage, block allocations, possible outstanding leaks, and so on.

While your application is running, get real-time visibility into:

» How your application is utilizing memory.

» Memory problems as they appear, and the impact of those problems on memory.

» Details of heap usage, block allocations, free memory, and frequency of memory access over time.

» Comparisons with historical data to view the outcomes of tuning your application's use of memory.