

Debugging Dynamic Memory in Multi-threaded Applications: Squash The Bugs Before They Bite You

North Haven, Conn, November 2, 2010 – Dynamic memory bugs are mistakes in the management of run-time allocated memory. They hide in your code and when they bite, they may slow down or even crash processes. Applications that are required to run 24x7 often must be restarted with a corresponding service outage. Traditional code debuggers won't find them but Leak Check and Memory Minder from Dynamic Memory Solutions can help you exterminate them from your software quickly and easily.

In modern C/C++ Unix environments, many if not most applications are multi-threaded to take advantage of modern multi-processor and multi-core system architectures. Multi-threading applications allows the developer to take advantage of these hardware capabilities to execute faster and provide greater responsiveness than single-threaded programs. These applications, however, are much more prone to dynamic memory bugs. Tens, hundreds or even thousands of threads may allocate dynamic memory from a common heap. Often, these allocated memory blocks are passed between threads and if even one thread fails to return a block when its usefulness has ended, run-time memory is leaked and the application, and possibly the entire system, is one step closer to death. Sometimes, unexpected execution paths may have more than one thread free the same allocated memory block. These critical errors may even crash the application. Any dynamic memory bugs in an application result in lower productivity when the appropriate debug tools are not available, poor customer satisfaction when the software has already been deployed and late night calls when the system fails.

Memory Debugging with Leak Check

When application code is written, reused or ported to a different system architecture, finding and debugging disruptive memory bugs can become a challenging task. Memory bugs can lurk in a code base for long periods of time and can become a serious issue when least expected.

Memory is leaked when a program allocates a block of memory, uses it, but then discards all references to the block without returning it for reuse. This results in programs not being able to use the leaked memory for a new purpose, creating the need to acquire additional memory from the operating system. Increased use of system memory not only impacts the program containing the error but may also impact all other applications. Critical memory errors, another species of memory bug, occur when poor software architecture or implementation allocates memory blocks and then does not manage them correctly.

The advanced memory debugging technology found in Leak Check and Memory Minder from Dynamic Memory Solutions is able to detect memory leaks by determining whether the program retains a reference to each block of allocated memory. Leaks can be detected when the program exits or at any point during program execution allowing maximum flexibility during the testing and debugging process. Additionally, Leak Check monitors the allocation and return of memory blocks and identifies and reports any critical memory

management errors.

Understanding Memory Usage With Memory Minder

In multi-threaded application, when tens, hundreds or thousands of threads are allocating memory, it can be difficult to know exactly how much memory is being used at a point in time or the maximum amount during program execution. Memory Minder tracks memory allocation during program execution, and identifies the memory usage of each function in the code. As a result, developers can determine which sections of code and which execution paths put the greatest stress on system memory, enabling them to focus their attention there if memory usage is a problem.

The Leak Check and Memory Minder Advantage

Leak Check and Memory Minder utilize advanced technology to make them the easiest to use memory debugging and analysis tools for developers. They require no recompilation of your code and have only a modest impact on the run-time performance of applications. The interface guides the user through the task of memory debugging and provides easy-to-understand graphical displays, powerful analysis tools and features to support collaboration. This makes it easy to identify and report a lurking memory bug to the colleague, library vendor or consultant responsible for the code in question.

Leak Check and Memory Minder from Dynamic Memory Solutions also feature the ability to compare memory statistics, examine heap status and detect memory management errors making them uniquely well suited for debugging multi-threaded applications.

About Dynamic Memory Solutions

Dynamic Memory Solutions, a leading developer of C/C++ software testing tools, leverages software technology to automate the laborious burden of software defect detection, debugging and testing. The result is vastly improved software quality with a reduction in labor costs and development schedules. Many users break-even in terms of ROI in just days of use. All of Dynamic Memory Solutions' software testing and debugging tools come with a free trial for prospective customers. For more information, please visit <http://www.dynamic-memory.com> or call (877) 293-4144.