

LynxInsure++

Maximize quality and performance of applications and testing processes



LynxInsure++ is a suite of integrated run-time analysis and visualization tools designed to maximize quality and performance and reduce the cost of developing and testing LynuxWorks™ LynxOS® applications. Each of the three auto-instrumenting products in the suite—LynxInsure++, Total Coverage Analysis (TCA) and Inuse—provide a unique value in a particular area of quality improvement. Used together, they enable developers to rapidly define a comprehensive quality improvement process.

More importantly, the LynxInsure++ suite requires no changes to source code and causes no disruption of normal system operations.

Driving product quality

The LynxInsure++ suite provides developers with detailed information to enable a deeper understanding of the operation of an application. This knowledge can then be applied to improving product quality while reducing long- and short-term development and maintenance costs. The detailed information delivered by the LynxInsure++ suite covers four key areas of software development:

- Memory reference validation
- Programming verification
- Optimization
- Coverage analysis

Within the suite, the LynxInsure++ runtime error detection module verifies memory references and program implementation—including the proper use of the API and proper programming practices such as valid pointer operations

and more. Thread-aware and thread-safe, LynxInsure++ detects a wide range of errors in C/C++ applications and threads. Unlike other automated detection tools, LynxInsure++ easily handles the non-linear nature of threaded applications and is particularly effective in finding errors in allocating and freeing dynamic memory. It supports all popular compilers, including cc, gcc and acc, and it is capable of checking third-party libraries and functions, as well as interfaces to modules written in languages other than C.

The TCA coverage analysis module acts as a display summary tool for LynxInsure++, providing displays of code execution and coverage on a line-by-line or a logical block basis.

Finally, the graphic Inuse dynamic memory visualization module provides animated information on memory allocation, including:

- Amount of memory in use
- Memory fragmentation
- Sizes of allocated blocks
- Number of calls to memory management functions

Driving product performance

Performance analysis can be a daunting task when designing sophisticated applications. The LynxInsure++ suite optimizes product performance by providing visibility into the time-varied interactions between different system components.

The LynxInsure++ module possesses built-in detailed knowledge about correct and incorrect

LynxInsure++ Advantages

- Enhanced product quality —detailed information drives improved product quality and reduced development and maintenance costs
- Optimized product performance — visibility into memory access/code execution interaction patterns
- Enhanced testing effectiveness — enables iterative improvement of existing test procedures

programming practices and the correct means of utilizing APIs. For example, incorrect uses of programming abstractions such as pointers are clearly displayed, as are incidents of memory corruption. This knowledge can be leveraged to reveal and correct conditions that could potentially cause normally efficient execution threads to de-synchronize during error handling.

The Inuse module, meanwhile, automatically instruments applications to log and reveal memory allocation patterns. As a result bottlenecks to application efficiency are easily determined.

At the same time, the TCA module's display of program execution provides developers with a concrete view of areas that run and their frequency.

Driving more effective testing

Generally, it is up to developers to write effective tests for their applications. But ensuring

that the tests are testing what they should is not always easy. The LynxInsure++ suite provides valuable insights into the effectiveness of application testing processes by correlating testing procedures to actual program coverage and performance.

Integration in existing test harnesses is fast and easy, due to the auto-instrumenting nature of the LynxInsure++ suite. This makes it simple to enhance the effectiveness of existing test suites through a process of iterative improvement—i.e.: run a test, view its effectiveness, improve the test, run it again, etc.—designed to optimize both application quality and performance.

Detecting elusive errors

LynxInsure++ is relentless in detecting elusive errors, including:

- Memory corruption
- Memory leaks
- Memory allocation errors
- Variable initialization errors
- Variable definition errors
- Pointer errors
- Library errors
- Logic errors
- Algorithmic errors

LynxInsure++ not only pinpoints the sources of memory leaks, but also identifies the line in the code where they occurred. When tracking down memory reference errors, LynxInsure++ avoids the statistical approach in favor of checking each memory reference (static, stack and dynamically allocated) for validity when that reference is executed.

1.800.255.5969



LynuxWorks, Inc.
855 Branham Lane East
San Jose, CA 95138-1018
408-979-3900
408-979-3920 fax
www.lynxworks.com

LynuxWorks Europe
2 Allee de la Fresnerie
78330 Fontenay Le Fleury
France
+33 1 30 85 06 00
+33 1 30 85 06 06 fax