

SpyKer Pro v2.0

Accelerate time-to-market and improve code quality and performance with LynuxWorks' dynamically instrumented trace analyzer



Improving embedded system performance

It takes detailed information to understand how an embedded system is performing. LynuxWorks' SpyKer™ Pro, the first dynamically instrumented system trace analyzer, provides comprehensive real-time information on the interaction between tasks, processor execution patterns and timing of system activities so developers can:

- Understand what is executing in a system over time
- Track down elusive application bugs
- Fine-tune embedded system performance

Fast, easy and non-intrusive

While traditional performance analysis tools require pre-instrumenting an application with special calls, or running it under a specially instrumented kernel, SpyKer Pro enables ordinary kernel and active applications to be auto-instrumented at runtime. This is accomplished by dynamically patching a LynxOS or

BlueCat Linux kernel and library binaries with logging calls. Because there's no need for lengthy instrumentation cycles and kernel reboots, debugging proceeds faster than ever and overall development costs are reduced. What's more, because SpyKer Pro gathers information on the actual kernel that is deployed and used, configuration-management issues between the instrumented kernel and deployed kernel are non-existent.

In addition, the overhead of a SpyKer trace patch is exceedingly low, which minimizes its impact on the target system. This results in truly objective measurements and eliminates possible timing-invasiveness problems that could stand in the way of finding particularly elusive bugs.

Instrument both local and remote targets

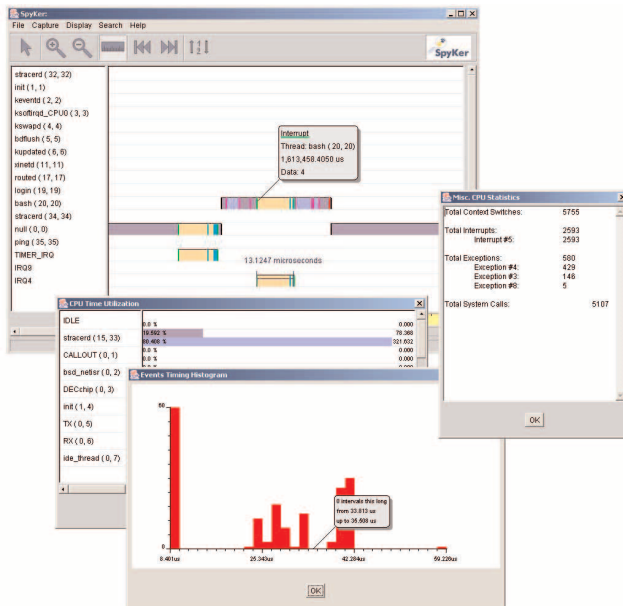
Written in Java, SpyKer Pro's intuitive front-end GUI runs on any PC or workstation with a Java Virtual Machine. This allows the front end to reside locally or remotely from the tar-

SpyKer Pro Benefits

- Decreased development costs
- Accelerated development and debugging
- Enhanced product quality and reliability

SpyKer Pro Features

- **Auto-instrumented tracing**—no lengthy instrumentation cycles or kernel reboots
- **Low overhead**—minimal patch trace overhead
- **Accelerated development and debugging**—easy, anytime visibility into program execution
- **Rich visualization**—intuitive easy-to-use GUI with multiple concurrent windows
- **Targeting flexibility**—measure systems locally or remotely
- **Extensive tracing capabilities**—system calls, interrupts, context switches, processor exceptions, and more
- **Dynamic tracing of library calls**—automatically instruments dynamically linked libraries
- **CPU statistics**—graphically depicts system memory use, CPU time use, context switches, and more
- **Event histograms**—enables fast isolation of timing areas in question
- **Selective tracing**—minimizes impact on a target system



get system, communicating via TCP/IP. SpyKer Pro's remote capability, combined with its innate non-intrusiveness, enables an auto-instrumentation module to be deployed in target systems for later use, either when searching for bugs or evaluating performance data during "real-world" sessions.

Easy event capture

A pull-down menu starts the capture process, enabling users to define events to be traced and time-stamped. Users quickly set up start and/or trigger events and remotely configure the data collection buffers on the host system (e.g. saved to disk, network drive or ring buffers). To minimize the influence of trace collection, only user-selected events are dynamically traced. Once the data are collected, the target is returned to its original state and all dynamic instrumentation points are removed.

Powerful visualization

SpyKer Pro complements its powerful dynamic instrumenting facility with equally powerful front-end visualization capabilities. Developers gather, display and interact with captured data, and can quickly comprehend normally difficult-to-understand relationships within the application.

For example, SpyKer Pro's front-end GUI can display event data across multiple windows concurrently, each with different visualization—ranging from displaying a full trace to displaying multiple in-depth looks. Collected data—as well as SpyKer Pro commands—are displayed in an uncluttered visual fashion, allowing developers to easily:

- Zoom in or out on areas of interest in event data
- Invoke "pop-up" boxes that display details about cursor-selected events

- Filter events and skip specific system calls or other events in order to filter out unwanted data
- Visualize and accurately measure times between different events

Event points are depicted as colored bars that can be mouse-selected to present expanded data, while lines of process or thread execution are color-coded to differentiate between user-mode processes and system-mode processes.

Detailed level of understanding

Ordinarily, program operations are assessed indirectly by running functional tests. SpyKer Pro, on the other hand, enables developers to gain a more detailed level of understanding through the observation of low-level systems operations. All events displayed are aspects of the processes that are running, and detailed information about the processes themselves is also displayed. For example:

- Developers can trace an extensive range of event types including system calls, interrupts, context switches, processor exceptions and many other system events
- Trace points can be statically and dynamically inserted into library calls, enabling users to visualize library routines by their familiar names

Additionally, CPU statistics are kept during a trace in order to help developers graphically visualize system memory utilization.

For rapid analysis of timing problems, an event histogram tool enables developers to easily visualize time spent in interrupt service routines (ISRs) or between any two event points in a system. The difference in time between the points is displayed in a histogram format and a single mouse click will take the user to the first occurrence of the selected data in the trace.

Flexible trace buffer management

SpyKer Pro presents several options for managing trace buffers, including saving trace buffers to disk, network mounted drive, or solely to the buffer itself. Developers can also save the buffers in non-volatile memory on the target system, if available. The data can be pulled from the non-volatile memory board should the system crash and reset to show valuable information on the last known execution of the system.

Dynamic new approach

SpyKer Pro's new dynamic approach to tracing is revolutionizing the way embedded developers collect and analyze event data. For the first time, developers have graphically rich, real-time visibility into the low-level operations of local or remote systems, in the lab or in the real world. SpyKer Pro removes the guesswork, labor performance "hits," and time delays from event tracing—and helps you move higher quality, more reliable products to market faster and less expensively than ever before.

SpyKer SE

LinuxWorks also offers an entry-level version of SpyKer: SpyKer SE. SpyKer SE provides the same dynamically instrumented trace-analyzer paradigm and intuitive GUI as SpyKer Pro, but with a subset of SpyKer Pro's more advanced tracing features. SpyKer SE provides a cost-effective way to familiarize a development team with the benefits of dynamically instrumented trace analysis and is well-suited to specific types of embedded projects that don't require extensive tracing functionality.

1.800.255.5969



LynuxWorks, Inc.

855 Embedded Way
San José, CA 95138-1018
408-979-3900
408-979-3920 fax
www.lynuxworks.com

LynuxWorks Europe

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