Runtime analysis tool C-RUN

Ensuring code quality through runtime analysis

With runtime analysis, you can find actual errors at runtime. Our add-on tool C-RUN is easy to use and has a wide range of features based on customer requests. C-RUN performs runtime analysis by checking application execution directly within the development environment. It checks for arithmetic issues, bounds issues and heap integrity and will tell you what went wrong and where.

Key features

- Analysis of C and C++ code
- Intuitive and easy-to-use settings
- Unique optimizations of test instrumentation minimizes code size overhead
- Comprehensive and detailed runtime error information
- Call stack information provided for each error found
- Code correlation and graphical feedback in editor
- Flexible error filter management
- Bounds checking to ensure accesses to arrays and other objects are within boundaries
- Buffer overflow detection
- Detection of value changes when casting between types
- Checks for overflow and wraparound in computations
- Discovery of bit losses in shift operations
- Heap and memory leaks checking
- Available as an add-on product for:
  - IAR Embedded Workbench for Arm, version 7.20 and forward
  - IAR Embedded Workbench for RX, version 3.10 and forward

Arithmetic issues, Bounds issues and Heap checking

Arithmetic issues includes overflow, wraparound, conversion errors, division by zero and missing default labels in switch statements. Such errors can be detected by inserting specific instrumentation code at all places where a potential error can happen. Source level instrumentation often insert an if-statement or equivalent that checks the condition and either print something to stdout or write a special value to a port to log the issue. Analogously, a compiler can insert instructions to check the condition and somehow report the issue at runtime. Code size will increase more or less linearly with the number of operations to check.

Bounds issues is a very broad category of problems that include typical out-of-bounds issues like writing or reading outside the defined bounds of an array. But the out-of-bounds concept can be generalized to deal with anything that is accessed through a pointer regardless of type or size. This includes things like pointers to scalar objects on the stack, so if you happen to change a pointer, or someone with malicious intent does it for you, to something on the stack, a state-of-the-art bounds checker can detect if the new value of the pointer is within the bounds of a valid object.

Heap checking assures that the heap retains its integrity and does not leak allocated blocks over time. Efficient heap checking is essentially an exercise in library implementation, but knowing the internals of the associated compiler can be beneficial if some functionality can be treated much the same way as other compiler intrinsic functions. Integrity checking is typically done on each call to malloc, free and their friends, both in the C and C++ world. Heap integrity checking can be a real drag on performance if the heap is big, since the checks can entail traversing the full heap, so a way to decide the frequency of checks can be crucial for some applications.
**Convenient runtime analysis and error checking with C-RUN**

C-RUN is an extension to IAR Embedded Workbench and is available for Arm and RX. C-RUN is designed to be a natural part of your development workflow, when working in a traditional edit/build/debug cycle, running unit tests or doing integration tests. C-RUN provides you with extremely valuable feedback already as soon as the first iteration of code is about to be taken for a test drive. Thanks its tight integration into IAR Embedded Workbench, C-RUN can be part of the daily work for any developer.

---

**IAR Embedded Workbench**

IAR Embedded Workbench is a complete C/C++ development toolchain for embedded applications. The toolchain offers leading code quality, outstanding optimizations for size and speed, as well as extensive debug functionality with a fully integrated debugger with simulator and hardware debugging support. C-RUN is fully integrated with the IAR Embedded Workbench IDE, which helps developers to ensure their code is safe and of high quality at an early stage, which also aids companies to shorten their time to market as impact of errors further down the line might be very time consuming and expensive.