

Runtime library

In computer programming, a **runtime library** is a set of low-level routines used by a compiler to invoke some of the behaviors of a runtime environment, by inserting calls to the runtime library into compiled executable binary. The runtime environment implements the execution model, built-in functions, and other fundamental behaviors of a programming language^[1]. During execution (run time) of that computer program, execution of those calls to the runtime library cause communication between the executable binary and the runtime environment. A runtime library often includes built-in functions for memory management or exception handling.^[2] Therefore, a runtime library is always specific to the platform and compiler.

The runtime library may implement a portion of the runtime environment's behavior, but if one reads the code of the calls available, they are typically only thin wrappers that simply package information, and send it to the runtime environment or operating system. However, sometimes the term *runtime library* is meant to include the code of the runtime environment itself, even though much of that code cannot be directly reached via a library call.

For example, some language features that can be performed only (or are more efficient or accurate) at runtime are implemented in the runtime environment and may be invoked via the runtime library API, e.g. some logic errors, array bounds checking, dynamic type checking, exception handling, and possibly debugging functionality. For this reason, some programming bugs are not discovered until the program is tested in a "live" environment with real data, despite sophisticated compile-time checking and testing performed during development.

As another example, a runtime library may contain code of built-in low-level operations too complicated for their inlining during compilation, such as implementations of arithmetic operations not directly supported by the targeted CPU, or various miscellaneous compiler-specific operations and directives.^{[2][3]}

The concept of a *runtime library* should not be confused with an ordinary program library like that created by an application programmer or delivered by a third party, nor with a dynamic library, meaning a program library linked at run time. For example, the C programming language requires only a minimal runtime library (commonly called crt0), but defines a large standard library (called C standard library) that has to be provided by each implementation.^[1]

References

1. "The C Runtime Initialization, crt0.o" (<http://www.embecosm.com/appnotes/ean9/html/ch05s02.html>). embecosm.com. 2010. Retrieved 2013-12-30.
2. "Internals of the GNU compilers, Chapter 4. The GCC low-level runtime library" (<https://gcc.gnu.org/onlinedocs/gccint/Libgcc.html>). *gnu.org*. Retrieved 2013-12-30.
3. "Other Built-in Functions Provided by GCC" (<https://gcc.gnu.org/onlinedocs/gcc/Other-Builtins.html#Other-Builtins>). *GCC Introduction*. gnu.org. Retrieved 2013-12-30.

External links

- What is the C runtime library? (<https://stackoverflow.com/questions/2766233/what-is-the-c-runtime-library>) (StackExchange)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Runtime_library&oldid=804597904"

This page was last edited on 10 October 2017, at 01:05.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.