# Multithreading In C

Multithreading (computer architecture)

*of event. This type of multithreading is known as block, cooperative or coarse-grained multithreading. The goal of multithreading hardware support is to*

In computer architecture, multithreading is the ability of a central processing unit (CPU) (or a single core in a multi-core processor) to provide multiple threads of execution.

Yield (multithreading)

*In computer science, yield is an action that occurs in a computer program during multithreading, of forcing a processor to relinquish control of the current*

In computer science, yield is an action that occurs in a computer program during multithreading, of forcing a processor to relinquish control of the current running thread, and sending it to the end of the running queue, of the same scheduling priority.

Thread (computing)

*concurrent execution. Multithreading can also be applied to one process to enable parallel execution on a multiprocessing system. Multithreading libraries tend*

In computer science, a thread of execution is the smallest sequence of programmed instructions that can be managed independently by a scheduler, which is typically a part of the operating system. In many cases, a thread is a component of a process.

The multiple threads of a given process may be executed concurrently (via multithreading capabilities), sharing resources such as memory, while different processes do not share these resources. In particular, the threads of a process share its executable code and the values of its dynamically allocated variables and non-thread-local global variables at any given time.

The implementation of threads and processes differs between operating systems.

Speculative multithreading

*(2005). "SableSpMT: A Software Framework for Analysing Speculative Multithreading in Java". Proceedings of the 6th ACM SIGPLAN-SIGSOFT workshop on Program*

Thread Level Speculation (TLS), also known as Speculative Multi-threading, or Speculative Parallelization, is a technique to speculatively execute a section of computer code that is anticipated to be executed later in parallel with the normal execution on a separate independent thread. Such a speculative thread may need to make assumptions about the values of input variables. If these prove to be invalid, then the portions of the speculative thread that rely on these input variables will need to be discarded and squashed. If the assumptions are correct the program can complete in a shorter time provided the thread was able to be scheduled efficiently.

C++11

*language, C++11 does make several additions to the core language. Areas of the core language that were significantly improved include multithreading support*

C++11 is a version of a joint technical standard, ISO/IEC 14882, by the International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), for the C++ programming language. C++11 replaced the prior version of the C++ standard, named C++03, and was later replaced by C++14. The name follows the tradition of naming language versions by the publication year of the specification, though it was formerly named C++0x because it was expected to be published before 2010.

Although one of the design goals was to prefer changes to the libraries over changes to the core language, C++11 does make several additions to the core language. Areas of the core language that were significantly improved include multithreading support, generic programming support, uniform initialization...

Boost (C++ libraries)

*the C++ programming language that provides support for tasks and structures such as linear algebra, pseudorandom number generation, multithreading, image*

Boost is a set of libraries for the C++ programming language that provides support for tasks and structures such as linear algebra, pseudorandom number generation, multithreading, image processing, regular expressions, and unit testing. It contains 164 individual libraries (as of version 1.76).

All of the Boost libraries are licensed under the Boost Software License, designed to allow Boost to be used with both free and proprietary software projects. Boost is used complementary with the C++ Standard Library to supplement its features. Many of Boost's founders are on the C++ standards committee, and several Boost libraries have been accepted for incorporation into the C++ Technical Report 1, the C++11 standard (e.g. smart pointers, thread, regex, random, ratio, tuple) and the C++17 standard...

C-slowing

*Architectures Simple Symmetric Multithreading in Xilinx FPGAs Post Placement C-Slow Retiming for Xilinx Virtex (.ppt) Post Placement C-Slow Retiming for Xilinx*

C-slow retiming or C-slowing is a technique used in conjunction with retiming to improve throughput of a digital circuit. Each register in a circuit is replaced by a set of registers in series, where C is the number of registers in each of those sets. This creates a circuit with C independent threads, as if the new circuit contained C copies of the original circuit. A single computation of the original circuit takes C times as many clock cycles to compute in the new circuit. C-slowing by itself increases latency, but throughput remains the same.

Increasing the number of registers allows optimization of the circuit through retiming to reduce the clock period of the circuit. In the best case, the clock period can be reduced by a factor of C. Reducing the clock period of the circuit reduces latency...

C++ string handling

*"Optimizations That Aren't (In a Multithreaded World)". C/C++ Users Journal. 17 (6). Stroustrup, Bjarne; Sutter, Herb, eds. (8 May 2025). "C++ Core Guidelines"*

The C++ programming language has support for string handling, mostly implemented in its standard library. The language standard specifies several string types, some inherited from C, some designed to make use of the language's features, such as classes and RAII. The most-used of these is std::string, however std::string_view is also used.

Since the initial versions of C++ had only the "low-level" C string handling functionality and conventions, multiple incompatible designs for string handling classes have been designed over the years and are still used instead of std::string, and C++ programmers may need to handle multiple conventions in a single application.

Embedded C++

*information features of standard C++ were either insufficient or not efficient enough for use in a high-performance, multithreaded kernel. "EC++ Rationale".*

Embedded C++ (EC++) is a dialect of the C++ programming language for embedded systems. It was defined by an industry group led by major Japanese central processing unit (CPU) manufacturers, including NEC, Hitachi, Fujitsu, and Toshiba, to address the shortcomings of C++ for embedded applications. The goal of the effort is to preserve the most useful object-oriented features of the C++ language yet minimize code size while maximizing execution efficiency and making compiler construction simpler. The official website states the goal as "to provide embedded systems programmers with a subset of C++ that is easy for the average C programmer to understand and use".

Herb Sutter

*better support multithreading (such as multi-core processors), and Software developers would be forced to develop massively multithreaded programs as a*

Herb Sutter is a prominent C++ expert. He is also an author of several books on C++ and was a columnist for Dr. Dobb's Journal.