

# Bibliography

- [1] *Debugging with gdb*. Available: <http://sourceware.org/gdb/current/onlinedocs/gdb/>. 2017.
- [2] *Gnu make manual*. Available: [www.gnu.org/software/make/manual/](http://www.gnu.org/software/make/manual/). 2016.
- [3] *How initialization functions are handled*. Available: <https://gcc.gnu.org/onlinedocs/gccint/Initialization.html>. 2017.
- [4] *Using ld, the gnu linker*. Available: [www.math.utah.edu/docs/info/ld\\_3.html](http://www.math.utah.edu/docs/info/ld_3.html). 1994.
- [5] *What is map-reduce?* Available: [www-01.ibm.com/software/data/infosphere/hadoop/mapreduce/](http://www-01.ibm.com/software/data/infosphere/hadoop/mapreduce/). 2017.
- [6] Jeff Andrews. *Branch and loop reorganization to prevent mispredicts*. Available: <https://software.intel.com/en-us/articles/branch-and-loop-reorganization-to-prevent-mispredicts>. May 2011.
- [7] C11 language standard—committee draft. [www.open-std.org/jtc1/sc22/wg14/www/standards](http://www.open-std.org/jtc1/sc22/wg14/www/standards). April 2011.
- [8] Luca Cardelli and Peter Wegner. On understanding types, data abstraction, and polymorphism. *ACM Comput. Surv.* 17(4):471–523. December 1985.
- [9] Ryan A. Chapman. *Linux 3.2.0-33 syscall table, x86 64*. [www.cs.utexas.edu/~bismith/test/syscalls/syscalls64\\_orig.html](http://www.cs.utexas.edu/~bismith/test/syscalls/syscalls64_orig.html).
- [10] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to algorithms*. New York: McGraw-Hill Higher Education, 2nd ed., 2001.
- [11] Russ Cox. *Regular expression matching can be simple and fast*. <https://swtch.com/~rsc/regexp/regexp1.html>. November 2007.
- [12] Ulrich Drepper. *What every programmer should know about memory*. <https://people.freebsd.org/~lstewart/articles/cpumemory.pdf>. November 2007.
- [13] Ulrich Drepper. *How to write shared libraries*. <https://software.intel.com/sites/default/files/m/a/1/e/dsohowto.pdf>. December 2011.
- [14] Jens Gustedt. *Myth and reality about inline in c99*. <https://gustedt.wordpress.com/2010/11/29/myth-and-reality-about-inline-in-c99/>. 2010.



- [15] Intel Corporation. Intel® 64 and IA-32 architectures software developer's manual. Available: [www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-manual-325462.pdf](http://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-manual-325462.pdf). September 2014.
- [16] Intel Corporation. *Intel® 64 and IA-32 architectures optimization reference manual*. Available: [www.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architectures-optimization-manual.html](http://www.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architectures-optimization-manual.html). June 2016.
- [17] David Kanter. *Intel's Haswell CPU microarchitecture*. Available: [www.realworldtech.com/haswell-cpu/1](http://www.realworldtech.com/haswell-cpu/1).
- [18] Brian W. Kernighan. *The C Programming Language*. Prentice Hall Professional Technical Reference, 2nd edition, 1988.
- [19] Petter Larsson and Eric Palmer. Image processing acceleration techniques using intel streaming simd extensions and intel advanced vector extensions. January 2010.
- [20] Doug Lea. *A memory allocator*. <http://g.oswego.edu/dl/html/malloc.html>. 2000.
- [21] Michael E. Lee. *Optimization of computer programs in c*. Available: <http://leto.net/docs/C-optimization.php>.
- [22] Lomont, Chris. "Fast inverse square root." Tech-315 nical Report (2003): 32. February 2003.
- [23] Love, Robert. *Linux Kernel Development* (Novell Press). Novell Press, 2005.
- [24] Michael Matz, Jan Hubicka, Andreas Jaeger, and Mark Mitchell. System V Application Binary Interface. AMD64 Architecture Processor Supplement. Draft version 0.99.6, 2013.
- [25] McKenney, Paul E. "Memory barriers: a hardware view for software hackers." Linux Technology Center, IBM Beaverton (2010).
- [26] Pawell Moll. How do debuggers (really) work? In *Embedded Linux Conference Europe*, October 2015. [http://events.linuxfoundation.org/sites/events/files/slides/slides\\_16.pdf](http://events.linuxfoundation.org/sites/events/files/slides/slides_16.pdf).
- [27] *The netwide assembler: NASM manual*. Available: [www.nasm.us/doc/](http://www.nasm.us/doc/).
- [28] N. N. Nepeyvoda and I. N. Skopin. *Foundations of programming*. RHD Moscow-Izhevsk, 2003.
- [29] Benjamin C. Pierce. *Types and programming languages*. Cambridge, MA: MIT Press, 1st ed. 2002.
- [30] Jeff Preshing. *The purpose of memory order consume in c++11*. [http://preshing.com/20140709/the-purpose-of-memory\\_order\\_consume-in-cpp11/](http://preshing.com/20140709/the-purpose-of-memory_order_consume-in-cpp11/). 2014.
- [31] Jeff Preshing. *Weak vs. strong memory models* <http://preshing.com/20120930/weak-vs-strong-memory-models/>. 2012.
- [32] Brad Rodriguez. *Moving forth: a series on writing Forth kernels*. <http://www.bradrodriguez.com/papers/moving1.html>. The Computer Journal #59 (January/February 1993).



