Efficient Memory Programming
"Code bloat" causes applications to run more slowly than everyone would like. One way of keeping programs running at an acceptable speed is through memory efficient handling. This text shows the programmer how to create efficient code by demonstrating how memory works.

**Synopsis**

Having a good understanding of the impact of the memory hierarchy is discussed throughout the book, including CPU registers, CPU caches, virtual memory, and the hard disk. Cost to access data at each level is also discussed. The concept locality of reference (temporal and spatial) is discussed and how that can affect performance. For example, while a linked list is a useful structure, it also has performance impacts because each node in the list can be in very different locations in memory, potentially causing a page fault for every node. This book is for the C/C++ programmer that wants to manage their own memory for high performance. It is not as useful for Basic or other interpreted languages. The author stays pretty general, but gives short examples using the Pentium processor. I have worked on creating a high-performance server and learned a few things from the book, as well as having some of my assumptions confirmed. The book is well worth your time if you're trying to create a high-performance application.
This is my first book on efficient programming technique. The book offers brief description on how different kinds of memories are organized (specifically, cache and virtual memory). In addition, it briefly introduces data dependencies analysis (which is what I was looking for). The book does not offer detail enough information on how to do the analysis. Also, the examples in the book are too simple to be applied to real-life problem. I was especially disappointed by the content of chapter 10 (the chapter on advance optimization). I was expecting something that I could use on my code, but all the chapters talked about was optimization that could only be done by compilers. In conclusion, I think this book is only good for beginners (like me :) who can benefit from some basic ideas on how cache and virtual memory operates and on how the compiler performs some of the optimizations.

This book is academic in nature. The material is well structured and explained in the way that is similar to advanced college writings. It gives pretty good overview of the hardware memory organization, principles of operation, and extensive overview of optimization techniques primarily targeted at people looking for general knowledge in that field but without much desire to get into details. Due to the good structural and logical organization this book is a perfect material to base an advanced computer science course on. But! If you’re looking for examples of practical use (and abuse) of memory to dramatically boost the performance of your application, the book holds little or no value for you. Starting from the fact that it only briefly describes the internals of PII processor and it has nothing on PIII, PIV and AMD Athlon principles of operating with memory and finishing with code examples which are very simplistic and impossible to take advantage of. If you’re into real-world memory “hacking” and not the academic stuff take a look at "Code optimization: effective memory usage" by Kris Kaspersky.

Download to continue reading...
