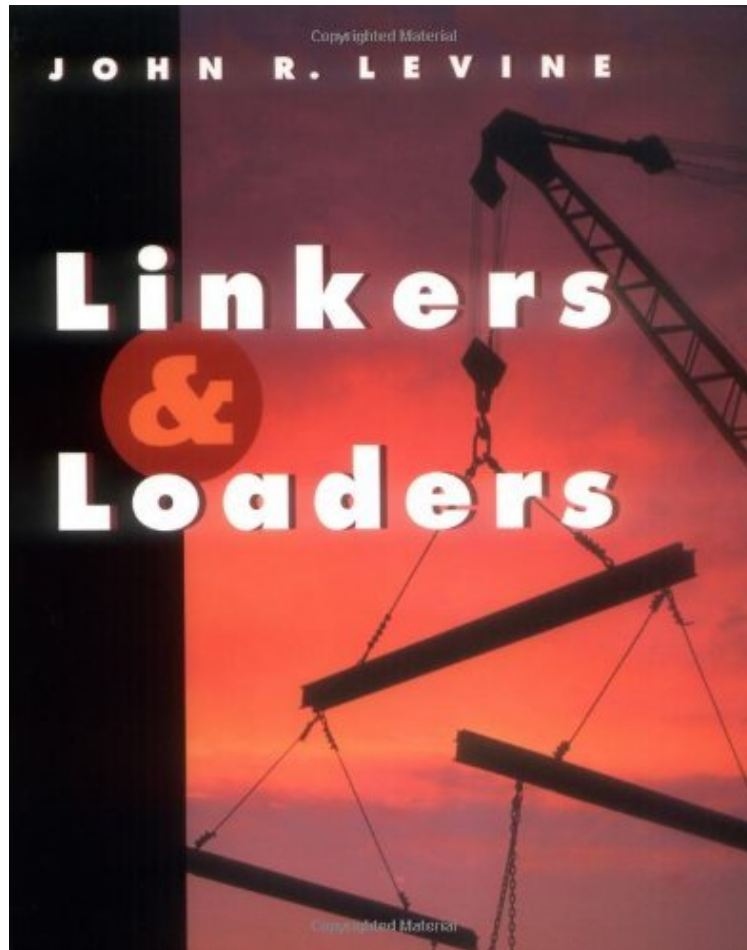


(Free) Linkers and Loaders (The Morgan Kaufmann Series in Software Engineering and Programming)

Linkers and Loaders (The Morgan Kaufmann Series in Software Engineering and Programming)

Von John R. Levine

ebooks / Download PDF / *ePub / DOC / audiobook



Produktinformation -Verkaufsrank: #341620 in eBooksVerffentlicht am: 1999-10-25Erscheinungsdatum: 1999-10-25File Name: B008HNGAG0 | File size: 55.Mb

Von John R. Levine : Linkers and Loaders (The Morgan Kaufmann Series in Software Engineering and Programming) before purchasing it in order to gage whether or not it would be worth my time, and all praised Linkers and Loaders (The Morgan Kaufmann Series in Software Engineering and Programming):

KundenrezensionenHilfreichste Kundenrezensionen2 von 2 Kunden fanden die folgende Rezension hilfreich. Shines a light on an obscure subjectVon Joshua R. PoulsonI picked up this book to delve into interesting problems with a loader that I work with and was amazed at the great story told of what happens to your code after it goes through that compiler and becomes an object. It's not done yet, folks.This book covers a broad range of topics, after first explaining the basics and architecture gotchas, to all the phases from the back end of the compiler to a program running in memory. Three example platforms are used to illustrate this journey: Intel x86 and 32-bit Windows, UltraSPARC and Solaris, and the IBM 360/370. However, it touches upon a great deal other challenges and formats.Some might

consider the symbolic journey from source code to running program to be equivalent to Conrad's "Heart of Darkness" but Levine's book makes it more like a guided tour at a Disney Theme park. There are wonderful examples and code snippets. Clear diagrams and excellent writing. My only complaint about this book is that the back cover makes a lot of noise about Java, but the material inside is pretty paltry alongside the more developed material on C, FORTRAN, and C++ issues. Java is really not that complicated or important to Linkers and Loaders.

1 von 1 Kunden fanden die folgende Rezension hilfreich. Good resource
Von Ein Kunde I am very excited when I discovered this book from . This book covered an important practical area of knowledge for software developers which was not mentioned elsewhere. This book is really very helpful to understand ROUGHLY how current linker and loader works if you just have a vague idea of how they works like me. On the other hand the author try to cover too much so that reader is lost before they really get the high level picture of exactly how various technique in different chapters play together, though the author's wide knowlegde in this area is amazing. I found I still need to read the specifications of ELF format and read the book again to gain the high level picture. I believe the book can be more helpful to an average software developer if the author treatment of ELF and PE is in greater detail that it is in the book. The treatment about C++ is very interesting. If there are second edition, hope to see more about this subject.

1 von 1 Kunden fanden die folgende Rezension hilfreich. Who knew linkers could be interesting?
Von JRZ This book will definitely surprise you if, like me, you had the idea that linkers were just simple, dull tools that ran in the background. The author provides a surprisingly thorough, but understandable, overview of all sorts of linker/loader issues: layout in memory, relocatable code, debugging information, etc., but he does it with such a light, readable style that the material itself doesn't seem particularly difficult. My one complaint is that the book has very little code. The perl-based linker project is basically just a short series of "suggested exercises" without much direction. Still, I wouldn't hesitate to recommend this to anyone interested in compilers or operating systems.

Kurzbeschreibung Whatever your programming language, whatever your platform, you probably tap into linker and loader functions all the time. But do you know how to use them to their greatest possible advantage? Only now, with the publication of *Linkers Loaders*, is there an authoritative book devoted entirely to these deep-seated compile-time and run-time processes. The book begins with a detailed and comparative account of linking and loading that illustrates the differences among various compilers and operating systems. On top of this foundation, the author presents clear practical advice to help you create faster, cleaner code. You'll learn to avoid the pitfalls associated with Windows DLLs, take advantage of the space-saving, performance-improving techniques supported by many modern linkers, make the best use of the UNIX ELF library scheme, and much more. If you're serious about programming, you'll devour this unique guide to one of the field's least understood topics. *Linkers Loaders* is also an ideal supplementary text for compiler and operating systems courses.

- *Includes a linker construction project written in Perl, with project files available for download.
- *Covers dynamic linking in Windows, UNIX, Linux, BeOS, and other operating systems.
- *Explains the Java linking model and how it figures in network applets and extensible Java code.
- *Helps you write more elegant and effective code, and build applications that compile, load, and run more efficiently.

.de Written for any programmer who works with compiled code, *Linkers and Loaders* surveys today's hardware platforms with a tour of how code is linked and executed on IBM mainframes, Unix, and Windows. This handy title fills a valuable niche for anyone who wants to understand how programs are built and run on today's computing systems. It's the cross-platform perspective that distinguishes this book. The author's wide-ranging perspective on IBM 370 mainframes, RISC platforms like the SUN SPARC and, of course, Microsoft Windows makes this book a commendable reference on the internals of linkers and program execution in each environment. There's also a digestible guide to the computer architecture (including registers, instruction formats, and memory addressing) for each platform. (Unix programmers will be pleased that the book has more information on non-Windows platforms than on Windows itself.) For C++ programmers, this text gives you a glimpse into the internals of such language features as macros, templates, and name mangling, and how linkers deal with them at build time. The book closes with useful material on static libraries and dynamic linking, plus a short tour of Java and its class loader (which can resolve classes on the fly as they are downloaded over the Internet). Short exercises are provided for each chapter, making this a useful resource for both classroom and self-study on what is an often overlooked topic.

--Richard Dragan Topics covered: History of linkers and loaders, application binary interfaces (ABIs), computer architecture basics, big- and little-endian memory addresses, register and instruction formats for IBM 370, SPARC and Intel x86, paging and virtual memory, position independent code (PIC), Intel x86 segmentation, embedded architectures, object files for DOS COM and EXE files, Unix a.out, Unix ELF, IBM 360 object format, Microsoft Portable Executable (PE) format, Intel Object Module Format (OMF), storage allocation, linking details for C++, symbol management, name mangling, weak and strong references, debugging information, library formats, COFF and ELF formats, relocation, loading and overlays, bootstrap loading, shared libraries, dynamic linking for Unix ELF and Microsoft Windows DLLs, advanced linking techniques for C++, and linking in Java.

Pressestimmen "I enjoyed reading this useful overview of the

techniques and challenges of implementing linkers and loaders. While most of the examples are focused on three computer architectures that are widely used today, there are also many side comments about interesting and quirky computer architectures of the past. I can tell from these war stories that the author really has been there himself and survived to tell the tale."--Guy Steele-I enjoyed reading this useful overview of the techniques and challenges of implementing linkers and loaders. While most of the examples are focused on three computer architectures that are widely used today, there are also many side comments about interesting and quirky computer architectures of the past. I can tell from these war stories that the author really has been there himself and survived to tell the tale.--Guy Steele