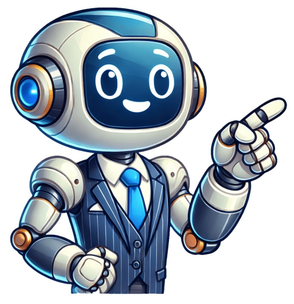
Click Here



Linkers and loaders by john r. levine

Page 2It looks like you're offline. Overview View 0 Editions Details Reviews Lists Related Books August 11, 2024 Edited by MARC Bot import existing book December 20, 2023 Edited by ImportBot import existing book September 13, 2021 Edited by ImportBot import existing book December 13, 2020 Edited by colejhudson Added series and copyright date. April 30, 2008 Created by an anonymous user Imported from amazon.com record 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. Everyday ... Linkers & Loaders is also an ideal supplementary text for compiler and operating systems courses. My takeaway is that the basic principles of linkers and loaders are straightforward enough but that every single specific implementation is a collection of nasty hacks influenced by the operating system and hardware.I came away thinking that there must be a better set of references online by now; if I had to guess I would start off at Wikipedia.December 10, 2010This is an *excellent* introduction into the requirements of both static linkers and dyanmic linkers (loaders). If you're serious about programming, you'll devour this unique guide to one of the field's least understood topics. Whatever your programming language, whatever your platform, you probably tap into linker and loader functions all the time. Still a great book on the subject.November 13, 2013You may have written hundreds, maybe thousands of programs, but if you are like most programmers then everything that happens after the compilation is kind of mysterious. (ISBN: 9781558604964) from Amazon's Book Store. Although I am a practicing C programmer and I consider I have the aforementioned knowledge, there were still parts of the book I didn't understand.The best part of the book for me was the last 3 chapters, which describe static and dynamic shared libraries in more depth and some advanced techniques in linking, like for C++ or Java.What I didn't like is that you can't read only the parts of the book that interests you, because the author is often referencing previous information from the book. Even if you're not, at least skim through the book to get a better understanding of linkers.November 18, 2018The book is certainly dated. This was a much denser read than expected. by Levine, John R. 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. 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. This would normally be fine, but in this case, I would have liked to skip stuff I have no interest for, like SPARC or IBM 360.August 8, 2023I've been tinkering with building an emulator for the PS4 and this book has been indispensable in understanding how dynamic linking and loading works in detail. 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. It is a must read for anyone who is thinking about hacking on a linker or loader e.g. the GNU linker or the dynamic loader in the GNU C library.June 16, 2012Displaying 1 - 15 of 15 reviewsGet help and learn more about the design. "I enjoyed reading this useful overview of the techniques and challenges of implementing linkers and loaders. There's also a digestible guide to the computer architecture (including registers, instruction formats, and memory addressing) for each platform. Overall, great book, but dated.December 18, 2021Quality book, still holds up fairly well despite being decades oldNegatives: fair amount of errors, so you need to consult the webpage errata. I considered rounding down for obsolete hardware. He holds a Ph.D. in Computer Science from Yale University. What are they? 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. Perhaps a case study, i.e. going through every single step towards running a complete program, would have been useful, instead of exposing how different systems solve the different steps one by one.Until I read this book I simply did not understand how a program actually ran on my computer. 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. What is this so-called linker who combines those files into a library, or an executable? "I enjoyed reading this useful overview of the techniques and challenges of implementing linkers and loaders. Why does the compiler have to create object files? Updated title, dimensions and pages. 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. Now the bad part is that there is too much information on obsolete systems, that may clutter some of the more detailed workings of the linker on a particular format. Would be a great resource if you're dealing with those older formats. 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. Loading1.4 Compiler Drivers1.5 Linking: A True Life Example2 Architectural Issues2.1 Application Binary Interfaces2.2 Memory Addresses2.3 Address Formation2.4 Instruction Formats2.5 Procedure Calls and Accessibility2.6 Data and Instruction References2.7 Paging and Virtual Memory2.8 Intel 386 Segmentation2.9 Embedded Architectures3 Object Files3.1 What Goes Into an Object File?3.2 The Null Object Format: DOS COM Files3.3 Code Sections: UNIX a.out Files3.4 Relocation: DOS EXE Files3.5 Symbols and Relocation3.6 Relocatalble a.out3.7 UNIX ELF3.8 IBM 360 Object Format3.9 Microsoft Portable Executable Format3.10 Intel/Microsoft OMF Files3.11 Comparison of Object Formats4 Storage Allocation4.1 Segments and Addresses4.2 Simple Storage Layout4.3 Multiple-Sgement Types4.4 Segment and Page Alignment4.5 Common Blocks and Other Special Segments4.6 Linker Control Scripts4.7 Storage Allocation in Practice5 Symbol Management5.1 Binding and Name Resolution5.2 Symbol Table Formats5.3 Name Mangling5.4 Weak External and Other Kinds of Symbols5.5 Maintaining Debugging Information6 Libraries6.1 Purpose of Libraries6.2 Library Formats6.3 Craeting Libraries6.4 Searching Libraries6.5 Performance Issues6.6 Weak External Symbols7 Relocation7.1 Hardware and Software Relocation7.2 Link-Time and Load-Time Relocation7.3 Symbol and Segment Relocation7.4 Basic Relocation Techniques7.5 Relinkable and Relocatable Output Formats7.6 Other Relocation Formats7.7 Relocaton Special Cases8 Loading and Overlays8.1 Basic Loading8.2 Basic Loading, with Relocation8.3 Position-Independent Code8.4 Bootstrap Loading8.5 Tree-Structured Overlays9 Shared Libraries9.1 Binding Time9.2 Shared Libraries in Practice9.3 Address Space Management9.4 Structure of Shared Libraries9.5 Creating Shared Libraries9.6 Limking With Shared Libraries9.7 Running With Shared Libraries9.8 The malloc Hack and Other Shared-Library Problems10 Dynamic Linking and Loading10.1 ELF Dynamic Linking10.2 Contents of an ELF File10.3 Loading a Dynamically Linked Program10.4 Lazy Procedure Linking with the PLT10.5 Other Peculiarities of Dynamic Linking10.6 Dynamic Loading of Run Time10.7 Microsoft Dynamic-Link Libraries10.8 OSF/1 Pseiudo-Static Shared Libraries10.9 Making Shared Libraries Fast10.10 Comparison of Dynamic Linking Approaches11 - Advanced Techniques11.1 Techniques for C++11.2 Incremental Linking and Relinking11.3 Link-Time Garbage Collection11.4 Link-Time Optimization11.5 Link-Time Code Generation11.6 The Java Linking Model Jump to ratings and reviewsWhatever your programming language, whatever your platform, you probably tap into linker and loader functions all the time. Much of the underlying architecture described is obsolete but, the linkers and loaders, along with the projects remain evergreen and are a fantastic resource. But still very useful and a great introduction to linkers and loaders. 49 people are currently readingDisplaying 1 - 15 of 15 reviewsJanuary 13, 2013Definitely worth reading if you write native code for a living (or hobby).Published in 2000, it covers Linux (ELF and a.out), Windows (COFF) and a huge mess of older obsure object formats and linking systems. Linkers & Loaders is also an ideal supplementary text for compiler and operating systems courses. Anybody , who has little understanding of computer can easily grasp at ease. Enjoyed very much , brushed and enhanced my understanding ,hope this will do a hell lot of good for me to poking into some of the stuff I dealt with day by day basis.This book written in extremely easy and consumable manner. 1 Linking and Loading1.1 What do Linkers and Loaders Do?1.2 Address Binding: A Histrical Perspective1.3 Linking vs. It's the cross-platform perspective that distinguishes this book. Unfortunately though I normally have given this five stars, I just can't in all honesty recommend this to most people, even though it's really well written. I wish I had read this book 10 years ago when I was still working with a custom loader implementation.September 18, 2017Given that it is now around 20 years old, and contains a lot of historical references, the relevancyof the content is so-so. In the end, the book presents recollection of dirty hacks from existing (and dated) implementation, however with explanations why there're these hacks in the first place. --Richard DraganTopics 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. *Explains the Java linking model and how it figures in network applets and extensible Java code. On top of this foundation, the author presents clear practical advice to help you create faster, cleaner code. I can tell from these war stories that the author really has been there himself and survived to tell the tale." -Guy SteeleWhatever your programming language, whatever your platform, you probably tap into linker and loader functions all the time. No solutions to exercises available.As an embedded engineer, by the end of the book I felt more and more things were unnecessary. A few details are still a bit fuzzy, but now I feel much better equipped for dealing with obscure linker errors or custom linker scripts. *Covers dynamic linking in Windows, UNIX, Linux, BeOS, and other operating systems. 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. It carefully explains step by step what happens from the moment the code is compiled until it actually runs on the machine; and what's more important, it makes it very clear why things are as they are today.I was recommended this book in a reply to a Stackoverflow question, and I am not disappointed. Linkers and Loaders are a fundamental part of modern software linking together assets with different rates of change (libraries which change rarely with applications which change more often). You'll learn to avoid the pitfalls associated with Windows DLLs, take advantage of ... 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 ... keeping tasks, with more powerful linkers doing the work. I think an Operating Systems course and at least a basic understanding of assembly language (and related knowledge like registries, counters, stack, etc.) are required to get something out of the book. John Levine is the author or co-author of many books, including lex & yac (O'Reilly), Programming for Graphics Files in C and C++ (Wiley), and The Internet for Dummies (IDG). Recommended, to the curious people wanted to know the inner working and importantly enhance the understanding of their curiosity.June 27, 20214.5, rounded down. Pretty much the only book of it's kind and a great primer if you're interested in writing your own linker or loader for an OS or just need to better understand native executables.It sometimes got lost in the ghosts of systems past like the IBM 360 object format, DOS .OBJ files (Intel OMF), real mode (ugh). He is also publisher emeritus of the Journal of C Language Translation, long-time moderator of the comp.compilers newsgroup, and the creator of one of the first commercial Fortran 77 compilers. But I'd rather have a book on design of new linkers and loaders, or at least in-depth analysis of at least one loader. 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. 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. The book itself is not necessarily outdated, because the principles haven't changed, but the description of some file formats is certainly outdated.I read it because I wanted to learn more about the linking process in particular, but I got the impression that this is more oriented towards the writer of a linker and/or loader.Thus there is some prerequisite knowledge the author assumes you already have. I understand new loaders are very rarely designed, but anyway, this book is of no help when they do.March 22, 2019Although very old, this is probably the book on the present subject. *Helps you write more elegant and effective code, and build applications that compile, load, and run more efficiently. The book goes occasionally perhaps a little bit too much into technical details, which I felt could be safely skipped. It'd be nice to have time to come back to it and write the projects but it's definitely still the best book on linking which is normally just left to shrugs.August 8, 2019Very good, in depth book on object formats (on a multitude of systems), and how linkers employ a bunch of hacks for the most part, to tie everything together and function. (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. What's its purpose? Highly recommended for any programmer who wants to get to the bottom of things.January 29, 2008I tried this out because of Greg Wilson's recommendation at at a point where I was wrestling with some shared library implementation questions.The book was a little too specific for me and I couldn't get much out of it. Linkers will also be more involved in global program optimization, since the linker is the only stage of the compiler process that ... Fascinating to see how this system evolved and everyone converged on a similar set of solutions.These parts of systems everyone depends on but few understand are opportunities to more deeply understand a critical system. This handy title fills a valuable niche for anyone who wants to understand how programs are built and run on today's computing systems. If you are looking an intro into writing linkers, this would be a good place to start. But do you know how to use them to their greatest possible advantage? I can tell from these war stories that the author really has been there himself and survived to tell the tale." —Guy SteeleWhatever your programming language, whatever your platform, you probably tap into linker and loader functions all the time. Linkers & Loaders is also an ideal supplementary text for compiler and operating systems courses.Features:* 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.Buy Linkers and Loaders (The Morgan Kaufmann Series in Software Engineering and Programming) Revised ed. The book begins with a detailed and comparative account of linking and loading that illustrates the differences among various compilers and operating systems. John Levine's book answers those questions, and more.Item 53 in 97 Things Every Programmer Should Know: Collective Wisdom from the Experts is "The Linker Is not a Magical Program", and this book goes a long way towards taking that magic away. My only semi-complaint is that it spent a lot of pages talking about real mode but being published in 2000, we were less than a decade removed from that fever dream of backwards compatibility.November 15, 2019This is a 'must read' book for software developers, it CONTAINS a lot of best practices and advices that are proven to be useful.February 26, 2013You barely can understand linkers by this book since it falls into implementation details of each specific OS and CPU too early. Features: * Includes a linker construction project written in Perl, with... There's always an edge where no one is looking. Linkers & Loaders is also an ideal supplementary text for compiler and operating systems courses.Features:* 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. [Just see Apple's recent improvement in linker and loader performance](and projects like [mold](January 14, 2021Well, such an important and essential topic, or should I say,the crux of the every possible software program execution on any environment .