

Declaration of Rachel J. Watters on Authentication of Publication

I, Rachel J. Watters, am a librarian, and the Head of Resource Sharing for the General Library System, Memorial Library, located at 728 State Street, Madison, Wisconsin, 53706. Part of my job responsibilities include oversight of Wisconsin TechSearch (“WTS”), an interlibrary loan department at the University of Wisconsin-Madison. I have worked as a librarian at the University of Wisconsin library system since 1998, starting as a graduate student employee in the Kurt F. Wendt Engineering Library and WTS, then as a librarian in Interlibrary Loan at Memorial Library. I began professional employment at WTS in 2002 and became WTS Director in 2011. In 2019, I became Head of Resource Sharing for UW-Madison’s General Library System. I have a master’s degree in Library and Information Studies from the University of Wisconsin-Madison. Through the course of my studies and employment, I have become well informed about the operations of the University of Wisconsin library system, which follows standard library practices.

This Declaration relates to the dates of receipt and availability of the following:

Levine, J.R. (2000). *Linkers and Loaders*. Academic Press.

Standard operating procedures for materials at the University of Wisconsin-Madison Libraries. When a volume was received by the Library, it would be checked in, added to library holdings records, and made available to readers as soon after its arrival as possible. The procedure normally took a few days or at most 2 to 3 weeks.

Declaration of Rachel J. Watters on Authentication of Publication

Exhibit A to this Declaration is a true and accurate copy of the front matter of the *Linkers and Loaders* (2000) publication, which includes a stamp on the verso page showing that this book is the property of the Kurt F. Wendt Library at the University of Wisconsin-Madison.

Attached as Exhibit B is the cataloging system record of the University of Wisconsin-Madison Libraries for its copy of the *Linkers and Loaders* (2000) publication. As shown in the "Receiving date" field of this Exhibit, the University of Wisconsin-Madison Libraries owned this volume and had it cataloged in the system as of February 29, 2000.

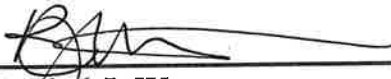
Members of the interested public could locate the *Linkers and Loaders* (2000) publication after it was cataloged by searching the public library catalog or requesting a search through WTS. The search could be done by title, author, and/or subject key words. Members of the interested public could access the publication by locating it on the library's shelves or requesting it from WTS.

I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Declaration of Rachel J. Watters on Authentication of Publication

Date: August 14, 2024

Memorial Library
728 State Street
Madison, Wisconsin 53706



Rachel J. Watters
Head of Resource Sharing

EXHIBIT A

QA

76.76

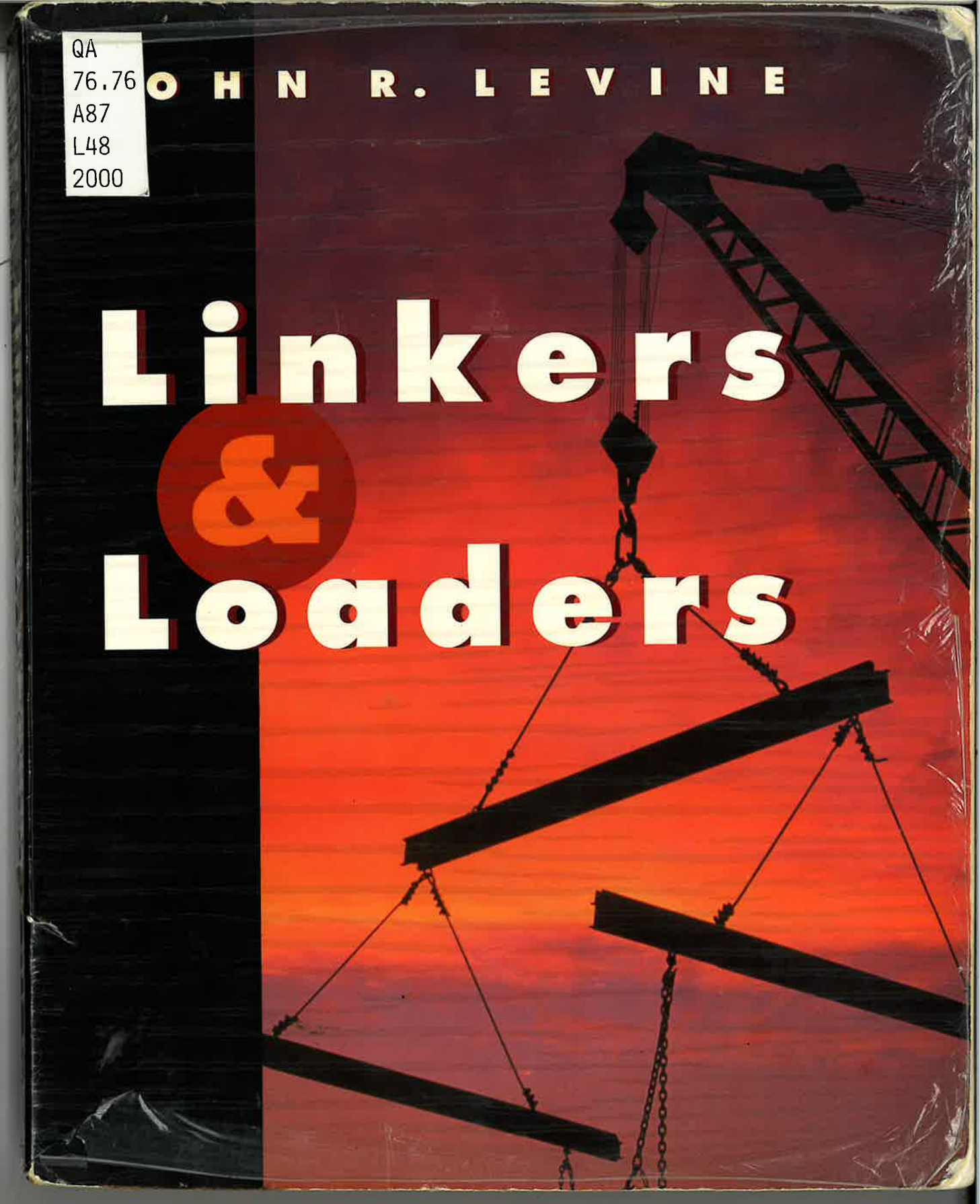
A87

L48

2000

JOHN R. LEVINE

Linkers & Loaders

The background of the book cover is a dramatic, high-contrast photograph of a construction site at sunset or sunrise. The sky is a deep, vibrant red-orange. Silhouetted against this sky are several large, heavy-duty steel beams being hoisted by a crane. The crane's lattice boom is visible in the upper right corner, extending diagonally across the frame. Multiple thick cables are attached to the beams, pulling them upwards. The overall composition is dynamic and industrial, emphasizing the theme of construction and heavy machinery.

Linkers & Loaders

JOHN R. LEVINE

89064930431



B89064930431A

"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

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.

Linkers and Loaders 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.

Features

- 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.
- Includes a linker construction project written in Perl, with project files available for download.

John Levine is the author or co-author of many books, including *lex & yacc* (O'Reilly), *Programming for Graphics Files in C and C++* (Wiley), and *The Internet for Dummies* (IDG). 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. He holds a Ph.D. in computer science from Yale University.



MORGAN KAUFMANN PUBLISHERS

AN IMPRINT OF ACADEMIC PRESS

A Harcourt Science and Technology Company

<http://www.mkp.com>

Programming
Operating Systems

ISBN 1-55860-496-0



9 781558 604964

U.S. \$36.95 U.K. £23.95
CANADA \$51.95

Kurt F. Wendi Library
University of Wisconsin - Madison
215 N. Randall Avenue
Madison, WI 53706-1688

LINKERS AND LOADERS

LINKERS AND LOADERS

John R. Levine



MORGAN KAUFMANN PUBLISHERS

AN IMPRINT OF ACADEMIC PRESS

A Harcourt Science and Technology Company

SAN FRANCISCO SAN DIEGO NEW YORK BOSTON
LONDON SYDNEY TOKYO

Kurt F. Wendt Library
University of Wisconsin - Madison
215 N. Randall Avenue
Madison, WI 53706-1688

Acquisitions Editor Tim Cox
Director of Production and Manufacturing Yonie Overton
Production Editor Edward Wade
Editorial Assistant Sarah Luger
Cover Design Ross Carron Design
Cover Photograph © Greg L. Ryan; Sally Beyer/AllStock/PNI
Text Design, Technical Illustration, and Composition Rebecca Evans & Associates
Copyeditor Jeff Van Bueren
Proofreader Jennifer McClain
Indexer Bruce Tracy
Printer Courier Corporation

Designations used by companies to distinguish their products are often claimed as trademarks or registered trademarks. In all instances where Morgan Kaufmann Publishers is aware of a claim, the product names appear in initial capital or all capital letters. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

ACADEMIC PRESS
A Harcourt Science and Technology Company
525 B Street, Suite 1900, San Diego, CA 92101-4495, USA
<http://www.academicpress.com>

Academic Press
24-28 Oval Road, London, NW1 7DX United Kingdom
<http://www.hbuk.co.uk/ap/>

Morgan Kaufmann Publishers
340 Pine Street, Sixth Floor, San Francisco, CA 94104-3205
<http://www.mkp.com>

© 2000 by Academic Press
All rights reserved
Printed in the United States of America

04 03 02 01 00 5 4 3 2

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopying, or otherwise—without the prior written permission of the publisher.

Library of Congress Cataloging-in-Publication Data

Levine, John R.
Linkers and loaders / John R. Levine.
p. cm.
ISBN 1-55860-496-0

1. Assembling (Electronic computers) 2. Linkers (Computer programs) 3. Loaders (Computer programs) I. Title.

QA76.76.A87 L48 2000
005.4'56—dc21

99-047127

Wendt

QA

76.76

A87

L48

2000

4836329

To Tonia and Sarah

CONTENTS

PREFACE	xi
----------------------	-----------

1	LINKING AND LOADING	1
----------	----------------------------------	----------

- 1.1 What Do Linkers and Loaders Do? 1
- 1.2 Address Binding: A Historical Perspective 1
- 1.3 Linking vs. Loading 5
- 1.4 Compiler Drivers 10
- 1.5 Linking: A True-Life Example 12

Exercises 17

2	ARCHITECTURAL ISSUES	19
----------	-----------------------------------	-----------

- 2.1 Application Binary Interfaces 20
- 2.2 Memory Addresses 20
- 2.3 Address Formation 22
- 2.4 Instruction Formats 23
- 2.5 Procedure Calls and Addressability 24
- 2.6 Data and Instruction References 28
- 2.7 Paging and Virtual Memory 34
- 2.8 Intel 386 Segmentation 41
- 2.9 Embedded Architectures 43

Exercises 45

3 OBJECT FILES 47

- 3.1 What Goes into an Object File? 47
- 3.2 The Null Object Format: DOS COM Files 49
- 3.3 Code Sections: UNIX a.out Files 50
- 3.4 Relocation: DOS EXE Files 56
- 3.5 Symbols and Relocation 58
- 3.6 Relocatable a.out 58
- 3.7 UNIX ELF 62
- 3.8 IBM 360 Object Format 71
- 3.9 Microsoft Portable Executable Format 75
- 3.10 Intel/Microsoft OMF Files 84
- 3.11 Comparison of Object Formats 90

Exercises 90

Project 90

4 STORAGE ALLOCATION 93

- 4.1 Segments and Addresses 93
- 4.2 Simple Storage Layout 94
- 4.3 Multiple-Segment Types 96
- 4.4 Segment and Page Alignment 96
- 4.5 Common Blocks and Other Special Segments 98
- 4.6 Linker Control Scripts 107
- 4.7 Storage Allocation in Practice 108

Exercises 114

Project 115

5 SYMBOL MANAGEMENT 117

- 5.1 Binding and Name Resolution 117
- 5.2 Symbol Table Formats 118
- 5.3 Name Mangling 124
- 5.4 Weak External and Other Kinds of Symbols 129
- 5.5 Maintaining Debugging Information 129

Exercises 132

Project 133

6 LIBRARIES 135

- 6.1 Purpose of Libraries 135
- 6.2 Library Formats 136
- 6.3 Creating Libraries 141
- 6.4 Searching Libraries 142
- 6.5 Performance Issues 144
- 6.6 Weak External Symbols 144

Exercises 145

Project 146

7 RELOCATION 149

- 7.1 Hardware and Software Relocation 150
- 7.2 Link-Time and Load-Time Relocation 151
- 7.3 Symbol and Segment Relocation 151
- 7.4 Basic Relocation Techniques 153
- 7.5 Relinkable and Relocatable Output Formats 159
- 7.6 Other Relocation Formats 160
- 7.7 Relocation Special Cases 162

Exercises 163

Project 164

8 LOADING AND OVERLAYS 167

- 8.1 Basic Loading 167
- 8.2 Basic Loading, with Relocation 168
- 8.3 Position-Independent Code 169
- 8.4 Bootstrap Loading 176
- 8.5 Tree-Structured Overlays 177

Exercises 184

Project 185

9 SHARED LIBRARIES 187

- 9.1 Binding Time 188
- 9.2 Shared Libraries in Practice 190
- 9.3 Address Space Management 190
- 9.4 Structure of Shared Libraries 192
- 9.5 Creating Shared Libraries 192

9.6	Linking with Shared Libraries	197
9.7	Running with Shared Libraries	197
9.8	The malloc Hack and Other Shared-Library Problems	199
	Exercises	202
	Project	203

10 DYNAMIC LINKING AND LOADING 205

10.1	ELF Dynamic Linking	206
10.2	Contents of an ELF File	206
10.3	Loading a Dynamically Linked Program	210
10.4	Lazy Procedure Linkage with the PLT	213
10.5	Other Peculiarities of Dynamic Linking	215
10.6	Dynamic Loading at Run Time	216
10.7	Microsoft Dynamic-Link Libraries	217
10.8	OSF/1 Pseudo-Static Shared Libraries	222
10.9	Making Shared Libraries Fast	223
10.10	Comparison of Dynamic Linking Approaches	224
	Exercises	226
	Project	226

11 ADVANCED TECHNIQUES 229

11.1	Techniques for C++	229
11.2	Incremental Linking and Relinking	233
11.3	Link-Time Garbage Collection	235
11.4	Link-Time Optimization	237
11.5	Link-Time Code Generation	238
11.6	The Java Linking Model	241
	Exercises	244
	Project	245

REFERENCES 246

INDEX 249

EXHIBIT B



Resource description

Linkers and loaders / John R. Levine, Levine, John R. Morgan Kaufmann, San Francisco : 2000 [1558604960]

Holdings

Holdings ID

Barcode

Item ID

Process type

MMS ID

Status

[Steenbock Library: Steenbock Stacks: QA76.76 A87 L48 2000](#)

22859344210002122

[View all holdings](#)

89064930431

23625738400002122

[View all items](#)

[Loan](#)

9948363293602122

[Browse shelf listing](#)

Item not in place

GeneralENUM/CHRONNotesHistory

General Information

Barcode

89064930431

Copy ID

1

Material type

Book

Item policy

general (book)

Provenance

-

Is magnetic

No

PO Line

Issue date

-

Receiving date

02/29/2000

Expected receiving date

-

Enumeration A

-

Enumeration B

-

Chronology I

-

Chronology J

-

Description

-

Pages

-

Pieces

1

Replacement cost

-

Receiving operator

import

Physical condition

-

Process type

Inventory Information

Inventory number

-

Inventory date

-

Inventory price

-

Location Information

Permanent location

Steenbock Library: Steenbock Stacks (stk)

Item call number type

-

Item call number

-

Source (Subfield 2)

-

Storage location ID

-

Temporary Location Information

Item is in temporary location

No

Temporary location

-

https://wisconsin-madison.alma.exlibrisgroup.com/ng/page;u=%2Frep%2Faction%2FpageAction.resource_editor.physical.item_general.xml.do%3Fpa... 1/2

Temporary call number type -

Temporary call number -

Temporary source (Subfield 2) -

Temporary item policy -

Due back date -

Retention Information

Committed to Retain -

Retention Reason -

Retention Note -