

Capability-Based Computer Systems

Capability-Based Computer Systems

Henry M. Levy

digital™
DIGITAL PRESS

Copyright © 1984 Digital Equipment Corporation

All rights reserved. Reproduction of this book, in part or in whole, is strictly prohibited. For copy information contact: Digital Press, 12 Crosby Dr., Bedford, Mass. 01730

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Documentation Number: EY-00025-DP
ISBN: 0-932376-22-3

Library of Congress Cataloging in Publication Data

Levy, Henry M., 1952–
Capability-based computer systems.

Bibliography: p. 205

Includes index.

1. Computer architecture. 2. Operating systems (Computers) 3. System design. I. Title.

QA76.9.A73L48 1983 621.3819'58 83-21029

ISBN 0-932376-22-3

Trademarks

Bell Laboratories: UNIX. Burroughs Corporation: B5000. Cambridge University: CAP. Control Data Corporation: CDC 6400, SCOPE. Digital Equipment Corporation: DEC, LSI-11, PDP-1, PDP-11, TOPS-20. Hewlett-Packard Company: HP 3000. Intel Corporation: iAPX 432, iMAX, Intel 8086. IBM: CPF, IBM 370, IBM System/38, SWARD. International Computers Ltd.: Basic Language Machine. Plessey Telecommunications Ltd. Plessey System 250. Xerox Corporation: Smalltalk.

*In Memory of **Manny** and **Sonia***

Preface

The purpose of this book is to provide a single source of information about capability-based computer systems. Although capability systems have existed for nearly two decades, only recently have they appeared in architecture and operating system textbooks. Much has been written about capability systems in the technical literature, but finding this information is often difficult.

This book is an introduction, a survey, a history, and an evaluation of capability- and object-based computer systems. It is intended for students, computer professionals, and computer-system designers. The book assumes a knowledge of the assembly-level architecture of at least one computer, an understanding of memory addressing and virtual memory systems, and some familiarity with operating systems. It can be used as a tutorial or reference text in advanced undergraduate or graduate courses in computer organization, computer architecture, or operating systems.

Chapter 1 introduces the concept of a capability and examines the use of capabilities in computer systems. It compares and contrasts simplified models of capability and conventional addressing and protection systems. The object-based design methodology is introduced, and the use of capabilities to support object-based systems is discussed.

Chapter 2 describes machines that preceded the formal definition of capabilities but had similar addressing mechanisms. Developed in the late 1950s and early 1960s, these machines include the Burroughs B5000, the Rice University computer, and the Basic Language Machine. Chapter 3 examines the Dennis and Van Horn hypothetical supervisor that introduced

the concept of capability, and the early university attempts to implement that concept: the MIT PDP-1 system, the Chicago Magic Number Machine, and the CAL-TSS system.

Chapter 4 describes the Plessey System 250. Built in the U.K., the Plessey 250 multiprocessor was the first commercially available capability-based computer system. Also built in the U.K., Cambridge University's CAP computer system, examined in Chapter 5, was the first successful university implementation of capability hardware.

Chapters 6 and 7 concentrate on two capability-based multiprocessor operating systems built at Carnegie-Mellon University: Hydra and STAROS. These systems were the first major object-based systems and used capabilities to provide object-level addressing and protection.

Chapters 8 and 9 examine the new generation of capability/object-based systems designed for the commercial marketplace: the IBM System/38 and the Intel iAPX 432. The System/38 is the first use of object-based methodology to build a business-oriented computer system. The Intel 432 is the first highly-integrated object-based microprocessor. Both systems use object-based methodology to raise the level of the architecture interface. This allows them to support sophisticated operating-systems operations in hardware.

Chapter 10 reviews many of the important design issues in capability- and object-based systems in light of the implementations discussed throughout the book. Alternative implementation decisions and their implications are examined.

Each survey section presents the important features of a particular system. For this reason, different systems may be described at somewhat different levels. However, all systems are discussed in sufficient detail to give the reader an understanding of both the concepts and the low-level capability addressing and object-support mechanisms. An important goal of the book is examination of hardware and operating-system implementations of capabilities. Although all of the systems begin with a similar conceptual view of capabilities, the implementations are vastly different.

All attempts have been made to see that the system presentations are accurate, and most of the sections have been reviewed by one or more of the system's designers. Still, these discussions should not be taken as the final word and the interested reader is referred to the latest technical literature for more detailed study. Each section contains suggestions for further reading, and a complete bibliography on capability and object systems is included at the end of the book.

Acknowledgments

I was fortunate to have as reviewers many of the creative people who helped design and build the systems described in the book. I would like to thank them for providing an interesting topic of study and for lending their valuable time. I was privileged to have the benefit of the technical and historical insight of Maurice Wilkes and Earl Van Horn. Pete Lee and Guy Almes provided detailed reviews (and re-reviews) of the entire manuscript. These contributed heavily to the book. I would like to thank the many other people who provided critiques of individual chapters.

This book is the result of a study that began when I was a Digital Equipment Corporation resident at the University of Washington. I would like to thank Bill Strecker for supporting my stay at Washington and for providing a creative working environment at Digital. Sam Fuller, Dieter Huttenberger, and Dick Eckhouse also helped to make the residency a success. At Washington, Guy Almes, Ed Lazowska, and John Bennett provided helpful reviews of early drafts of this work. Finally, I would like to thank Sandy Kaplan for her technical assistance, encouragement, patience, and humor that made the writing process more enjoyable.