VAX

STRUCTURED ASSEMBLY LANGUAGE PROGRAMMING

SECOND EDITION

ROBERT W. SEBESTA

VAX

STRUCTURED ASSEMBLY LANGUAGE PROGRAMMING

Second Edition

Robert W. Sebesta
University of Colorado
at Colorado Springs

The Benjamin/Cummings Publishing Company, Inc.
Redwood City, California • Menlo Park, California
Reading, Massachusetts • New York • Don Mills, Ontario
Wokingham, U.K. • Amsterdam • Bonn • Sydney
Singapore • Tokyo • Madrid • San Juan

To Joanne

Sponsoring Editor: John Thompson
Production Coordinator: Eleanor Renner Brown
Cover and Chapter Opener Designer: Eleanor Mennick
Electronic Composition and Art: Ocean View Technical Publications

Library of Congress Cataloging-in-Publication Data
Sebesta, Robert W.
VAX : structured assembly language programming / Robert W. Sebesta. – 2nd ed.
p. cm.
Rev. ed. of VAX II. © 1984.
Includes index.
ISBN 0-8053-7122-2
1. VAX-11 (Computer)–Programming. 2. Assembler language (Computer program language) 3. Structured programming.
I. Sebesta, Robert W. VAX 11 II. Title.
QA76.8.V37S4 1991
005.2'45–dc20 90-26190
CIP

ISBN 0-8053-7122-2

2 3 4 5 6 7 8 9 10 HA 95 94 93 92 91

The Benjamin/Cummings Publishing Company, Inc.
390 Bridge Parkway
Redwood City, California 94065

VAX

STRUCTURED ASSEMBLY LANGUAGE PROGRAMMING

Second Edition

THE BENJAMIN/CUMMINGS SERIES IN COMPUTER SCIENCE

G. Booch,
Object-Oriented Design with Applications (1989)

G. Brookshear
Computer Science: An Overview, Third Edition (1991)

F. Carrano
Assembler Language Programming for the IBM 370 (1988)

D. M. Etter
Structured FORTRAN 77 for Engineers and Scientists, Third Edition (1990)

P. Helman, R. Veroff, and F. Carrano
Intermediate Problem Solving and Data Structures: Walls and Mirrors, Second Edition (1991)

P. Helman and R. Veroff
Walls and Mirrors: Intermediate Problem Solving and Data Structures—Modula II (1988)

N. Miller and C. G. Peterson
File Structures With Ada (1990)

A. Kelley and I. Pohl
A Book on C: An Introduction to Programming in C, Second Edition (1990)

A. Kelley and I. Pohl
C by Dissection: The Essentials of C Programming (1988)

I. Pohl
C++ for C Programmers

I. Pohl
C++ for Pascal Programmers

I. Pohl
Turbo C++ (1991)

W. J. Savitch
Pascal: An Introduction to the Art and Science of Programming, Third Edition (1990)

W. J. Savitch
Turbo Pascal 4.0/5.0 with 5.5 Supplement (1990)

R. Sebesta
VAX Structured Assembly Language Programming, Second Edition (1991)

R. Sebesta
Concepts of Programming Languages (1989)

M. Thorne
Computer Organization and Assembly Language Programming for IBM PCs and Compatibles (1991)

Titles of Related Interest

M. Sobell
A Practical Guide to the UNIX System, Second Edition (1989)

M. Sobell
A Practical Guide to UNIX System V, Version 4 (1991)

PREFACE

This book is intended to be used for the standard course on assembly language programming, with the assumption that the reader has had at least one course in high-level language programming.

Learning an assembly language has two related yet distinct benefits: the acquisition of the skill of producing software in assembly language and gaining an in-depth knowledge of the architecture of a particular computer.

In this book the VAX series of superminicomputers is presented as a vehicle for discussing these two topics, because we believe that the VAX is particularly suited to the study of assembly language. First, the VAX architecture represents that of many other computers. The VAX instruction set includes many of the features of the instruction sets of other computers. As a result, the assembly languages of most other computers are easy to learn for those who know VAX assembly language. The second advantage of studying VAX assembly language lies in the complexity of the VAX architecture. The VAX has a rich set of instructions and addressing modes that make programming it in assembly language far easier than on simpler computers. The obvious price of this higher degree of writeability is the increased difficulty in learning the assembly language. However, the orthogonality of the VAX instruction set goes a long way toward balancing the increase in complexity with a measure of elegant simplicity.

Our approach to software development in assembly language is as follows. A pseudocode solution to a given problem is created first. This pseudocode is then translated, using consistent techniques, to an assembly language program. The pseudocode is included as comments in the final program, providing documentation. This program design methodology is described in detail and used in all example programs. This method, which is clearly a top-down approach, results in well-structured programs that are reasonably easy to read, while sacrificing very little in efficiency.

Macroinstructions are used for terminal input and output. This allows the reader to write complete programs very early in the course. Getting such an early start is important, because we believe that a significant amount of programming experience is necessary to acquire the expertise to be an effective assembly language programmer.

Features of the Book

Program Examples

This book contains a large number of program examples, many of which are complete stand-alone programs. Most of the complete examples are preceded by pseudocode algorithms that describe their actions.

The VAX/VMS Debugger

There is an entire chapter devoted to the VAX/VMS Debugger. This chapter describes the most useful debugger commands, including screen mode use of the debugger. The complete text of a debugging session is also included.

Input/Output Macroinstructions

Macroinstructions for terminal input and output of three integer type values and character strings, and also output for floating-point values, are used for the example programs. The software package that provides these services also includes a macro for dumping registers and memory. The method of obtaining this software is described below.

Student Aids

Every chapter except the first has a summary, lists of new terms and new instructions, and a problem set. Nearly all of the problem sets have both programming and nonprogramming exercises. We have included a description of all of the nonprivileged VAX instructions. Instructors who feel that the entire instruction set cannot be effectively covered in a course can choose sections to exclude according to their tastes.

The VAX Record Management System

The book has a complete chapter on RMS, the VMS subsystem that the VAX/VMS high-level languages use for input and output.

An Alternative Introduction

Appendix A contains a chapter-length introduction to computer organization, machine language, and assembly language programming. This appendix uses a very simple and idealized computer named SIMCOM. The SIMCOM approach to introducing the important concepts of this book is intended to be used for classes in which the students have relatively weak backgrounds in computing (for example, those who have not had a course in computer organization and have never learned anything about instruction sets and low-level programming).

Instructional Software

Both the SIMCOM simulator and the VAX input/output package are available through an anonymous FTP account from node HAPPY.UCCS.COLORADO.EDU. The user id is ANONYMOUS and the password is GUEST. All of the required files are in the directory named MACRO. For further information about software please contact the author at the University of Colorado at Colorado Springs or your local Benjamin/Cummings representative.

Using the Book in the Classroom

Chapter 1 contains background information on the evolution of computer architecture and computer languages, along with some justification for learning any assembly language, and VAX assembly language in particular. Chapter 2 covers the necessary material on binary and hexadecimal numbers, the addition and subtraction operations on binary and hexadecimal numbers, number base conversions, and twos complement notation. Chapters 1 and 2 may be skipped by classes whose students have some background in computer organization and binary and hexadecimal arithmetic.

Most of the essential material of the book appears in Chapters 3-11. Chapter 3 first describes general computer architecture and CPU operation, and then introduces the VAX architecture and a small collection of VAX instructions and directives. The process of writing and running complete programs is also described.

Chapter 4 introduces VAX assembly language implementations of the fundamental program control constructs. The methodology uses standard techniques of implementing pseudocode versions of selection and looping structures.

Chapter 5 is a description of the VAX/VMS debugger, including its use in screen mode. A complete debugging session on an example program is a significant part of the chapter.

Chapter 6 introduces the other VAX integer data types, operand expressions, and simple macros. Chapter 7 discusses the use of indexing for array processing and the VAX implementation of indexing. Chapter 8 introduces the concept of indirect addressing and covers the VAX addressing modes that implement it.

Chapter 9 describes the character manipulation instructions of the VAX and how they can be used for programming solutions to simple problems.

Chapter 10 is a thorough discussion of VAX assembly language subprograms, including the various parameter passing methods, recursion, and subprogram libraries. Chapter 11 covers the remaining features of macros, along with all of the VAX techniques for conditional assembly.

Chapter 12 describes VAX facilities for dealing with floating-point and decimal data. Bit and logic instructions and their applications are discussed in Chapter 13. Chapter 14 briefly describes the fundamental features of the RMS input/output system of VAX/VMS.

Acknowledgments

Textbooks are created over a long period of time by a large number of people. Some of the people who were involved in the development of this book deserve mention here.

The first edition of this book was reviewed by Theodore Bashkow of Columbia University, M. Faiman of the University of Illinois, George Rice of De Anza College, Arthur Gill of the University of California at Berkeley, Henry Leitner of Harvard University (Aiken Computation Laboratory), and Robert Muller of Boston University. Richie L. Lary of Digital performed a technical review of the first edition.

The second edition was reviewed by John Sheehan of San Francisco State University and Myers L. Foreman of Lamar University.

Alan Apt, former editor at Benjamin/Cummings, encouraged and advised me throughout the efforts to produce the two editions of this book.

The UCCS Computer Center provided the use of their VAX systems for the development of the input/output package and the example programs that appear in this book.

Finally, I thank my wife Joanne for her patience during the seemingly endless hours it has taken me to write the two editions of this book.

ROBERT W. SEBESTA

BRIEF CONTENTS

CONTENTS

1

INTRODUCTION

There are a few important preliminaries to a serious study of assembly-language programming. The first two chapters and part of the third chapter of this book cover the most important of these. Chapters 1 and 3 include discussions of the fundamentals of the architecture of digital computers and their operation. Chapter 2 covers the required information on nondecimal numbers and the methods of storing integer data in a computer's memory.

In this chapter we discuss the fundamentals of computers in rather general terms. Included is a brief historical introduction to the development of the hardware and architecture of contemporary computers, and also a brief discussion of the early history of programming languages. With this background we can put assembly language and the VAX family of computers in perspective. This chapter also includes some reasons for studying assembly-language programming in general, and the VAX architecture and assembly language in particular.