Preface

The primary goal of this book is to teach the IA-32 assembly language programming under the Linux operating system. A secondary objective is to provide a gentle introduction to the Fedora Linux operating system. Linux has evolved substantially since its first appearance in 1991. Over the years, its popularity has grown as well. According to an estimate posted on http://counter.li.org/, there are about 18 million Linux users worldwide. Hopefully, this book encourages even more people to switch to Linux.

The book is self-contained and provides all the necessary background information. Since assembly language is very closely linked to the underlying processor architecture, a part of the book is dedicated to giving computer organization details. In addition, the basics of Linux are introduced in a separate chapter. These details are sufficient to work with the Linux operation system.

The reader is assumed to have had some experience in a structured, high-level language such as C. However, the book does not assume extensive knowledge of any high-level language—only the basics are needed.

Approach and Level of Presentation

The book is targeted for software professionals who would like to move to Linux and get a comprehensive introduction to the IA-32 assembly language. It provides detailed, step-by-step instructions to install Linux as the second operating system.

No previous knowledge of Linux is required. The reader is introduced to Linux and its commands. Four chapters are dedicated to Linux and NASM assembler (installation and usage). The accompanying DVD-ROMs provide the necessary software to install the Linux operating system and learn assembly language programming.

The assembly language is presented from the professional viewpoint. Since most professionals are full-time employees, the book takes their time constraints into consideration in presenting the material.

Summary of Special Features

Here is a summary of the special features that sets this book apart:

- The book includes the Red Hat Fedora Core 3 Linux distribution (a total of two DVD-ROMs are included with the book). Detailed step-by-step instructions are given to install Linux on a Windows machine. A complete chapter is used for this purpose, with several screenshots to help the reader during the installation process.
- Free NASM assembler is provided so that the readers can get hands-on assembly language programming experience.
- Special I/O software is provided to simplify assembly language programming. A set of input and output routines is provided so that the reader can focus on writing assembly language programs rather than spending time in understanding how the input and output are done using the basic I/O functions provided by the operating system.
- Three chapters are included on computer organization. These chapters provide the necessary background to program in the assembly language.
- Presentation of material is suitable for self-study. To facilitate this, extensive programming examples and figures are used to help the reader grasp the concepts. Each chapter contains a simple programming example in "Our First Program" section to gently introduce the concepts discussed in the chapter. This section is typically followed by "Illustrative Examples" section, which gives more programming examples.
- This book does not use fragments of code in examples. All examples are complete in the sense that they can be assembled and run, giving a better feeling as to how these programs work. These programs are on the accompanying DVD-ROM (DVD 2). In addition, you can also download these programs from the book's Web site at the following URL: `http://www.scs.carleton.ca/~sivarama/linux_book`.
- Each chapter begins with an overview and ends with a summary.

Overview of the Book

The book is divided into seven parts. Part I provides introduction to the assembly language and gives reasons for programming in the assembly language. Assembly language is a low-level language. To program in the assembly language, you should have some basic knowledge about the underlying processor and system organization. Part II provides this background on computer organization. Chapter 2 introduces the digital logic circuits. The next chapter gives details on memory organization. Chapter 4 describes the Intel IA-32 architecture.

Part III covers the topics related to Linux installation and usage. Chapter 5 gives detailed information on how you can install the Fedora Core Linux provided on the accompanying DVD-ROMs. It also explains how you can make your system dual bootable so that you can select the operating system (Windows or Linux) at boot time. Chapter 6 gives a brief introduction to the Linux operating system. It gives enough details so that you feel comfortable using the Linux operating system. If you are familiar with Linux, you can skip this chapter.

Part IV also consists of two chapters. It deals with assembling and debugging assembly language programs. Chapter 7 gives details on the NASM assembler. It also describes the I/O routines developed by the author to facilitate assembly language programming. The next chapter looks at the debugging aspect of program development. We describe the GNU debugger (`gdb`), which is a command-line debugger. This chapter also gives details on Data Display Debugger (DDD), which is a nice graphical front-end for `gdb`. Both debuggers are included on the accompanying DVD-ROMs.

After covering the setup and usage details of Linux and NASM, we look at the assembly language in Part V. This part introduces the basic instructions of the assembly language. To facilitate modular program development, we introduce procedures in the third chapter of this part. The remaining chapters describe the addressing modes and other instructions that are commonly used in assembly language programs.

Part VI deals with advanced assembly language topics. It deals with topics such as string processing, recursion, floating-point operations, and interrupt processing. In addition, Chapter 21 explains how you can interface with high-level languages. By using C, we explain how you can call assembly language procedures from C and vice versa. This chapter also discusses how assembly language statements can be embedded into high-level language code. This process is called inline assembly. Again, by using C, this chapter shows how inline assembly is done under Linux.

The last part consists of five appendices. These appendices give information on number systems and character representation. In addition, Appendix D gives a summary of the IA-32 instruction set. A comprehensive glossary is given in Appendix E.

Acknowledgments

I want to thank Wayne Wheeler, Editor and Ann Kostant, Executive Editor at Springer for suggesting the project. I am also grateful to Wayne for seeing the project through.

My wife Sobha and daughter Veda deserve my heartfelt thanks for enduring my preoccupation with this project! I also thank Sobha for proofreading the manuscript. She did an excellent job!

I also express my appreciation to the School of Computer Science at Carleton University for providing a great atmosphere to complete this book.

Feedback

Works of this nature are never error-free, despite the best efforts of the authors and others involved in the project. I welcome your comments, suggestions, and corrections by electronic mail.

Ottawa, Canada
January 2005

Sivarama P. Dandamudi
`sivarama@scs.carleton.ca`
`http://www.scs.carleton.ca/~sivarama`