Linux Assembly Programming

Konstantin Boldyshev

konst@linuxassembly.org

Brian Raiter

breadbox@muppetlabs.com

H-Peter Recktenwald

phpr@snafu.de

Paul Furber

m@verick.co.za

Linux Assembly Programming

by Konstantin Boldyshev, Brian Raiter, H-Peter Recktenwald, and Paul Furber

0.1 Edition
Published $Date: 2001/04/01 07:35:02 $

Table of Contents

Preface

1. Why read this book?

2. The book’s audience

3. Foreword

Chapter 1. Linux OS

Chapter 2. Linux and Assembly

2.1. Why using assembly?

2.2. Common myths about unix and assembly language

2.3. When assembly is needed

2.4. When assembly is useless

Chapter 3. Assemblers

3.1. nasm

3.2. gas

3.3. other tools you need

Chapter 4. System calls

4.1. What is a system call

4.2. View from the kernel side

4.3. View from the userland

4.4. Using system calls

Chapter 5. ELF Files

5.1. introduction/overview

5.2. sections and segments

5.3. symbols and strings

5.4. relocation records

5.5. dynamic linking data

5.6. GOT and PLT

5.7. putting it all together

5.8. other kinds of ELF files: object files and libraries

5.9. squeezing it all down

Chapter 6. Assembly programming

6.1. Two ways to go

6.2. Source code layout

6.3. Compiling a program

6.4. Debugging

Chapter 7. C and Assembly

7.1. Interfacing C code from assembly

7.2. Interfacing assembly code from C

7.3. gcc inline assembly

7.4. Optimizing C code with assembly

Chapter 8. Assembly fun

8.1. Startup process details

8.2. Treating command line

8.3. Writing in a portable way: is it possible?

8.4. Optimization issues

8.5. Tips and tricks

8.6. Frequently asked questions

Chapter 9. Fit it in a hand

This chapter will include a lot of source code examples, and is intended to be a tour on Linux IA-32 assembly programming.

9.1. Description of distribution parts

9.2. Rewriting usual utils in assembly

9.3. Implementing libc in assembly

Appendix A. System call list

Appendix B. Porting DOS code to Linux

Appendix C. Graphics programming in Linux

Appendix D. References