

# 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 \$

Copyright © 2001 by authors

# Table of Contents

<b>Preface</b> .....	<b>5</b>
1. Why read this book?.....	5
2. The book's audience.....	5
3. Foreword.....	5
<b>1. Linux OS</b> .....	<b>6</b>
<b>2. Linux and Assembly</b> .....	<b>7</b>
2.1. Why using assembly?.....	7
2.2. Common myths about unix and assembly language .....	7
2.3. When assembly is needed .....	7
2.4. When assembly is useless .....	7
<b>3. Assemblers</b> .....	<b>8</b>
3.1. nasm .....	8
3.2. gas.....	8
3.3. other tools you need .....	8
<b>4. System calls</b> .....	<b>9</b>
4.1. What is a system call .....	9
4.2. View from the kernel side.....	9
4.3. View from the userland .....	9
4.4. Using system calls.....	9
<b>5. ELF Files</b> .....	<b>10</b>
5.1. introduction/overview.....	10
5.2. sections and segments .....	10
5.3. symbols and strings .....	10
5.4. relocation records .....	10
5.5. dynamic linking data .....	10
5.6. GOT and PLT .....	10
5.7. putting it all together .....	10
5.8. other kinds of ELF files: object files and libraries .....	10
5.9. squeezing it all down.....	10
<b>6. Assembly programming</b> .....	<b>11</b>
6.1. Two ways to go.....	11
6.2. Source code layout .....	11
6.3. Compiling a program .....	11
6.4. Debugging .....	11
<b>7. C and Assembly</b> .....	<b>12</b>
7.1. Interfacing C code from assembly.....	12

7.2. Interfacing assembly code from C.....	12
7.3. gcc inline assembly .....	12
7.4. Optimizing C code with assembly .....	12
<b>8. Assembly fun .....</b>	<b>13</b>
8.1. Startup process details .....	13
8.2. Treating command line.....	13
8.3. Writing in a portable way: is it possible?.....	13
8.4. Optimization issues .....	13
8.5. Tips and tricks .....	13
8.6. Frequently asked questions .....	13
<b>9. Fit it in a hand .....</b>	<b>14</b>
9.1. Description of distribution parts .....	14
9.2. Rewriting usual utils in assembly.....	14
9.3. Implementing libc in assembly .....	14
<b>A. System call list .....</b>	<b>15</b>
<b>B. Porting DOS code to Linux .....</b>	<b>16</b>
<b>C. Graphics programming in Linux.....</b>	<b>17</b>
<b>D. References .....</b>	<b>18</b>

# **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**

