

<<x86/x64体系探索及编程>>

图书基本信息

书名：<<x86/x64体系探索及编程>>

13位ISBN编号：9787121181764

10位ISBN编号：7121181762

出版时间：2012-10

出版时间：电子工业出版社

作者：邓志

页数：813

字数：1000000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<x86/x64体系探索及编程>>

### 内容概要

《x86/x64体系探索及编程》是对Intel手册所述处理器架构的探索和论证。全书分五大部分，对多个方面对处理器架构相关的知识进行了梳理介绍。书中每个章节都有相应的测试实验，所运行的实验例子都可以在真实的机器上执行。通过阅读本书，读者应能培养自己动手实验的能力。如果再有一些OS方面的相关知识，基本上就可以写出自己简易的OS核心。

## <<x86/x64体系探索及编程>>

### 作者简介

邓志，1977年生于广东，在银行工作十余年，现自由职业者。  
对计算机有一股热情和蛮劲，善于思考，特别喜欢琢磨底层架构。  
熟悉C语言，并且精通x86/x64平台的汇编语言与机器指令系统，能用汇编写简易的OS核心。

## &lt;&lt;x86/x64体系探索及编程&gt;&gt;

## 书籍目录

## 第一篇 x86基础

## 第1章 数与数据类型

## 1.1 数

## 1.1.1 数字

## 1.1.2 二进制数

## 1.1.3 二进制数的排列

## 1.1.4 十六进制数

## 1.1.5 八进制数与十进制数

## 1.2 数据类型

## 1.2.1 integer数

## 1.2.2 floating-point数 9

## 1.2.3 real number (实数) 与NaN (not a number)

## 1.2.4 unsupported编码值

## 1.2.5 浮点数精度的转换

## 1.2.6 浮点数的溢出

## 1.2.7 BCD码

## 1.2.8 SIMD数据 21

## 第2章 x86/x64编程基础

## 2.1 选择编译器

## 2.2 机器语言

## 2.3 Hello world

## 2.3.1 使用寄存器传递参数

## 2.3.2 调用过程

## 2.3.3 定义变量

## 2.4 16位编程、32位编程, 以及64位编程

## 2.4.1 通用寄存器

## 2.4.2 操作数大小

## 2.4.3 64位模式下的内存地址

## 2.4.4 内存寻址模式

## 2.4.5 内存寻址范围

## 2.4.6 使用的指令限制

## 2.5 编程基础

## 2.5.1 操作数寻址

## 2.5.2 传送数据指令

## 2.5.3 位操作指令

## 2.5.4 算术指令

## 2.5.5 CALL与RET指令

## 2.5.6 跳转指令

## 2.6 编辑与编译、运行

## 第3章 编写本书的实验例子

## 3.1 实验的运行环境

## 3.2 生成空白的映像文件

## 3.2.1 使用nasm编译器生成

## 3.2.2 使用bximage工具

## 3.3 设置bochs配置文件

## &lt;&lt;x86/x64体系探索及编程&gt;&gt;

- 3.4 源代码的基本结构
- 3.5 编译源代码
- 3.6 映像文件内的组织
- 3.7 使用merge工具
  - 3.7.1 merge的配置文件
  - 3.7.2 执行merge命令
- 3.8 使用U盘启动真实机器
  - 3.8.1 使用merge工具写U盘
  - 3.8.2 使用hex编辑软件写U盘
- 3.9 编写boot代码
  - 3.9.1 LBA转换为CHS
  - 3.9.2 测试是否支持int 13h扩展功能
  - 3.9.3 使用int 13h扩展读磁盘
  - 3.9.4 最后看看load\_module()
- 3.1 总结
- 第4章 处理器的身份
  - 4.1 测试是否支持CPUID指令 67
  - 4.2 CPUID指令的术语及表达
  - 4.3 基本信息与扩展信息
  - 4.4 处理器的型号 ( family,model与stepping )
  - 4.5 最大的物理地址和线性地址
  - 4.6 处理器扩展状态信息
    - 4.6.1 探测Processor Extended State子叶 75
    - 4.6.2 Processor Extended State子叶所需内存size
    - 4.6.3 Processor Extended State的保存
    - 4.6.4 Processor Extended State的恢复
  - 4.7 处理器的特性
  - 4.8 处理器的Cache与TLB信息
  - 4.9 MONITOR/MWAIT信息
- 4.1 处理器的long mode
- 第5章 了解Flags
  - 5.1 Eflags中的状态标志位
    - 5.1.1 signed数的运算 86
    - 5.1.2 unsigned数的运算
  - 5.2 IOPL标志位 90
  - 5.3 TF标志与RF标志
  - 5.4 NT标志 95
  - 5.5 AC标志 96
  - 5.6 VM标志
  - 5.7 eflags寄存器的其他事项
- 第6章 处理器的控制寄存器
  - 6.1 CR8
  - 6.2 CR3
  - 6.3 CR0
    - 6.3.1 保护模式位PE
    - 6.3.2 x87 FPU单元的执行环境
    - 6.3.3 CR0.PG控制位 108

## &lt;&lt;x86/x64体系探索及编程&gt;&gt;

- 6.3.4 CR0.CD与CR0.NW控制位
- 6.3.5 CR0.WP控制位 110
- 6.3.6 CR0.AM控制位
- 6.4 CR4
  - 6.4.1 CR4.TSD与CR4.PCE控制位
  - 6.4.2 CR4.DE与CR4.MCD控制位
  - 6.4.3 CR4.OSFXSR控制位
  - 6.4.4 CR4.VMXE与CR4.SMXE控制位
  - 6.4.5 CR4.PCIDE与CR4.SMEP控制位
  - 6.4.6 CR4.OSXSAVE控制位
  - 6.4.7 CR4中关于页的控制位
- 6.5 EFER扩展功能寄存器
- 第7章 MSR
  - 7.1 MSR的使用
  - 7.2 MTRR
    - 7.2.1 Fixed-range区域的映射
    - 7.2.2 MTRR的功能寄存器
  - 7.3 MSR中对特殊指令的支持
    - 7.3.1 支持sysenter/sysexit指令的MSR
    - 7.3.2 支持syscall/sysret指令的MSR
    - 7.3.3 支持swaps指令的MSR 127
    - 7.3.4 支持monitor/mwait指令的MSR
  - 7.4 提供processor feature管理
  - 7.5 其他未列出来的MSR
  - 7.6 关于MSR一些后续说明
- 第二篇 处理器的工作模式
- 第8章 实地址模式
  - 8.1 真实的地址
  - 8.2 real mode的编址
  - 8.3 real mode的状态
  - 8.4 段基址的计算
  - 8.5 第1条执行的指令
  - 8.6 实模式下的执行环境
  - 8.7 实模式下的IVT
  - 8.8 突破64K段限
  - 8.9 A20地址线
- 第9章 SMM系统管理模式探索
  - 9.1 进入SMM
  - 9.2 SMM的运行环境 141
    - 9.2.1 SMRAM区域
    - 9.2.2 SMM执行环境的初始化
    - 9.2.3 SMM下的operand与address
    - 9.2.4 SMM下的CS与EIP
    - 9.2.5 SMM下的SS与ESP
  - 9.3 SMM里的中断
  - 9.4 SMI的Back-to-Back响应
  - 9.5 SMM里开启保护模式 147

## &lt;&lt;x86/x64体系探索及编程&gt;&gt;

- 9.6 SMM的版本 148
- 9.7 I/O指令的重启及Halt重启
- 9.8 SMM的退出 152
- 9.9 SMBASE的重定位
- 9.1 SMI处理程序的初始化
- 9.11 SMM的安全
  - 9.11.1 芯片组的控制
  - 9.11.2 处理器对SMRAM空间的限制
  - 9.11.3 cache的限制
- 9.12 测试SMI处理程序
- 第10章 x86/x64保护模式体系（上）
  - 10.1 x86/x64的权限
  - 10.2 保护模式下的环境
    - 10.2.1 段式管理所使用的资源
    - 10.2.2 paging分页机制所使用的资源
  - 10.3 物理地址的产生 166
  - 10.4 段式管理机制
    - 10.4.1 段式内存管理
    - 10.4.2 段式的保护措施
  - 10.5 段式管理的数据结构 169
    - 10.5.1 Segment Selector（段选择子）
    - 10.5.2 Descriptor Table（描述符表）
    - 10.5.3 Segment Selector Register（段寄存器）
    - 10.5.4 Segment Descriptor（段描述符）
    - 10.5.5 LDT描述符与LDT 258
  - 10.6 开启保护模式
    - 10.6.1 初始化GDT
    - 10.6.2 初始化IDT
    - 10.6.3 切换到保护模式
- 第11章 x86/x64保护模式体系（下）
  - 11.1 物理页面
    - 11.1.1 处理器的最高物理地址（MAXPHYADDR）
    - 11.1.2 物理页面的大小
    - 11.1.3 页转换模式（Paging Mode） 268
  - 11.2 paging机制下使用的资源 270
    - 11.2.1 寄存器
    - 11.2.2 CPUID查询leaf
    - 11.2.3 寄存器的控制位
    - 11.2.4 页转换表资源
  - 11.3 32位paging模式（non-PAE模式）
    - 11.3.1 CR3结构
    - 11.3.2 32位paging模式下的PDE结构
    - 11.3.3 使用32位paging
  - 11.4 PAE paging模式 282
    - 11.4.1 在Intel64下的CR3与PDPTE寄存器
    - 11.4.2 在AMD64下的CR3
    - 11.4.3 PAE paging模式里的PDPTE结构

<<x86/x64体系探索及编程>>

- 11.4.4 PAE paging模式里的PDE结构
- 11.4.5 PAE paging模式里的PTE结构
- 11.4.6 使用和测试PAE paging模式 288
- 11.4.7 使用和测试Execution Disable功能
- 11.5 IA-32e paging模式
  - 11.5.1 IA-32e paging模式下的CR3
  - 11.5.2 IA-32e paging模式下的PML4E结构
  - 11.5.3 IA-32e paging模式下的PDPTE结构
  - 11.5.4 IA-32e paging模式下的PDE结构
  - 11.5.5 IA-32e paging模式下的PTE

## 章节摘录

版权页：插图：在IA-32e paging模式下，但CR4.PCIDE=0，即未开启PCID功能时，使用默认的PCID值。

默认的PCID值为000H，因此在上述情况下，处理器只维护000H编号的TLB和paging-structure cache，实际效果等同于在legacy处理器上未实现PCID功能。

更新PCID值 当执行mov CR3, reg64指令对CR3进行刷新时，TLB和paging-structure cache的失效依赖于CR3（63）位，如下面的代码所示。

mov rax, PML4T\_BASE | 0 × 1 ; PCID=01值 mov cr3, rax ; 更新CR3 这个代码是在CR4.PCIDE=1的前提下，使用了PCID值为1去更新CR3，并且CR3（63）=0，表明需要更新TLB及paging-structure cache，那么这时候指令对TLB和paging-structure cache有下面几方面的情形。

使原来PCID为001H编号的TLB无效，即刷新TLB。

使原来PCID为001H编号的paging-structure cache无效，即刷新paging-structure cache。

对global page无影响，不会刷新global page。

对其他PCID编号的TLB和paging-structure cache无影响，不会刷新其他PCID编号的TLB和paging-structure cache内容。

因此，处理器会保留其他PCID编号的virtual address space在TLB及paging-structure cache的内容，即virtual address的page及table entry。

## 媒体关注与评论

在学习x86汇编语言的过程中，总会遇到这样一种情况：基础的指令和架构已经学完，驱动或者应用也会开发了，但想要再进一步发掘处理器的新增指令集以及新特征，却发现参考资料只有Intel的指令手册，每条指令寥寥数语的说明文字对于了解复杂的新特征根本是杯水车薪。

现在，本书以详尽的示例带领读者探索这部分内容，全面深入地为读者展现了x86处理器的高级特征。罗云彬畅销书《琢石成器&mdash;&mdash;Windows环境下32位汇编语言程序设计》作者这本书真正是让我眼前一亮。

到目前为止，这是我见过的对x86处理器介绍得最详尽又最具实践指导意义的书。

我如果学习的话，一定会选择这本书。

很显然，在实践中解决困难，应用所学知识的乐趣，是任何高大全的课程所无法比拟的。

如果耐心地将这本书上的内容读过，将作者提供的例子一一运行过，我相信对x86处理器的知识，必定会了然于胸。

谭文 畅销书《天书夜读&mdash;&mdash;从汇编语言到 Windows 内核编程》《寒江独钓&mdash;&mdash;Windows 内核安全编程》作者

## <<x86/x64体系探索及编程>>

### 编辑推荐

《x86/x64体系探索及编程》是对x86处理器介绍得最详尽又最具实践指导意义的一本书。

## 名人推荐

在学习x86汇编语言的过程中，总会遇到这样一种情况：基础的指令和架构已经学完，驱动或者应用也会开发了，但想要再进一步发掘处理器的新增指令集以及新特征，却发现参考资料只有Intel的指令手册，每条指令寥寥数语的说明文字对于了解复杂的新特征根本是杯水车薪。

现在，本书以详尽的示例带领读者探索这部分内容，全面深入地为读者展现了x86处理器的高级特征。

——罗云彬，畅销书《琢石成哭—Windows环境下32位汇编语言程序设计》作者 这本书真正是让我眼前一亮。

到目前为止，这是我见过的对x86处理器介绍得最详尽又最具实践指导意义的书。

我如果学习的话，一定会选择这本书。

很显然，在实践中解决困难，应用所学知识的乐趣，是任何高大全的课程所无法比拟的。

如果耐心地将这本书上的内容读过，将作者提供的例子一一运行过，我相信对x86处理器的知识，必定会了然于胸。

——谭文，畅销书《天书夜读—从汇编语言到Windows内核编程》《寒江独钓—Windows内核安全编程》作者

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>