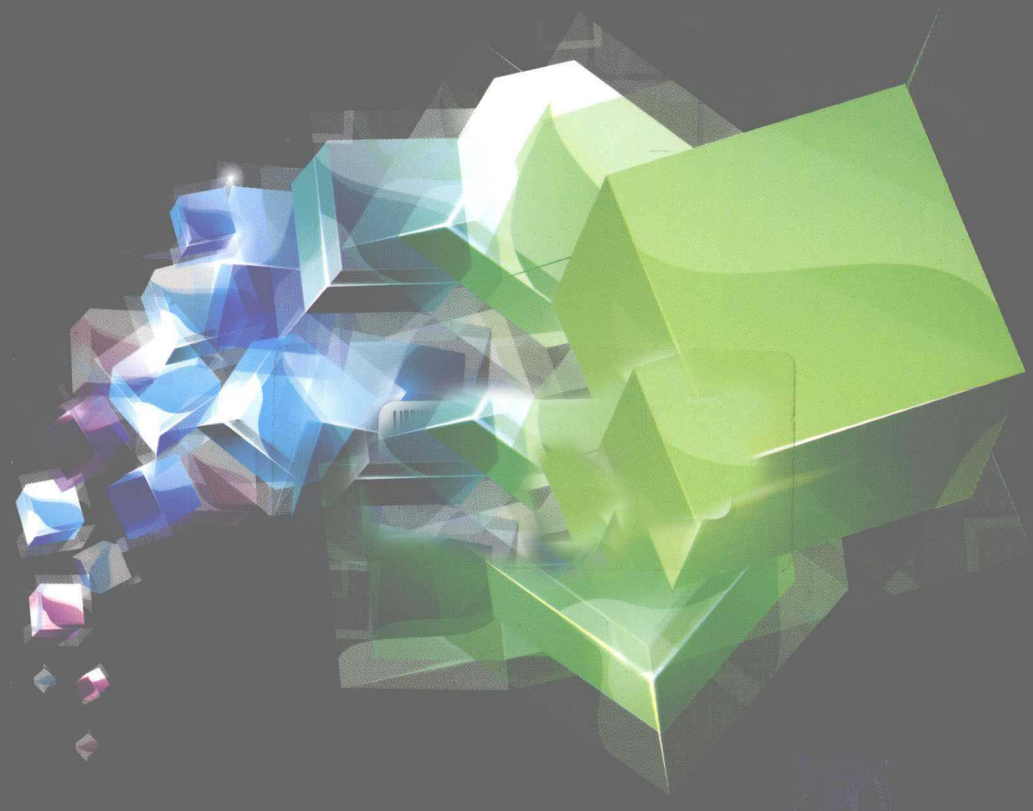
全球首本深入源代码层面剖析硬件虚拟化技术编程细节的图书
国家自然科学基金/IBM大学合作项目书籍出版计划资助项目

于淼 戚正伟 等编著

NewBluePill

深入理解硬件虚拟机



清华大学出版社

NSFC

面剖析硬件虚拟化技术编程细节的图书
大学合作项目书籍出版计划资助项目

NewBluePill
深入理解硬件虚拟机

内 容 简 介

本书是国内外第一本基于源代码详细分析 Intel 和 AMD 硬件虚拟化实现细节的书籍。本书首先串讲 NewBluePill 的启动、运行和卸载场景，之后详细讲解 NewBluePill 的各个组成模块。本书介绍 NewBluePill 的各个重要数据结构，重点在于引导读者去探索 NewBluePill 及硬件虚拟化，因此本书并不具体剖析代码中涉及的细节知识。本书附录提供了本书相关技术，包括截止到定稿时有影响力的项目和项目分析，以及相关论文和文献。

本书可以作为高校计算机相关专业师生参考读物，也可以作为硬件虚拟化技术从业人员的参考资料。

图书在版编目（CIP）数据

NewBluePill：深入理解硬件虚拟机 / 于淼，戚正伟等编著. —北京：清华大学出版社，2011.6
ISBN 978-7-302-24938-2

Ⅰ. ①N…　Ⅱ. ①于…　②戚…　Ⅲ. ①虚拟处理机　Ⅳ. TP338

中国版本图书馆 CIP 数据核字（2011）第 040534 号

责任编辑：夏兆彦
责任校对：徐俊伟
责任印制：杨　艳
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**邮　购**：010-62786544
投稿与读者服务：010-62795954，jsjjc@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：清华大学印刷厂
装 订 者：三河市李旗庄少明装订厂
经　　销：全国新华书店
开　　本：185×230　**印　张**：11.25　**字　数**：260 千字
版　　次：2011 年 6 月第 1 版　　**印　　次**：2011 年 6 月第1 次印刷
印　　数：1～3500
定　　价：29.00 元

产品编号：039304-01

序

本书是国内外第一本基于源代码详细分析 Intel 和 AMD 硬件虚拟化实现细节的书籍。

硬件虚拟化技术由 IBM 提出，诞生于 20 世纪 70 年代，然而，直到 2005 年底和 2006 年初 Intel 和 AMD 才相继推出各自的硬件虚拟化产品，此时该技术才开始快速推广到市场中，同时学术界针对其进行的相关研究才逐渐变得热门。由于问世时间较短，因此相关的软件产品和组件很少，只有诸如 VMWare、Xen、Hyper-V 等虚拟机软件利用了相关的技术，甚至于 Intel 和 AMD 旗下最新的产品也尚未实现硬件虚拟化技术的所有特性。

硬件虚拟化技术有着广泛的前景，不仅在虚拟化领域，在计算机安全、可信计算、调试等领域都有很大的发展空间。即便在当下最热门的云计算领域，硬件虚拟化技术也是其基础技术之一。

笔者学习研究硬件虚拟化技术始于 2008 年底，一方面为硬件虚拟化技术的能力所深深吸引，另一方面却发现无论是国内还是国外，没有一本详细深入讲解硬件虚拟化的书籍，材料无非只有 Intel 和 AMD 的手册。对一个初学者而言，过于详细的技术介绍反而成了掌握技术的最大障碍，同时由于硬件虚拟化技术引入了新的指令，在学习过程中不可避免地需要掌握 C 语言、汇编甚至机器码的开发技巧，这一切也让笔者在初涉该领域时颇感头痛。

随着对硬件虚拟化技术的深入了解和一系列相关项目研究，觉得很有必要推广硬件虚拟化技术，降低入门难度，让更多的人了解它并开发利用它，尽管在此过程中，市面上出现了一些介绍硬件虚拟化技术的书籍，但它们只注重于介绍技术本身，或者作为介绍虚拟化技术的补充部分，缺少结合实际代码的分析，因此很难真正地将这一技术推广。鉴于此，笔者决定以介绍硬件虚拟化技术为主题，选定 NewBluePill 项目作为研究素材完成此书。之所以选定该项目，是因为它体积较小，逻辑清晰，是 CPU Rootkit 的创始之作，以硬件虚拟化技术为基础，同时包含很多新的想法。因此笔者使用该项目全景展现从入门开始的探索学习过程，也正是这个过程让笔者自身受益匪浅，从一名仅略懂 x86 和 ARM 汇编入门知识的嵌入式学习者成为了一名 x86、x64 平台硬件虚拟化技术的研究者，在这个过程中还学会了 Windows 驱动开发。所以阅读本书不要求读者有多深的相关知识，由浅入深的介绍和探索研究将一步一步带领读者进入硬件虚拟化技术的世界。

为了更好地让读者掌握书中内容，推荐读者结合 NewBluePill 项目源代码阅读本书。编写本书时取自当时的 0.32 版，当读者看到本书时，可能已不是这个版本，推荐读者尽量下载这个版本的源代码，因为书中所出现代码行号均是依据此版本。同时设计了若干实验穿插于其中，在书的最后还设计了几个大型的有挑战性的实验，读者可以在理解相应章节内容后，结合实验亲身感受硬件虚拟化技术。

本书的主要部分首先串讲 NewBluePill 的启动、运行和卸载场景，之后详细讲解 NewBluePill 的各个组成模块，通过这样的安排，既便于初学者对硬件虚拟化技术有大体印象，熟悉开发逻辑，又满足了需要深入理解 NewBluePill 的读者对模块细节的需要。同时，本书强调介绍 NewBluePill 的各个重要数据结构，好的数据结构是程序的灵魂，一个定义优秀的数据结构能够使人立刻读懂其中的思路，所以着重于介绍其中所涉及的数据结构，力求用最短的篇幅说明 NewBluePill 的世界。但是读者一定要知道，本书力在引导读者去探索 NewBluePill 及硬件虚拟化，因此本书并不具体到代码中涉及的每个角落。

对于有志于或正在从事硬件虚拟化技术相关系统设计或实现的读者，笔者建议在阅读本书后一定要关注附录部分，其中涉及了有联系的技术，截止到初稿定稿时为止的有影响力的项目和项目分析，以及相关论文和文献。一个项目的成功总是由于它站在了另一个成功项目的肩膀上，本书的另一个目的，就是通过提供对这些已有项目的介绍，给开发者提供大量素材，对于需要快速开发的开发者来说，只需通过阅读本书最后的项目介绍，找到适合于自己的项目，在许可证权限内剪裁相关文件即可，这样就省去许多到互联网上搜索相关项目的时间。而对于研究者，也可以通过阅读本书获得大量的参考文献来源。

希望读者在每读完一部分后能做个小结，记录下重要的数据结构和 NewBluePill 对该模块的处理流程。同时，建议读者反复阅读此书，对于初学者，建议在第一遍阅读本书时先跳过“深入 HEV 技术细节”相关知识部分，在阅读 NewBluePill 流程分析时再对照该部分的叙述，结合代码明白其原因和实现方法，而在再次阅读本书或利用本书进行硬件虚拟化技术应用开发时着重关注这一部分，获得最具体的细节分析。我们相信，读者在阅读此书的过程中，一定会有一种在导游陪同下探索一个新的岛屿，最后站在一个高峰俯瞰全岛的感觉。

在编写过程中，特别感谢戚正伟和徐昊所给予的指导帮助和对本书的审核工作，如果没有他们大量无私的帮助，笔者无法在短时间内掌握 NewBluePill 项目的核心思想并如此深入地探索研究虚拟化技术，他们对新技术敏锐的眼光以及应用新技术为我国可信计算研究和产品做出的贡献和决策，令人钦佩。同时，对上海交通大学虚拟机项目组在编写本书期间所给的鼓励、支持和审核工作表示感谢，他们在本书的编写过程中提供了大量的硬件设备以供实验，并且通过与他们的讨论使得笔者更快地理解了 VT 下

NewBluePill 很多模块的处理过程。

此外，还要感谢董浩亮、李秉昱、徐昊、韩志坤、星月王者、金毅、顾玉婷、夏鸣远、俞培杰、朱闵、林芊，在编写过程中，他们利用业余时间贡献了很多很好的思路和资料，同时他们也为本书部分实验和过程分析付出了大量的辛勤劳动，在此一并表示由衷的感谢。

虽然在成稿前后经过了反复修改，发现了很多错误并予以更正，但是我们仍然无法保证本书已经没有错误。我们只能说，所有已发现的错误都已改正，读者在阅读过程中发现任何错误，欢迎通过清华大学出版社与我们联系，请广大读者批评指正。

于　淼

2010 年 12 月

前言

FOREWORD

本书是专门为学习研究硬件虚拟化技术的读者准备的。本书入门简单，具备 x86 基础和嵌入式设备程序开发经验的读者即可阅读本书，通过阅读本书，读者能够熟悉硬件虚拟化技术并能在其基础上开发相关应用程序。对于研究者而言，本书附录中出现的论文和书籍提供了丰富的引用资源；而对于开发者来说，本书中对相关项目的分析和相关技术的介绍提供了丰富的开发素材。同时对所有读者，本书结合 NewBluePill 项目源代码和穿插实验将快速带您进入硬件虚拟化技术的世界。

本书结构

本书首先介绍已有硬件虚拟化技术，并大致讲解它们的工作原理和异同。随后通过讲解 NewBluePill 源代码来展示硬件虚拟化技术的编程方法和带来的神奇之处。第 2 章着重展示 NewBluePill 的效果，给读者展示对 NewBluePill 最直接的认识。从第 4 章开始，本书将逐一分析 NewBluePill 的各个模块源代码。在最后，本书给出 4 个实验，用于读者检验自己对硬件虚拟化编程的掌握情况。同样本书也给出一系列有关项目和参考文献，用于读者在未来进行更深入的学习。

实验部分

本书的实验部分需要读者在被调试机器上安装 64 位 Windows 操作系统，并且在调试机上安装 Windbg 调试工具。本书的各章节内的实验均出现在“实验”标题方框中，其中包含了完成实验的具体步骤、示例输出和简要分析。建议读者按照书中步骤完成各个实验，以加深印象。在本书最后还有几个大型实验，只是提供了完成实验需要注意的地方和部分示例输出，并未随书提供相关源代码，鼓励读者独立完成实验。

未涵盖课题

硬件虚拟化技术包含了很多方面，本书由于是通过讲解 NewBluePill 项目源代码来讨论硬件虚拟化技术的，因此并未涉及某些硬件虚拟化技术的细节部分，如 VT-d、EPT 的具体细节，以及 Intel® TXT 技术的使用详情，这些部分在诸如 VMWare Workstation 6.5.1 和 Xen 较高版本中都有所体现。

如果读者确实需要了解这些方面的内容或者需要更深地探索硬件虚拟化技术，可参

考 Intel 和 AMD 相关文档描述。

注意事项

本书所分析的 NewBluePill 项目源代码是 0.32 公开版本，作者 Invisible Things Lab，所使用 Intel 手册是 2009 年 3 月版，AMD 手册是 2007 年 9 月版（Rev 3.14）。在读者阅读本书时，很可能已经可以下载到更新的源代码和手册。推荐读者下载上文所列代码和手册版本，以保证与书中内容和引用统一。

此外，由于 NewBluePill 是研究型项目且运行在系统底层，在运行时难免存在漏洞，读者在做相关实验时务必提前关闭任何打开的文档，以避免可能由于系统崩溃造成的数据丢失，如果确实造成损害，NewBluePill 项目作者和本书作者对此不承担任何责任。

书中表示法

本书涉及的表示方法如下。

Windows Internals, 4th Edition	书名
Hypervisor	英文名词
http://www.invisiblethingslab.com	超级链接
`HvmSubvertCpu()`	书中代码及函数名
Important	重点注意
Note	提示
Think	思考题

支持

对于在本书中遇到的任何技术性错误或说明不准确的地方，或者更深层次讨论硬件虚拟化和相关技术，我们欢迎您发送信件到 superymkvmm@hotmail.com，或者访问 http://www.cnblogs.com/superymk/获取最新动态并留下您的评价。

目 录

CONTENTS

第 3 篇 深入篇

第 4 篇 实验篇

第1章 概 述

本章先介绍一些贯穿全书的概念，比如 Hypervisor、VT-x、VT-d、SVM 等，然后简略介绍 NewBluePill 项目背景及其所采用的硬件虚拟化技术。

本章只是介绍这些技术大致的轮廓，详细内容会在后面各章节中逐一介绍。

1.1 虚拟化技术概述

1.1.1 虚拟化的历史

在讨论 Hypervisor 之前我们首先谈谈虚拟概念。虚拟（virtualization）是指对计算机资源的抽象，一种常用的定义是："虚拟就是这样的一种技术，它隐藏了系统、应用和终端用户赖以交互的计算机资源的物理性的一面，最常用的方法就是把单一的物理资源转化为多个逻辑资源，当然也可以把多个物理资源转化为一个逻辑资源（这在存储设备和服务器上很常见）。"

实际上，虚拟技术早在 20 世纪 60 年代就已出现，最早由 IBM 提出，并且应用于计算技术的许多领域，模拟的对象也多种多样，从整台主机到一个组件，其实打印机就可以看成是一直在使用虚拟化技术的，总是有一个打印机守护进程运行在系统中，在操作系统看来，它就是一个虚拟的打印机，任何打印任务都是与它交互，而只有这个进程才知道如何与真正的物理打印机正确通信，并进行正确的打印管理，保证每个任务按序完成。

长久以来，用户常见的都是进程虚拟机，也就是作为已有操作系统的一个进程，完全通过软件的手段去模拟硬件，软件再翻译内存地址的方法实现物理机器的模拟，比如较老版本的 VMWare、VirtualPC 软件都属于这种。

在 2005 年和 2006 年，Intel 和 AMD 都开发出了支持硬件虚拟技术的 CPU，也就是在此时，x86 平台才真正有可能实现完全虚拟化[1]。2007 年年初，Intel 还进一步地发布了 VT-d 技术规范，从而在硬件上支持 I/O 操作的虚拟化。随着硬件虚拟化技术越来越广泛的采用，开发者也开始用虚拟技术来做一些其他的事情。当前硬件虚拟机（Hardware Virtual Machine，HVM）已经在虚拟机、安全、加密等领域上有所应用，例如，VMware Fusion、Parallels Desktop for Mac、Parallels Workstation 和 DNGuard HVM，随着虚拟化

1 完全虚拟化（Full Virtualization），完整虚拟底层硬件，这就使得能运行在该底层硬件上的所有操作系统和它的应用程序，也都能运行在这个虚拟机上。

办公和应用的兴起，相信虚拟化技术也会在未来得到不断的发展。

1.1.2 硬件虚拟化技术

有了虚拟技术的基本概念，下面我们谈谈硬件虚拟化技术（HEV）。硬件虚拟化技术（Hardware Enabled Virtualization，HEV）也就是在硬件层面上，更确切地说是在 CPU 里（VT-d 技术是在主板上北桥芯片支持），对虚拟技术提供直接支持，并通过这种设计提高虚拟效率、降低开发难度。在硬件虚拟化技术诞生前，在编写虚拟机的过程中，为了实现多个虚拟机上的真实物理地址隔离，需要编程实现把客户机的物理地址翻译为真实机器的物理地址。同时也需要给不同的客户机操作系统编写不同的虚拟设备驱动程序，使之能够共享同一真实硬件资源。硬件虚拟化技术则在硬件上实现了内存地址甚至于 I/O 设备的映射，因此大大简化了编写虚拟机的过程。而其硬件直接支持二次寻址和 I/O 映射的特性也提升了虚拟机在运行时的性能。[1]

在硬件虚拟化技术中，一个重要的概念就是 VMM （Virtual Machine Monitor，在本书中称为 Hypervisor），它专指在使用硬件虚拟化技术时创建出的特权层，该层提供给虚拟机开发者，用来实现虚拟硬件与真实硬件的通信和一些事件处理操作（图 1.1），因此 Hypervisor 的权限级别要大于或等于操作系统权限。

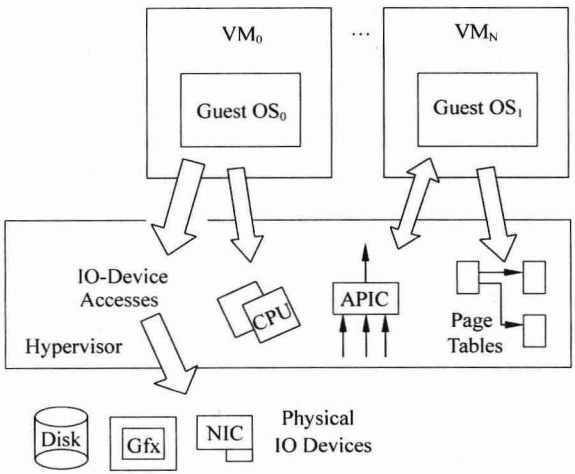


图 1.1 硬件虚拟化技术架构示意图

1 一些优化技术也在硬件中被采用，如专门用于二次寻址的 TLB，详细信息可以参考 Intel 和 AMD 的手册。

1.1.3 HEV 技术应用模型

Hypervisor 使用架构图如图 1.2 所示。

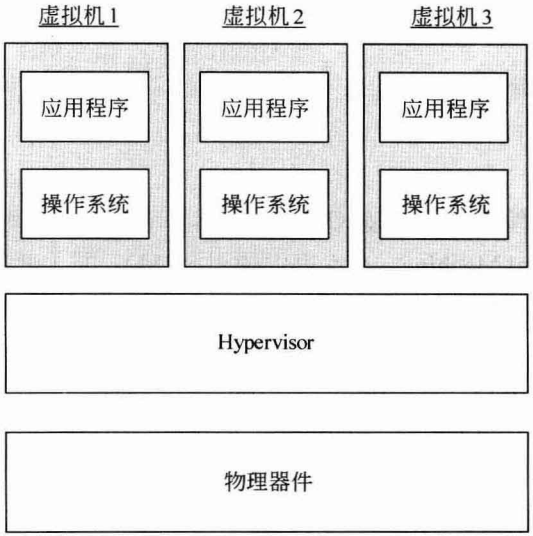


图 1.2 Hypervisor 使用架构图

前文中已经提过 Hypervisor 层的权限要大于或等于操作系统的权限。操作系统的内核态已经处在了 Ring0 特权级上，因此 Hypervisor 层实际上要运行在一个新的特权级别上，称为“Ring -1”特权级。同时需要新的指令，寄存器以及标志位去实现这个新增特权级的功能。

作为一种最佳实践方案，一般 Hypervisor 层的实现都是越简单越好。一方面，简单的实现能够尽量降低花在 Hypervisor 上的开销[1]，毕竟大多数这些开销在原先的操作系统上是不存在的。另一方面，复杂的程序实现容易引入程序漏洞，Hypervisor 也是如此，且一旦 Hypervisor 中的漏洞被恶意使用，由于其所处特权级高于操作系统，将使隐藏在其中的病毒、恶意程序很难被查出。

1.2 已有的 HEV 技术平台介绍

现今两大主要硬件厂商 Intel 和 AMD 均已推出了支持硬件虚拟化技术的产品，两者

1 关于 Hypervisor 的开销问题，后面的章节会有介绍。

大体功能和实现方法近似（意料之中，因为两家公司在过去你死我活的市场拼斗中，每次也都是实现功能和方法类似，只不过名字不同罢了）。下面简略介绍这两家公司的支持HEV技术的平台，读者可以首先对这两种平台有概念，同时作为对比，将简单介绍IBM Power架构上的虚拟化技术。在后面的章节会对Intel和AMD这两个平台虚拟化技术作更详细的描述。

1.2.1 SVM

1．概述

AMD芯片支持硬件虚拟化的技术称为AMD-V（在技术文档中也称为SVM，其全称是AMD Secure Virtual Machine，在本书中使用SVM）。其主要是通过一组能够影响到Hypervisor和Guest Machine（客户机，下文简称Guest）的中断实现的。AMD-V技术设计目标如下。

（1）引入客户机模式（Guest Mode）[1]。

（2）Hypervisor和Guest之间的快速切换。

（3）中断Guest中特定的指令或事件（Events）。

（4）DMA外部访存的保护。

（5）中断处理上的辅助并对虚拟中断（Virtual Interrupt）提供支持。

（6）新的嵌套页表用来实现地址翻译。

（7）一个新的TLB（Translation Lookaside Buffer，页表缓冲）来减少虚拟化造成的性能下降。

（8）对系统安全的支持。

① 新的客户机模式。通过VMRUN指令即可进入这种新的处理器模式，当进入客户机模式后，为了辅助虚拟化过程，一些x86汇编指令的语义会发生变化。

> **Important** 注意本书中强调处理器（Processor）和CPU的差别，文中若无特别说明，处理器指逻辑处理器（Logical Processor）。在多核时代，一个CPU上可能会有多个核，而在操作系统视角中，一个核才是一个逻辑处理器，因此通过操作系统查看的逻辑处理器数量往往大于真实CPU的数量，并且逻辑处理器才是能够运行Hypervisor的基础。

② 外部访问保护。过去，Guest可以直接访问选定的I/O设备。现在，硬件上已经实现这样的安全功能，能够阻止某个虚拟机拥有的某个设备访问其他虚拟机的内存。

1 x86上原有处理器模式包括保护模式（Protected Mode）、管理模式（SMM）、实模式（Real Mode）。

③ 中断上的支持。为了辅助中断的虚拟化，下列各项现在已经得到硬件支持，并且可以通过配置 VMCB 结构体[1]的方法使用，该结构体用于通知物理 Processor 要拦截的事件，以及在进出 Hypervisor 上下文切换时保存 Hypervisor 和 Guest 的各项寄存器。

- 拦截物理中断分发（Intercepting Physical Interrupt Delivery），发生在物理硬件上的中断能够让虚拟机发生#VMEXIT 事件，陷入 Hypervisor，从而使得 Hypervisor 可以首先处理这个中断。
- 虚中断（Virtual Interrupts），Hypervisor 能够提供给 Guest 一套虚拟的中断机制。其实现机制为，Hypervisor 会给这个 Guest 复制一份 EFLAGS.IF 用作中断屏蔽位（Interrupt Mask Bit），同时复制 APIC[2]中的中断优先级寄存器提供给 Guest，从而 Guest 就会去操纵这套假的中断机制，而不是直接去操纵物理中断。
- 共享物理 APIC AMD 的 SVM 技术能够允许多个 Guest 共享同一物理 APIC，同时又能保护这个 APIC 以免某个 Guest 不慎或恶意地在未经其他 Guest 许可的情况下，将可接收中断优先级设置为高优先级，从而清空了所有其他 Guest 的中断。

④ 被标记的 TLB（Tagged TLB）。为了降低 Guest 模式和 VMM 模式的切换开销，在 TLB 上新加了一个 ASID 标记（Address Space Identifier，地址空间描述符），这个标记可以区分 TLB 上的一块地址是 Hypervisor 范围内的地址还是 Guest 的地址，通过减少每次切换涉及的刷新项数量而加速了上下文切换速度（特指 Hypervisor 和 Guest 之间的切换速度）。

⑤ 安全方面的支持。现在提供的安全方面的支持主要是利用和 TPM 模块（Trusted Platform Module）[3]的交互，基于与安全 Hash 值的比较。

在 SVM 技术下的 Hypervisor 生命周期如图 1.3 所示。

软件开启虚拟机管理器（VMM）运行环境后，通过使用 VMRUN 指令使得目标系统正式运行在虚拟机中。当某条指令产生了#VMEXIT 事件或者虚拟机显式地发出了 VMMCALL 指令后会陷入 VMM 中。待其处理完这个事件，可以继续通过 VMRUN 指令将控制权移交回虚拟机。最后在某个时刻，Hypervisor 必须手动拆除其运行环境并恢复相关型号特定寄存器（Model Specific Register，MSR）状态， Hypervisor 才会被关闭。

---

1 VMCB 结构体，Virtual Machine Control Block，也称 VMCB 控制块，Intel 的相应结构体名称为 Virtual Machine Control Sector，VMCS。后面的章节中会有对这个结构体的详细介绍。

2 APIC，Advanced Programmable Interrupt Controller，高级可编程中断控制器，第 3 章有关于此主题内容。

3 TPM 模块会在附录 B 中介绍。