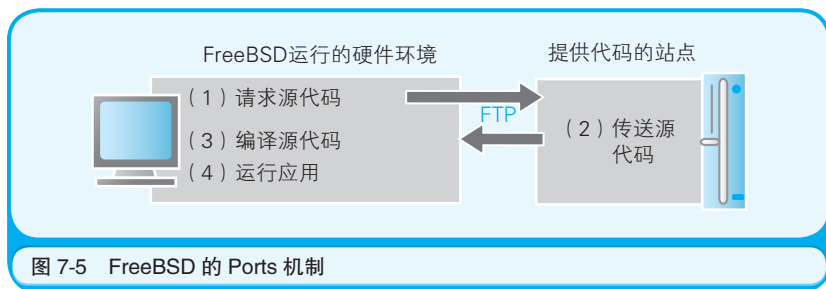


同样的本地代码无法重复利用，那么为何不直接把源代码分发给程序呢？”的确，这也是一种方法。部分 Unix 系列操作系统就对此进行了灵活应用。

Unix 系列操作系统 FreeBSD 中，存在一种名为 **Ports** 的机制。该机制能够结合当前运行的硬件环境来编译应用的源代码，进而得到可以运行的本地代码系统。如果目标应用的源代码没有在硬件上的话，Ports 就会自动使用 FTP^① 连接到相关站点来下载代码（图 7-5）。



全球很多站点都提供适用于 FreeBSD 的应用源代码。通过使用 Ports 可以利用的程序源代码，大约有 16000 种。这些代码还被按照不同的领域进行了分类整理，可以随时拿来使用。

FreeBSD 上应用的源代码，大部分都是用 C 语言来记述的。FreeBSD 等 Unix 系列操作系统中，都带有标准的 C 编译器。C 编译器可以结合 FreeBSD 的运行环境生成合适的本地代码。因而，使用 FreeBSD 的同时，肯定也会享受到 Ports 带来的益处。可以说 Ports 能够克服包含 CPU 在内的所有硬件差异的系统。而且，Ports 这个术语，表示的是 porting（移植）的意思。而根据不同的运行环境来重新调整程

① FTP（File Transfer Protocol）是连接到互联网上的计算机之间传送文件的协议。

序，一般也称为“移植”。

7.5 利用虚拟机获得其他操作系统环境

即使不通过移植，也可以使用别的方法来运行其他操作系统的应用。这里我们要介绍的方法就是利用**虚拟机**软件。笔者的计算机上就安装了 Macintosh 的“Virtual PC for Mac”^①。通过利用该虚拟机，我们就可以在 Macintosh 的 Mac 操作系统上运行 Windows 应用了。

Virtual PC for MAC 可以使 Macintosh 这一硬件变得同 AT 兼容机一样，从而能在该硬件上安装 Windows。这样一来，Windows 下的所有应用就都可以正常运行了。Windows 应用利用的是 Windows 操作系统的 API。虽然表面上是 Windows 将硬件处理为了 AT 兼容机，但由于 Virtual PC for MAC 的作用，实际上运行的是 Macintosh 这一硬件。

图 7-6 是在 PowerBook G4 这个机型（CPU 不是 x86 而是 PowerPC G4）的 Macintosh 上，通过使用 Virtual PC for MAC 起动 Windows XP 来运行 Windows 的音乐应用“BAND IN A BOX 14”的情况。可以发现，虽然运行速度有点慢，但确实能正常运行。

① Macintosh（统称为 Mac）是美国苹果公司生产的计算机。这些计算机用的是名为 Mac OS 的操作系统。Virtual PC for Mac 是美国微软的产品，需要单独购买。2006 年，美国微软终止了 Virtual PC for Mac 的开发。这是因为 Mac 采用了 Intel CPU 的缘故。这里介绍的 Virtual PC for Mac 是采用 Power PC CPU 的 Mac 上使用的软件。

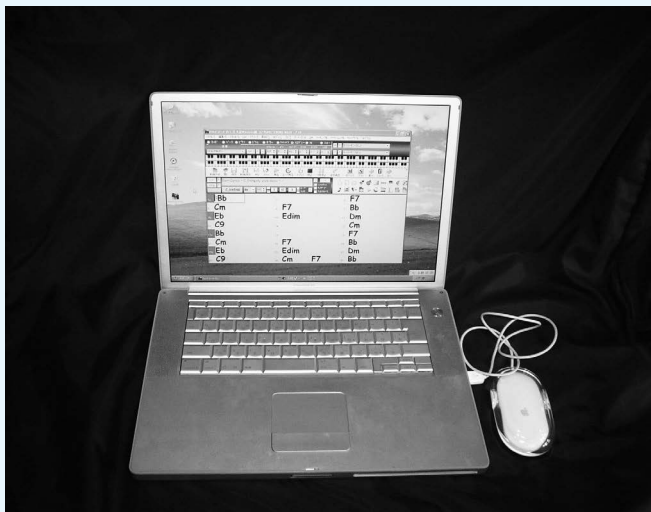


图 7-6 在 Macintosh 上运行 Windows XP 和 Windows 的应用

7.6 提供相同运行环境的 Java 虚拟机

除虚拟机的方法之外，还有一种方法能够提供不依赖于特定硬件及操作系统的程序运行环境，那就是 Java。

大家说的 Java，有两个层面的意思。一个是作为编程语言的 Java，另一个是作为程序运行环境的 Java。同其他编程语言相同，Java 也是将 Java 语法记述的源代码编译后运行。不过，编译后生成的并不是特定 CPU 使用的本地代码，而是名为**字节代码**的程序。字节代码的运行环境就称为**Java 虚拟机**（JavaVM，Java Virtual Machine）。Java 虚拟机是一边把 Java 字节代码逐一转换成本地代码一边运行的。

例如，在使用用于 AT 兼容机的 Java 编译器和 Java 虚拟机的情况

下，编译器会将程序员编写的源代码（sample.java）转换成字节代码（sample.class）。而Java虚拟机（java.exe）则会把字节代码转换成x86系列CPU适用的本地代码，然后由x86系列CPU负责实际的处理。

在程序运行时，将编译后的字节代码转换成本地代码，这样的操作方法看上去有些迂回，但由此可以实现同样的字节代码在不同的环境下运行。如果能够结合各种类型的操作系统和硬件作成Java虚拟机，那么，同样字节代码的应用就可以在任何环境下运行了（图7-7）。

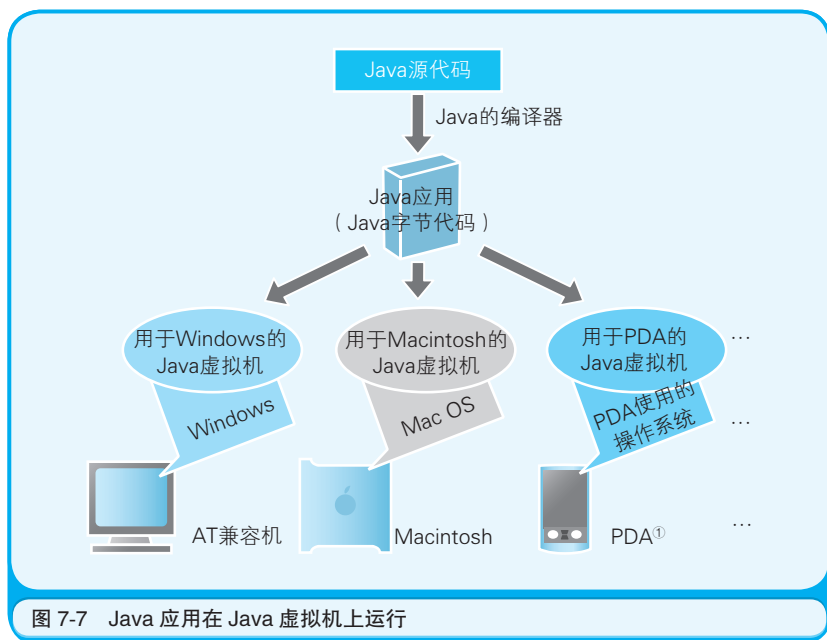


图 7-7 Java 应用在 Java 虚拟机上运行

Windows 有 Windows 专用的 Java 虚拟机，Macintosh 也有 Macintosh 专用的 Java 虚拟机。从操作系统方面来看，Java 虚拟机是一个应用，

① PDA（Personal Digital Assistant）是指可以放入手提包中的小型手持计算机。也称为“手持设备”。

而从 Java 应用方面来看, Java 虚拟机就是运行环境。虽然这样看起来 Java 虚拟机全是好处,但其实也有不少问题。其中一点就是,不同的 Java 虚拟机之间无法进行完整互换。这是因为,想让所有字节代码在任意 Java 虚拟机上都能运行是比较困难的。而且,当我们使用只适用于某些特定硬件的功能时,就会出现在其他 Java 虚拟机上无法运行,或者功能使用受限等情况。

另一点就是运行速度的问题。Java 虚拟机每次运行时都要把字节代码变换成本机代码,这一机制是造成运行速度慢的原因。为此,目前业界也在努力改善这一问题,比如把首次变换后的本地代码保存起来,第 2 次以后直接利用本地代码,或是对字节代码中处理较为费时的部分进行优化(改善生成的本地代码质量)等。

7.7 BIOS 和引导

最后对一些比较基础(和硬件相近的部分)的内容做一下补充说明。程序的运行环境中,存在着名为 BIOS(Basic Input/Output System)的系统。BIOS 存储在 ROM 中,是预先内置在计算机主机内部的程序。BIOS 除了键盘、磁盘、显卡等基本控制程序外,还有启动“引导程序”的功能。引导程序是存储在启动驱动器起始区域的小程序。操作系统的启动驱动器一般是硬盘,不过有时也可以是 CD-ROM 或软盘。

开机后, BIOS 会确认硬件是否正常运行,没有问题的话就会启动引导程序。引导程序的功能是把硬盘等记录的 OS 加载到内存中运行。虽然启动应用是 OS 的功能,但 OS 并不能自己启动自己,而是通过引导程序来启动。

Bootstrap 的原意是指靴子上部的“拔靴带”。BIOS 这样小的程序

(拔靴带)，可以带动(启动)操作系统这样的大程序(靴子)，所以由此得名(图 7-8)。虽然操作系统运行以后，程序员就不用再关注 BIOS 及引导程序了，但需要知道它们的存在。



图 7-8 小引导程序带动大操作系统

本章我们一起了解了应用程序的运行环境，并对源代码和本地代码进行了简单的说明。下一章，我们将对源代码转换到本地代码的流程，也就是“编译”进行详细说明。



第8章

从源文件到可执行文件

热身问答

阅读正文前，让我们先回答下面的问题来热热身吧。



问题

1. CPU 可以解析和运行的程序形式称为什么代码？
2. 将多个目标文件结合生成 EXE 文件的工具称为什么？
3. 扩展名为 .obj 的目标文件的内容，是源代码还是本地代码？
4. 把多个目标文件收录在一起的文件称为什么？
5. 仅包含 Windows 的 DLL 文件中存储的函数信息的文件称为什么？
6. 在程序运行时，用来动态申请分配的数据和对象的内存区域形式称为什么？

怎么样？是不是发现有一些问题无法简单地解释清楚呢？下面是笔者的答案和解析，供大家参考。

答案

1. 本地代码（机器语言代码）
2. 链接器
3. 本地代码
4. 库文件
5. 导入库
6. 堆

解析

1. 通过编译源代码得到本地代码。
2. 通过编译和链接，得到 EXE 文件。
3. 通过对源文件进行编译，得到目标文件。例如，C 语言中，将 Sample1.c 这个源文件编译后，就会得到 Sample1.obj 这个目标文件。目标文件的内容是本地代码。
4. 链接器会从库文件中抽取出必要的目标文件并将其结合到 EXE 文件中。此外，还存在一种程序运行时结合的 DLL 形式的库文件。
5. 把导入库信息结合到 EXE 文件中，这样程序在运行时就可以利用 DLL 内的函数了。
6. 堆的内存空间会根据程序的命令进行申请及释放。

本章重点

源代码完成后，就可以编译生成可执行文件了。负责实现该功能的是编译器。本章将围绕着编译器的功能，详细介绍从程序编写到运行为止的流程。首先，我们会和大家一起看一下源文件是如何通过编译转换成可执行文件的。接下来，我们会继续关注可执行文件被加载到内存后的运行机制。此外，还会对程序运行时内存上的栈及堆进行说明。由于篇幅有限，本章只介绍了用 C 语言编译器^①来编写 Windows 用的可执行文件（EXE 文件）的示例，不过其他环境及编程语言等采用的基本上是同样的机制。因此，即使不了解 C 语言的相关知识也不会有影响，这一点请大家放心。

8.1 计算机只能运行本地代码

首先，请大家看一下代码清单 8-1。这是一个用 C 语言记述的 Windows 程序。该程序运行后，会把 123 和 456 的平均值 289.5 显示在消息框^②（图 8-1）中。程序的内容并没有什么意思，这里仅仅是作为例子使用而已。

代码清单 8-1 求解平均值的程序

```
#include <windows.h>
#include <stdio.h>

// 消息框的标题
char* title = " 示例程序 1";
```

- ① 本书使用的是 Borland C++ Compiler 5.5。命令行版的 Borland C++ Compiler 5.5 可以从 Borland 的网站（<http://borland.com/jp/products/cbuilder/freecompiler.html> 2007.1）上免费下载。C++ 是在 C 语言的基础上追加相应功能而开发出来的编程语言。用 C 语言编写的源文件，也可以在 C++ 编译器上进行编译。
- ② 消息框是一个为了显示短消息而出现的小窗口。