



# 第7章

## 程序是在何种环境中运行的

### 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. 应用的运行环境，指的是什么？
2. Macintosh 用的操作系统 ( MacOS ) , 在 AT 兼容机上能运行吗？
3. Windows 上的应用，在 MacOS 上能运行吗？
4. FreeBSD 提供的 Ports，指的是什么？
5. 在 Macintosh 上可以利用的 Windows 环境模拟器称为什么？
6. Java 虚拟机的功能是什么？

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. 操作系统和计算机本身（硬件）的种类
2. 无法运行
3. 无法运行
4. 通过使用源代码来提供应用，并根据运行环境进行整合编译，从而得以在该环境下运行的机制
5. Virtual PC for Mac
6. 运行 Java 应用的字节代码

### 解析

1. 应用的运行环境通常是用类似于 Windows(OS) 和 AT 兼容机（硬件）这样的 OS 和硬件的种类来表示的。
2. 不同的硬件种类需要不同的操作系统。
3. 应用是为了在特定操作系统上运行而作成的。
4. FreeBSD 是一种 Unix 操作系统。通过在各个环境中编译 Ports 中公开的代码，就可以执行由此生成的本地代码了。
5. 模拟器是指在 Macintosh 上提供虚拟的 Windows 环境。
6. 只要分别为各个环境安装专用的 Java 虚拟机，同样的字节代码就能在各种环境下运行了。

## 本章重点

由于同一个程序能被大量用户使用，所以说程序具有很大的价值。如果将程序拿来出售的话，只要销量大，肯定就能收到非常可观的利润。而即便是自由软件（free soft）<sup>①</sup>，若是有大量用户使用的话，那也是一件让人高兴的事情。大家也都希望自己编写的程序被尽可能多的用户喜欢并使用吧。但是，如果运行环境不同，程序是无法运行的。例如，在 Macintosh 上直接运行 Windows 用的程序，基本上是无法实现的。大家都知道这是因为运行环境不同造成的。那么，运行环境不同指的是什么呢？为什么运行环境不同，应用就无法运行呢？本章将对这些问题进行解答，并介绍多个解决方法。

### 7.1 运行环境 = 操作系统 + 硬件

程序中包含着运行环境这一内容。大家手头若有有购买的应用软件的话，可以稍微观察一下它的安装包或者目录。通常在某个位置会写有“运行环境”这一项。例如，2007 Microsoft Office System（下文简称为 Office 2007）需要的运行环境，就如表 7-1 所示。从中可以看出，在表示程序的运行环境时，列出了 Operating System（操作系统）和计算机的主机（硬件）两项，由此，大家可以清楚地知道运行环境是这两者的综合。也就是说，操作系统和硬件决定了程序的运行环境。

<sup>①</sup> 自由软件一般都是免费的。用户可以从互联网上下载，或者从书、杂志等附带的 D-ROM 中获取。

表 7-1 2007 Microsoft Office system 的运行环境 (这里省略了部分内容)

|                      |   |
|----------------------|---|
| 日语版操作系统              | Microsoft Windows XP Service Pack(SP)2、Windows Server 2003 SP1 及以上版本的操作系统 |
| 计算机和 CPU (PC/AT 兼容机) | 500MHZ 以上的 CPU  |
| 内存                   | 256MB 以上的内存, 需要高速检索的情况下, 推荐使用 512MB 以上的内存                                 |
| 硬盘                   | 2GB 以上的剩余空间, 安装后, 删除硬盘上下载的安装包的话, 会稍微释放出一点空间                               |
| 显示器                  | 1024x768 以上的高解析度显示器   |
| 磁盘设备                 | CD-ROM 驱动器或者 DVD-ROM 驱动器  |

同一类型的硬件可以选择安装多种操作系统。例如, 同样的 AT 兼容机<sup>①</sup>中, 既可以安装 Windows, 也可以安装 Linux<sup>②</sup>等操作系统。正因为如此, Office 2007 的运行环境中, 把硬件和操作系统的种类这两方面内容都列了出来 (图 7-1)。不过, Windows 及 Linux 操作系统也存在多种版本。根据应用的具体情况, 有时只有在特定版本的操作系统上才能运行。

从程序的运行环境这一角度来考量硬件时, CPU 的种类是特别重要的参数。为了保证 Office 2007 的正常运行, 需要具备 Pentium 等被称为 x86<sup>③</sup>的 CPU (微处理器)。

- 
- ① AT 兼容机是指, 可以和 IBM 开发的 PC/AT 在硬件上相互兼容的计算机的总称。称为“PC/AT 兼容机”和“DOS/V 机”。现在市面上销售的大部分计算机都是 AT 兼容机。另外, IBM 现在已经把计算机事业部卖给了联想。
- ② Linux 是 1991 年赫尔辛基大学的 Linus Torvalds 开发的 Unix 系操作系统。发布后得到了很多有志者的协助, 为其追加了大量的功能。在服务端操作系统中占有比较高的比率。
- ③ 美国 Intel 的微处理器, 是按照 8086、80286、80386、80486、Pentium……这样的顺序不断升级的。因为这些型号的后面都带有 86, 所以总称为 x86。32 位处理器也称为“IA-32”。



图 7-1 操作系统和硬件共同决定应用的运行环境

CPU 只能解释其自身固有的机器语言。不同的 CPU 能解释的机器语言的种类也是不同的。例如，CPU 有 x86、MIPS、SPARC、PowerPC<sup>①</sup> 等几种类型，它们各自的机器语言是完全不同的。

机器语言的程序称为**本地代码** (native code)。程序员用 C 语言等编写的程序，在编写阶段仅仅是文本文件。文本文件（排除文字编码的问题）在任何环境下都能显示和编辑。我们称之为**源代码**。通过对源代码进行编译，就可以得到本地代码。在市面上出售的用于 Windows 的应用软件包 CD-ROM 中，收录的就不是源代码，而是本地代码<sup>②</sup>（图 7-2）。

① MIPS 是美国 MIPS 科技公司开发的 CPU。曾出现过面向 MIPS 工作站的 Windows，不过现在市面上已经不再出售了。SPARC 是美国 SUN 系统开发的 CPU。很多工作站都采用了该 CPU。PowerPC 是美国苹果、IBM、摩托罗拉共同开发的 CPU。苹果的 Power Mac 及 IBM 的工作站都采用了该 CPU。不过现在的 Mac 采用的是 Intel 的 x86 系列 CPU。

② Windows 应用程序的本地代码，通常是 EXE 文件及 DLL 文件等形式。

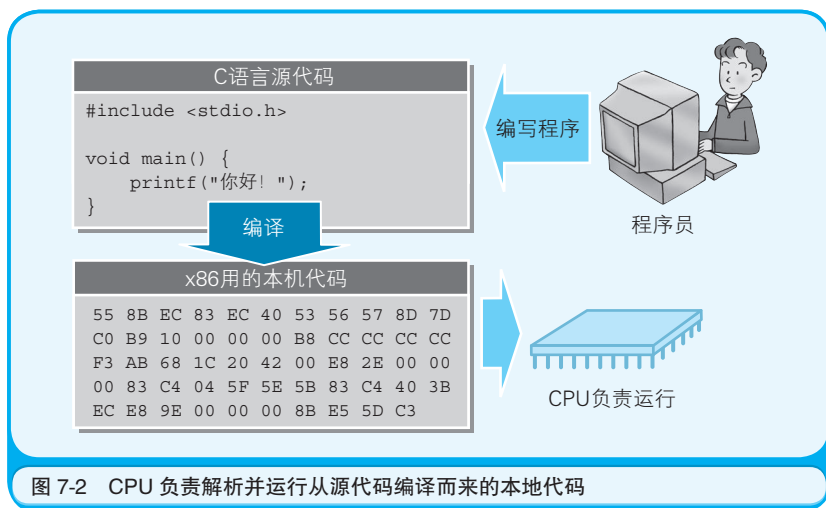


图 7-2 CPU 负责解析并运行从源代码编译而来的本地代码

## 7.2 Windows 克服了 CPU 以外的硬件差异

计算机的硬件并不仅仅是由 CPU 构成的，还包括用于存储程序指令和数据的内存，以及通过 I/O 连接的键盘、显示器、硬盘、打印机等外围设备。而计算机是如何控制这些外围设备的呢？这和计算机的机型有着很大的关系。

Windows 操作系统对克服这些硬件构成的差异做出了很大贡献。在介绍 Windows 之前，让我们先来回顾一下 Windows 的前身操作系统 MS-DOS<sup>①</sup> 广泛使用的时代。在 20 年前的 MS-DOS 时代，日本国内市场上有 NEC 的 PC-9801、富士通的 FMR、东芝的 Dynabook 等各种机型的计算机。Windows 3.0 及 3.1 问世前后，AT 兼容机开始普及，并开始同 PC-9801 争夺市场份额。

<sup>①</sup> MS-DOS (Microsoft Disk Operating System) 是 20 世纪 80 年代普遍使用的计算机操作系统。

这些机型虽然都搭载了 486 及 Pentium 等 x86 系列的 CPU，不过内存和 I/O 地址的构成等都是不同的，因此每个机型都需要有专门的 MS-DOS 应用。x86 提供有专门用来同外围设备进行输入输出的 I/O 地址空间（I/O 地址分配）。至于各外围设备会分配到什么样的地址，则要以计算机的机型来定。

例如，如果想使用当时大热的文字处理软件——JustSystem 的“一太郎”的话，就必须买各个机型专用的一太郎软件（图 7-3(a)）。这是因为，应用软件的功能中，存在着直接操作计算机硬件的部分。而这又是为什么呢？原因主要有两点，一是当时 MS-DOS 的功能尚不完善，二是为了提高程序的运行速度。

不过，随着 Windows 的广泛使用，这样的局面也得到了大幅改善。因为只要 Windows 能正常运行，同样的应用（本地代码）在任何机型上都是可以运行的（图 7-3(b)）。

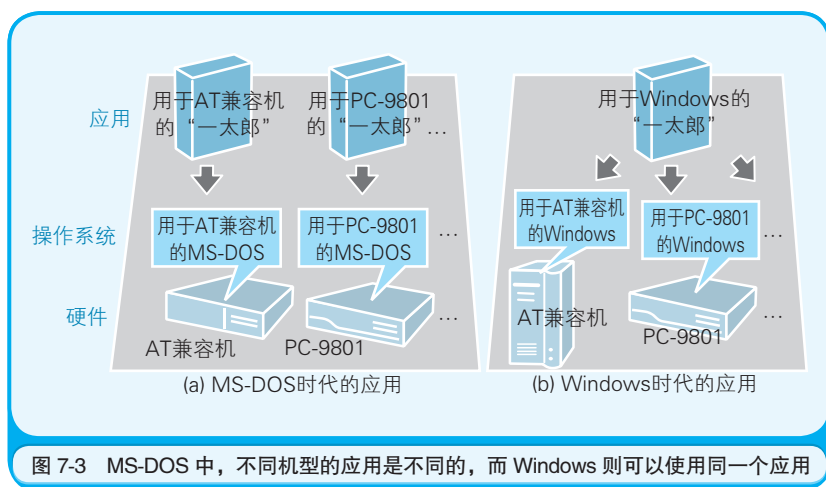
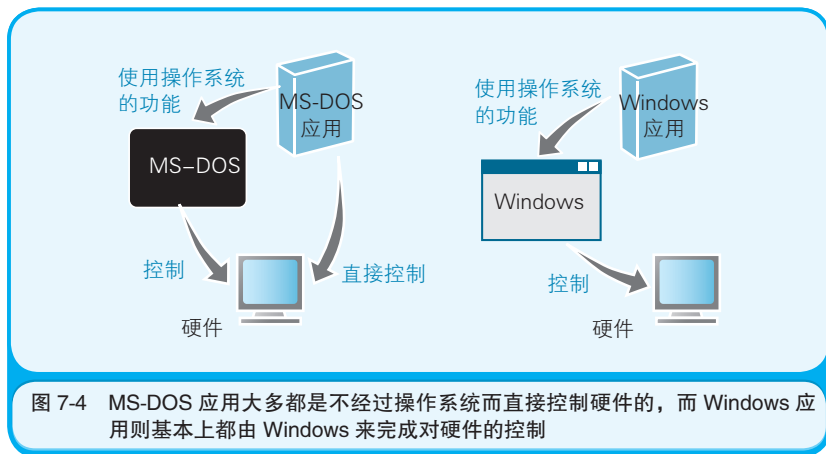


图 7-3 MS-DOS 中，不同机型的应用是不同的，而 Windows 则可以使用同一个应用



在 Windows 的应用软件中，键盘输入、显示器输出等并不是直接向硬件发送指令，而是通过向 Windows 发送指令来间接实现的。因此，程序员就不用注意内存和 I/O 地址的不同构成了。因为 Windows 操作的是硬件而非应用软件，而且针对不同的机型，这些硬件的构成也是有差异的（图 7-4）。不过，Windows 本身则需要为不同的机型分别提供专用的版本，比如用于 AT 兼容机的 Windows、用于 PC-9081 的 Windows 等。



而即便是 Windows，也依然无法吸收 CPU 类型的差异。这是因为，市面上销售的 Windows 应用软件，都是用特定的 CPU 的本地代码来完成的。

### 7.3 不同操作系统的 API 不同

接下来让我们看一下操作系统的种类。同样机型的计算机，可安装的操作系统类型也会有多种选择。例如，AT 兼容机的情况下，除

Windows 之外,还可以采用 Unix 系列的 Linux 及 FreeBSD<sup>①</sup> 等多个操作系统。当然,应用软件则必须根据不同的操作系统类型来专门开发。CPU 的类型不同,所对应的机器语言也不同,同样的道理,操作系统的类型不同,应用程序向操作系统传递指令的途径也是不同的。

应用程序向操作系统传递指令的途径称为 API (Application Programming Interface)<sup>②</sup>。Windows 及 Unix 系列操作系统的 API,提供了任何应用程序都可以利用的函数组合。因为不同操作系统的 API 是有差异的,因此,将同样的应用程序移植到其他操作系统时,就必须重写应用中利用到 API 的部分。像键盘输入、鼠标输入、显示器输出、文件输入输出等同外围设备进行输入输出操作的功能,都是通过 API 提供的。

在同类型操作系统下,不管硬件如何,API 基本上没有差别。因而,针对某特定操作系统的 API 所编写的程序,在任何硬件上都可以运行。当然,由于 CPU 种类不同,机器语言也不相同,因此本地代码当然也是不同的。这种情况下,就需要利用能够生成各 CPU 专用的本地代码的编译器,来对源代码进行重新编译了。

程序(本地代码)的运行环境是由操作系统和硬件来决定的,这一点想必大家都清楚了吧。



## 7.4 FreeBSD Port 帮你轻松使用源代码

不知道各位读者会不会有这样的想法:“既然 CPU 类型不同会导致

- 
- ① FreeBSD 是 1993 年加州大学伯克利分校的 Computer Systems Research Group 在 4.4BSD-Lite 的基础上开发的 Unix 系列操作系统。
  - ② API 也称为“系统调用”,是应用调用操作系统功能的手段。关于系统调用,我们会在第 9 章进行详细说明。