

时,也同时发明了称为JVM (Java Virtual Machine)的虚拟机(一种体系结构)。Java编译器为JVM生成代码,这些代码以后可以由一个软件JVM解释器执行。这种处理方式的优点在于,JVM代码可以通过Internet传送到任何有JVM解释器的计算机上,并在该机器上执行。举例来说,如果编译器生成了SPARC或Pentium二进制代码,这种代码不可能轻易地送到任何地方就能执行。(当然,Sun可以生产一种生成SPARC二进制代码的编译器,并且发布一种SPARC解释器,但是JVM具有非常简单的、只需要解释的体系结构。)使用JVM的另一种优点是,如果解释器正确地完成,并不意味着就结束了,还要对所输入的JVM进行安全性检查,然后在一种保护环境下执行,这样,这些程序就不能偷窃数据或进行其他任何有害的操作。

1.7.6 外核

与虚拟机克隆真实机器不同,另一种策略是对机器进行分区,换句话说,给每个用户整个资源的一个子集。这样,某一个虚拟机可能得到磁盘的0至1023盘块,而另一台虚拟机会得到1024至2047盘块,等等。

在底层中,一种称为外核(exokernel, Engler等人,1995)的程序在内核态中运行。它的任务是为虚拟机分配资源,并检查试图使用这些资源的企图,以确保没有机器会使用他人的资源。每个用户层的虚拟机可以运行自己的操作系统,如VM/370和Pentium虚拟8086等,但限制在只能使用已经申请并且获得分配的那部分资源。

外核机制的优点是,它减少了映像层。在其他的设计中,每个虚拟机都认为它有自己的磁盘,其盘块号从0到最大编号,这样虚拟机监控程序必须维护一张表格用以重映像磁盘地址(以及其他资源)。有了外核这个重映像处理就不需要了。外核只需要记录已经分配给各个虚拟机的有关资源即可。这个方法还有一个优点,它将多道程序(在外核内)与用户操作系统代码(在用户空间内)加以分离,而且相应负载并不重,这是因为外核所做的一切,只是保持多个虚拟机彼此不发生冲突。

1.8 依靠C的世界

操作系统通常是由许多程序员写成的,包括很多部分的大型C(有时是C++)程序。用于开发操作系统的环境,与个人(如学生)用于编写小型Java程序的环境是非常不同的。本节试图为那些有时编写Java的程序员简要地介绍编写操作系统的环境。

1.8.1 C语言

本部分不是C语言的指南,而是一个有关C和Java之间的关键差别的简要介绍。Java是基于C的,所以两者之间有许多类似之处。两者都是命令式的语言,例如,有数据类型、变量和控制语句等。在C中基本数据类型是整数(包括短整数和长整数)、字符和浮点数等。使用数组、结构体和联合,可以构造组合数据类型。C语言中的控制语句与Java类似,包括if、switch、for以及while等语句。在这两个语言中,函数和参数大致相同。

一项C语言中有的而Java中没有的特点是显式指针(explicit pointer)。指针是一种指向(即包含对象的地址)一个变量或数据结构的变量。考虑下面的语句

```
char c1, c2, *p;  
c1 = 'c';  
p = &c1;  
c2 = *p;
```

这些语句声明c1和c2是字符变量,而p是指向一个字符的变量(即包含字符的地址)。第一个赋值语句将字符c的ASCII代码存到变量c1中。第二个语句将c1的地址赋给指针变量p。第三个语句将由p指向变量的内容赋给变量c2,这样,在这些语句执行之后,c2也含有c的ASCII代码。在理论上,指针是输入类型,所以不能将浮点数地址赋给一个字符指针,但是在实践中,编译器接受这种赋值,尽管有时给出一个警告。指针是一种非常强大的结构,但是如果不仔细使用,也会是造成大量错误的一个原因。

C语言中没有的包括内建字符串、线程、包、类、对象、类型安全(type safety)以及垃圾回收(garbage collection)等。最后这一个是操作系统的一个“淋浴器塞子”。在C中分配的存储空间或者是静态的,或者是程序员明确分配和释放的,通常使用malloc以及free库函数。正是由于后面这个性质——全部由程序员控制所有内存——而且是用明确的指针,使得C语言对编写操作系统而言非常有吸引力。操作系统