

机系统都是批处理系统,直到20世纪60年代早期M.I.T.发明了交互式分时系统为止。20世纪60年代后期,即在Doug Engelbart于斯坦福研究院发明鼠标和图形用户接口之前,所有的计算机都是基于文本的。有谁会知道下一个发明将会是什么呢?

在本小节和本书中相关的其他章节中,我们会简要地介绍一些在过去5至10年中操作系统的研究工作,这是为了让读者了解可能会出现什么。这个介绍当然不全面,而且主要依据在高水平的期刊和会议上已经发表的文章,因为这些文章为了得以发表至少需要通过严格的同行评估过程。在有关研究内容一节中所引用的多数文章,它们或者发表在ACM刊物、IEEE计算机协会刊物或者USENIX刊物上,并对这些组织的(学生)成员们在Internet上开放。有关这些组织的更多信息以及它们的数字图书馆,可以访问:

ACM	http://www.acm.org
IEEE Computer Society	http://www.computer.org
USENIX	http://www.usenix.org

实际上,所有的操作系统研究人员都认识到,目前的操作系统是一个大的、不灵活、不可靠、不安全和带有错误的系统,而且特定的某个操作系统较其他的系统有更多的错误(这里略去了名称以避免责任)。所带来的结果是,大量的研究集中于如何构造更好的操作系统。近来出版的文献有如下一些,关于新操作系统(Krieger等人,2006),操作系统结构(Fassino等人,2002),操作系统正确性(Elphinstone等人,2007;Kumar和Li,2002;Yang等人,2006),操作系统可靠性(Swift等人,2006;LeVasseur等人,2004),虚拟机(Barham等人,2003;Garfinkel等人,2003;King等人,2003;Whitaker等人,2002),病毒和蠕虫(Costa等人,2005;Portokalidis等人,2006;Tucek等人,2007;Vrable等人,2005),错误和排错(Chou等人,2001;King等人,2005),超线程与多线程(Fedorova,2005;Bulpin和Pratt,2005),用户行为(Yu等人,2006),以及许多其他课题。

1.10 本书其他部分概要

我们已经叙述完毕引论,并且描绘了鸟瞰式的操作系统图景。现在是进入具体细节的时候了。正如前面已经叙述的,从程序员的角度来看,操作系统的基本目的是提供一些关键的抽象,其中最重要的是进程和线程、地址空间以及文件。所以后面三章都是有关这些关键主题的。

第2章讨论进程与线程,包括它们的性质以及它们之间如何通信。这一章还给出了大量关于进程间如何通信的例子以及如何避免某些错误。

第3章具体讨论地址空间以及关联的内存管理。讨论虚拟内存等重要课题,以及相关的概念,如页面处理和分段等。

第4章里,我们会讨论有关文件系统的所有重要内容。在某种程度上,用户大量看到的是文件系统。我们将研究文件系统接口和文件系统的实现。

输入/输出是第5章的内容。这一章介绍设备独立性和设备依赖性的概念。将把若干重要的设备,包括磁盘、键盘以及显示设备作为示例讲解。

第6章讨论死锁。在这一章中我们概要地说明什么是死锁,不过这章里有大量的内容需要介绍。还讨论了避免死锁的方法。

到此,我们完成了对单CPU操作系统基本原理的学习。不过,还有更多的高级内容要叙述。在第7章里,我们将了解多媒体系统,这类系统的大量特性和要求与传统的操作系统存在着差别。而在其他的篇幅里,我们会讨论多媒体的本质对调度处理和文件系统的影响。另一个高级课题是多处理器系统,包括多处理器、并行计算机以及分布式系统。这些内容放在第8章中讨论。

有一个非常重要的主题,就是操作系统安全,它是第9章的内容。在这一章中讨论的内容涉及威胁(例如,病毒和蠕虫)、保护机制以及安全模型。

随后,我们安排了一些实际操作系统的案例。它们是Linux(第10章)、Windows Vista(第11章)以及Symbian(第12章)。本书以第13章关于操作系统设计的一些思考作为结束。

1.11 公制单位

为了避免混乱,有必要在本书中特别指出,考虑到计算机科学的通用性,所以我们采用公制以代替