

5.1 操作系统的定义

一个完整的计算机系统由软件和硬件两部分组成,如图 5-1 所示。操作系统(Operating System, OS)是所有软件中最基础、核心的部分,是计算机用户和计算机硬件之间的中介,它为用户执行程序提供更方便、有效的运行环境。

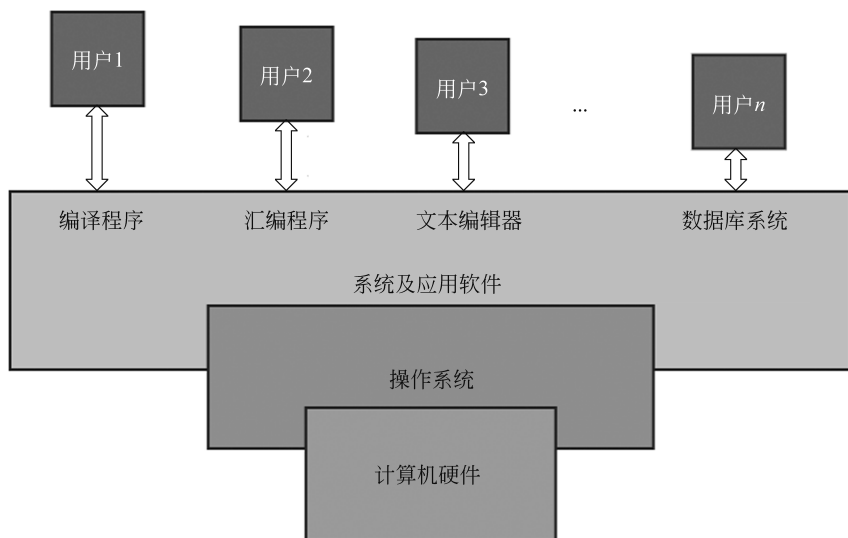


图 5-1 计算机系统的组成部分

计算机系统的基本目标是执行用户程序并使解决问题变得更加简单,为了实现这一目标,科学家和工程师创造了计算机硬件;单纯的硬件不便于使用,因此开发了应用程序;这些

程序需要某些通用操作,例如控制 I/O 设备的操作;最后,将这些控制和分配资源的通用功能整合到一个软件中就成了操作系统。操作系统的引入使得用户能拥有更方便使用的计算机系统。

5.2 操作系统的功能

操作系统是计算机系统的重要组成部分。计算机系统解决问题时需要用到如 CPU、内存空间、文件存储空间、I/O 设备等多种资源,操作系统是这些资源的管理器。面对多个程序对这些计算资源的请求,操作系统需要决定如何将它们分配给特定的程序和用户,以便其可以高效、公平地操作计算机系统^[60]。在多用户访问同一台大型机或小型计算机的情况下,资源分配尤其重要。同时,操作系统还要控制各种 I/O 设备、用户程序、管理用户程序的执行,以防止程序出错和计算机的不当使用。

以 Linux 操作系统为例,其内核主要有以下 5 个功能:内在管理、进程管理、文件系统、设备控制、网络接口。

5.2.1 内存管理

在早期的计算机中,程序会通过物理地址直接操作计算机内存。假如程序设计出现错误,有可能导致内存的非法访问,如访问不存在的地址,或者对系统程序所使用的内存进行读写,造成程序崩溃;同时,由于程序直接访问物理地址,有可能在多进程时导致不同作业对相同的内存进行读写,甚至导致系统崩溃。

Linux 内核提供了一种名为动态重定位的内存管理功能,如图 5-2 所示,允许程序装入内存时使用相对地址,程序在内存中的起始地址记录在重定位寄存器中;程序执行时,系统会调取重定位寄存器中的起始地址,将其与程序的相对地址相加,得到真正访问的内存地址。动态重定位可以解决内存物理地址暴露产生的问题,使每个程序都有自己的内存空间而不会互相干扰。

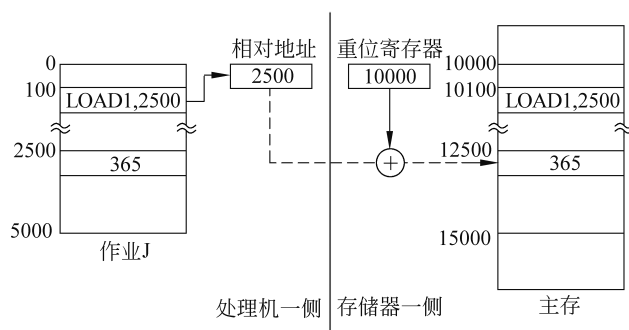


图 5-2 内存动态重定位

Linux 内核还实现了虚拟内存管理业务。一个程序需要将其整体装入内存才能运行,但是由于物理内存容量有限,因此会出现程序过大不能完全装入内存的情况,导致运行失败;或是由一个程序占用大量内存,导致大量程序在外存等待执行的问题。虚拟内存技术将

内存分页,先将当前需要访问的页放入物理内存,暂未使用的部分仍存放在外存中;程序执行期间,若需要的页存放在内存中,则继续运行;否则触发缺页中断,将相应的页调入内存中。虚拟内存技术解决了由于物理内存限制而可能产生的问题,通过虚拟内存从逻辑上扩充内存容量,允许程序使用超过物理内存大小的内存。

5.2.2 进程管理

进程(Process)是操作系统资源调配的基本单位。进程管理是操作系统最基本的功能之一^[61]。每一个程序至少对应一个进程。程序是一系列指令的有序集合,是一个静态实体;当程序被编译为可执行文件运行时,处在执行状态的程序称为进程,是一个动态的实体,有自己的生命周期。进程管理的任务是为进程分配资源,选择使用 CPU 的进程并控制使用的时长。

一个进程基本包含 3 种状态(就绪状态、执行状态、阻塞状态)和两种常见的状态(创建状态、终止状态)。如图 5-3 所示,在进程创建时,系统会为其分配一个进程控制块,记录进程所需的资源信息,由操作系统为其分配,当除 CPU 外所有资源需求得到满足时,进程进入就绪状态,在就绪队列中进行等待;操作系统会以一定的进程调度算法为就绪队列中的进程分配 CPU 资源,得到 CPU 资源后,进程变为执行状态。在操作系统分配的时间片用完后,进程会暂停并重新进入就绪队列,等待操作系统调度;若因 I/O 请求等事件之行受阻,则进入阻塞状态,待事件完成后重新进入就绪队列;若进程执行完成,则进入终止状态,系统会将其删除并释放资源。

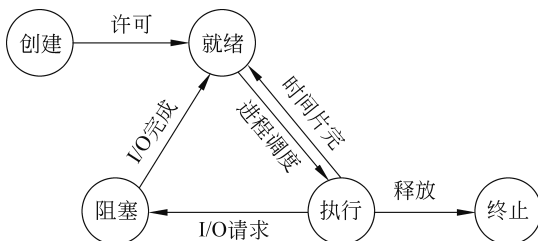


图 5-3 进程的 5 种基本状态及转换

5.2.3 文件系统

Linux 系统的一大特点是“一切皆文件”——将系统中的数据和硬件抽象为文件进行管理。Linux 将资源抽象为文件的做法十分清晰、优雅：文件系统需要提供 API 给系统调用,以方便对文件系统的内容进行操作。由于“文件”的操作只有读和写,因此将设备等进行字节流输入输出的对象统一抽象为文件,便可使用一组通用的 API 和工具对 Linux 的资源进行读写。

Linux 一共有以下 7 种文件类型。

(1) 普通文件。包括视频、文档、图片等应用层面上的文件,可以进行查看、修改和删除等操作。

(2) 目录文件。就是将文件夹描述为文件,通过打开目录文件可以查看里面包含的内容。

(3) 符号链接。类似于快捷方式,通过访问符号链接文件即可访问链接的目标文件或目录。

(4) 套接字。一般用于网络通信,利用套接字的相关函数可以完成网络的通信过程。

(5) 块设备。如磁盘。

(6) 字符设备。如键盘、鼠标、打印机、硬盘和光驱等设备。

(7) 管道文件。是一种特殊的文件,用于进程间的数据交换。

Linux 通过挂载点“/”对这些文件进行结构化管理,如设备文件一般存储在“/dev”目录中,套接字文件则主要存储在“/var/run”目录中。通过文件的抽象模型,Linux 可以对系统数据及设备进行结构化管理。Linux 系统的目录结构如 5-4 所示。

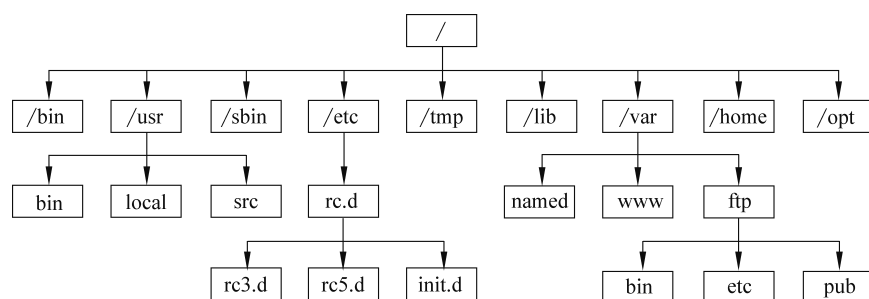


图 5-4 Linux 系统的目录结构

Linux 文件系统还提供了一个十分安全的树形目录结构模型。一般用户只能访问其“/home”目录下所拥有的文件,对于其他不属于该用户或者处于其父目录的文件,要求该用户属于超级用户,并使用超级用户权限进行访问。

5.2.4 设备控制

驱动硬件是 Linux 内核最重要的功能之一。操作系统内核有微内核与宏内核之分,如图 5-5 所示。微内核一般只包含基础 IPC、虚拟内存和调度等基本功能,而 Linux 使用的宏

基于单片内核的操作系统

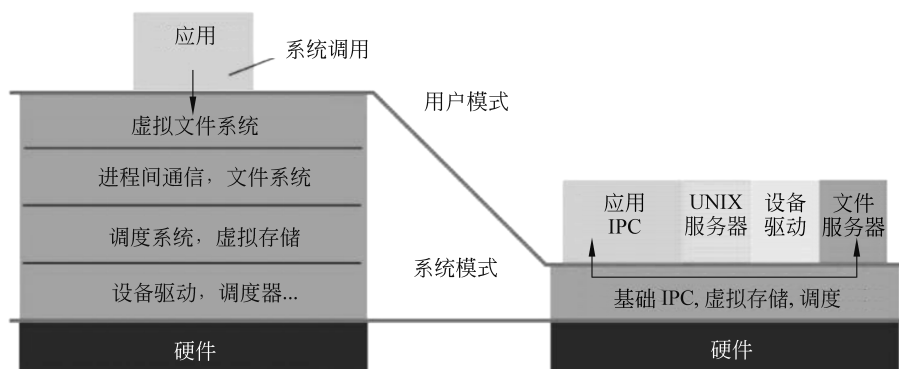


图 5-5 微内核与宏内核

内核将所有的驱动与服务打包在内核里。一般来说,内核版本越高,支持的硬件越多,计算机中的声卡、显卡、鼠标、键盘等硬件都需要依靠内核进行驱动。驱动程序相当于硬件的接口,实现了操作系统对硬件的有效管理,内核只有通过驱动程序才能访问硬件。

在有硬件接口之后,内核通过硬件抽象将硬件接口封装为系统调用接口,面对不同的硬件,程序员只需针对系统提供的 API 进行开发,而无须理会硬件操作的细节。例如,Linux 内核将鼠标、键盘、打印机和硬盘等设备抽象为文件,访问设备时只需调用对设备文件的读写操作。这使得计算机硬件对程序员透明,使得开发过程更简单。

5.2.5 网络接口

Linux 系统的网络功能,如数据报文的收集、识别、分发、程序与网口间的数据报处理及所有的路由和地址解析,都依靠网络接口实现。网络接口包含网络设备接口、网络接口核心、网络协议簇和网络接口 Socket 层 4 部分。

5.3 操作系统的发展史

计算机硬件和软件共同构成计算机系统。计算机硬件的每一次革新都伴随着操作系统的发展。操作系统的发展大致分为 4 个阶段,如图 5-6 所示。第一代的电子管计算机诞生于 20 世纪 40 年代,当时操作系统尚未出现,程序员直接与硬件打交道;第二代的晶体管计算机始于 20 世纪 50 年代,为了提高计算资源的使用效率,减少空闲时间,提出了单道批处理系统;20 世纪 60 年代,随着小规模集成电路的发展,出现了多道批操作系统,以进一步提高资源的使用效率;20 世纪 70 年代,大规模集成电路飞速发展,操作系统百家争鸣,涌现出 UNIX、DOS、Windows、Mac OS、Linux 等著名的操作系统。

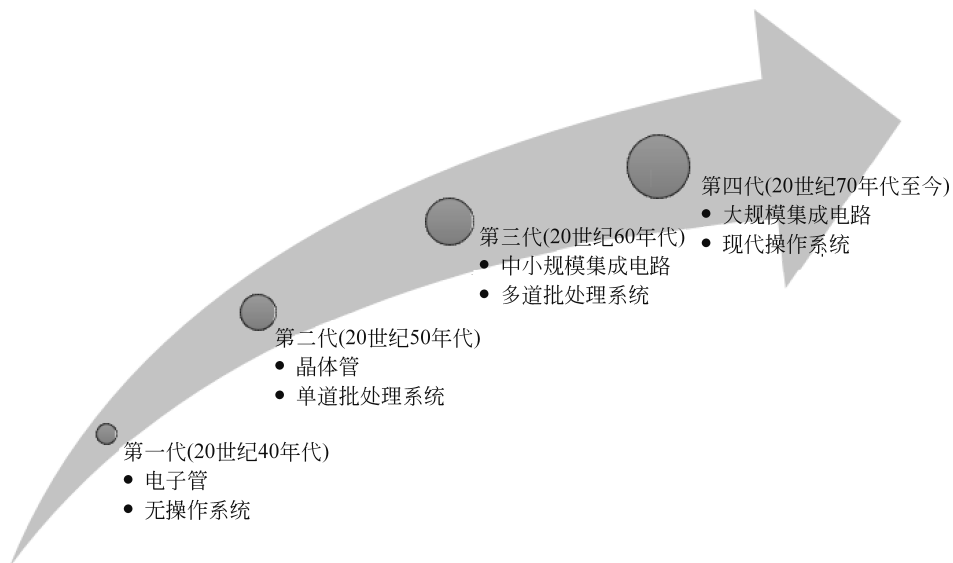


图 5-6 操作系统的发展史

5.3.1 第一代操作系统

早期计算机的功能十分简单,如图 5-7 所示,使用者(程序员)通过人工操作的方式直接与硬件打交道,程序作为控制器的一部分,编程需要硬件化操作。当需要进行计算时,程序员通过插件板和电缆直接连接计算机的各个部件实现基本计算功能;在计算完成后,下一位使用者需先将原来的线缆拔出,再按照需求重新连接电路,以完成其他任务。

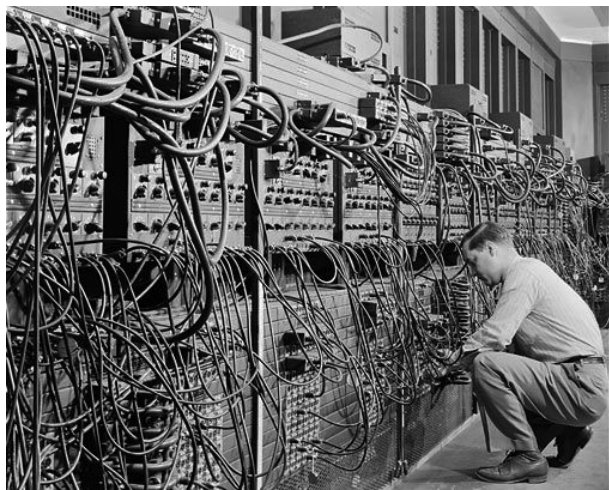


图 5-7 程序员通过插拔电缆进行编程

显然,这种通过插拔电缆进行编程的方法效率太低,于是工程师尝试寻求一种可以将程序与数据统一存储在存储器中的方法,以提高执行效率。1945 年,数学家冯·诺依曼通过数学语言系统地阐述了这个思想,提出存储程序型计算机模型,将程序编码为数据一同存放在存储器中。存储程序型计算机模型的提出是计算机领域的一大突破,从此计算机变得可以编程。在冯·诺依曼结构中,内存被划分为内存单元,如图 5-8 所示,每个内存单元都有唯一的地址进行索引。每个内存单元分为两部分,一部分为操作码(ADD/MOV 等)或操作数(数字/地址);另一部分为下一个内存单元的地址。计算机从程序的起始地址开始读取内存单元中的指令并执行,然后根据指令给出的地址读取下一个内存单元中的内容。自此,程序的执行变为依次从存储器中取出指令并执行。

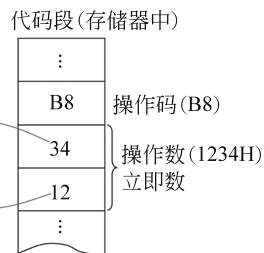


图 5-8 内存示意图

1949 年 8 月,第一台冯·诺依曼计算机 EDVAC 交付给弹道研究实验室,并于 1953 年开始运行。由于那个年代还没有键盘、屏幕等 I/O 设备,程序员会通过打孔机的键盘将代码转录到穿孔卡片中,最后将卡片放进输入设备进行加载,程序执行完成后,将结果输出到打印机进行打印。

总的来说,第一代计算机尚未使用操作系统,程序员直接使用硬件系统(通过插拔电缆或打孔卡片编程),这种方式的优点是在预约上机时间内可以独享整台计算机,方便程序员

对程序进行调试。但这种方式存在一个不足——浪费计算资源。由于计算资源独占,预约时间内仅有一个人在使用计算机,若使用者提前完成工作则会导致计算资源的闲置;若在预约时间内不能完成工作,计算也会被终止,导致已使用资源的浪费。同时,由于存储介质容量有限,使用者需要频繁地换下已读卡片,换上新的卡片,在切换过程中计算机的 CPU 仍在运行,由此造成资源闲置的问题。

5.3.2 第二代操作系统

20 世纪 50 年代中期,随着晶体管的提出,计算机逐渐从使用真空管转为使用晶体管,从而出现了第二代计算机。由于晶体管的自身特性,第二代计算机更为精简、节能,同时性能和可靠性大幅提高,此时计算机已经拥有推广价值,但由于低效调度造成的计算资源闲置难以令人接受,因此,使用计算机系统的成本仍十分高。

为进一步提高利用率,减少空闲时间,如图 5-9 所示,20 世纪 50 年代人们普遍使用的一种方法是将批处理与脱机输入/输出方式相结合,即程序员将穿孔卡片交由专门的计算机操作员;由操作员将作业(程序)按顺序组织成批读到输入磁带上;由计算机监督程序控制作业顺序执行;将输出结果写入输出磁带。当输入磁带上的作业执行完成后,由操作员更换输入磁带,并将输出磁带的内容进行打印。

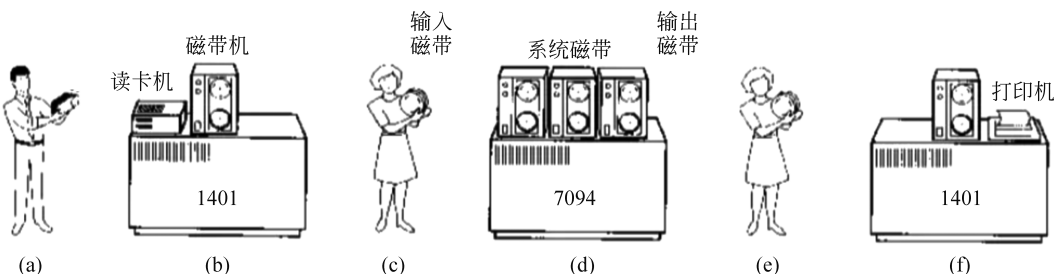


图 5-9 磁带机与单道批处理系统

监督程序在执行过程中会先将磁带上的作业装入内存,编译源程序并检查源程序。若正确,则将计算机控制权交给该作业;作业执行完成后,将控制权交还给监督程序,监督程序再读取下一个作业,直至执行完所有作业。

由于作业按批以流的形式进入内存,因此这种系统被称作单道批处理系统。单道批处理系统的代表是通用汽车公司(General Motors, GM)公司和北美航空(North American Aviation, NAA)公司为 IBM 704 编写的 GM-NAA I/O 系统,该系统的主要功能是在批处理过程中,当一个作业执行完成后,调入一个新的作业。

由于批处理的特点,作业成批进行输入输出,减少了人员调度及装卸卡片所耗费的时间,计算机的执行效率得以提高。但是单道批处理系统仍存在一定问题,磁带中的作业串行执行(只有一个作业),此时若作业在等待 I/O 操作,会造成计算资源闲置,影响资源使用效率;另外,程序员需要等待一批作业处理完成后,才能得到打印结果,不利于程序的交互及调试。

5.3.3 第三代操作系统

20 世纪 60 年代中期,人们开始利用小规模集成电路制作计算机,生产出第三代计算机。其较晶体管计算机,无论在体积、功耗、速度和可靠性上,都有显著的改善。因此,单道批处理系统的劣势在新一代计算机中被放大——单道使得监督程序每次将一个作业调入内存,当该作业因某些原因如等待外设响应而需要暂停时,将会使得计算资源空闲。

为进一步提升资源的使用率和系统吞吐量,操作系统需具备多作业共享一台计算机的功能,当一个作业因各种原因需要暂停时,系统可以将空闲出来的 CPU 分配给其他作业,这个思想称为多道,这种操作系统称为多道批处理系统。如图 5-10 所示,多道系统在外存中设置有后备队列,用户提交的作业都存储在队列中,由调度程序根据调度算法将作业送到内存;处理器会根据需要从一个作业切换到另一个作业,以在保持外设使用的同时保持多个作业的进行。

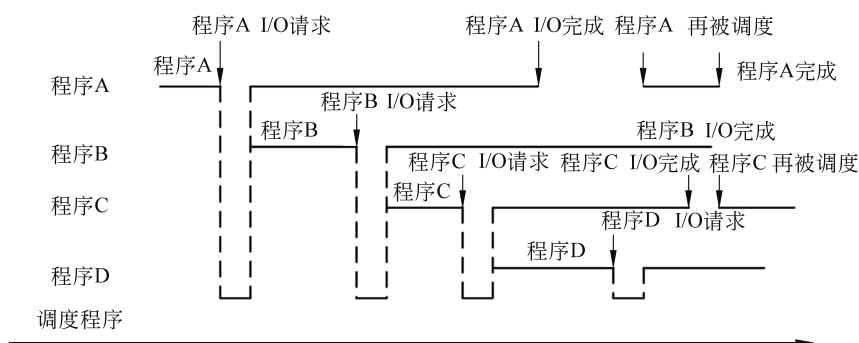


图 5-10 多道批处理系统的运行情况

1964 年,IBM 发布了大型主机的经典之作 IBM System/360(360 机),与 360 机一同推出的还有适配整个 System/360 系列的操作系统 OS/360,该系统最多可运行 15 道程序。早在 20 世纪 60 年代,这个系统的开发耗资高达 5 亿美元,开发难度之高,骇人听闻。最终 IBM 投入超过 2000 名工程师,耗费 5000 人·年,成功开发出 OS/360,后被评价为 IBM 历史上最重要的项目之一。

分时系统是第三代操作系统的另一个代表。多道批处理系统在内存和 I/O 设备的利用率以及系统吞吐量方面实现了较大的提升,然而其交互能力仍有欠缺:作业一旦被提交,将由系统负责进行调度,在作业完成之前用户都不能与程序进行互动,对修改和调试程序造成了极大的不便。使用者希望能像使用第一代计算机一样,由个人独占计算资源,以便与程序交互及进行调试,这给操作系统提出了一个要求——多名用户同时共享一台计算机。分时系统的出现满足了这个需求:操作系统按照设定的时间周期执行多个用户的作业,当一个时间周期结束而作业尚未执行完成时,系统会将其暂停并执行下一个用户的作业;由于系统切换速度极快,因此用户很难感知系统在切换不同的作业。

同一时期,还出现了外部设备联机并行操作技术(simultaneous peripheral operations on line,SPOOLing),该技术的出现进一步缓和了 I/O 速度与 CPU 速度不匹配的问题。SPOOLing 技术在高速的磁盘上设置了输入井与输出井,用以模拟脱机输入输出技术中的

输入磁盘与输出磁盘,使得可在主机控制下直接实现脱机输入输出的功能,简化人工搬运磁带的步骤。

5.3.4 第四代操作系统

20 世纪 70 年代,随着大规模集成电路的发展,微处理器的性能日益强大,计算机逐渐从大型机发展为微型机,操作系统的发展也进入了个人计算机和 workstation 时代。两款著名的操作系统占领了大部分 PC 市场:微软公司编写的 MS-DOS 操作系统主要运行在 IBM 个人计算机和使用英特尔处理器的计算机中;UNIX 则主要运行在使用摩托罗拉处理器的大型个人计算机中。

1. UNIX

20 世纪 60 年代初,肯·汤普森(Ken Thompson)和丹尼斯·里奇(Dennis Ritchie)曾参与设计 MIT 的分时操作系统 MULTICS,这个操作系统的设计思路对往后的操作系统造成了深远的影响。1969 年,当时在新泽西州 AT&T 贝尔实验室工作的肯·汤普森希望为他的 PDP-7 计算机编写一个操作系统,以方便使用计算机;为向 MULTICS 致敬,系统起初命名为 UNICS,后来拼写改为 UNIX。UNIX 操作系统如图 5-11 所示,是操作系统的一大里程碑,它因强大的功能受到广大计算机专业人士的喜爱。同时,丹尼斯·里奇正在开发一门以“B 语言”为蓝本的编程语言“C”。1973 年,UNIX 的编写从原来使用的汇编语言改为 C 语言。

```
MAN(1)                                NetBSD General Commands Manual                                MAN(1)
NAME
    man - display the on-line manual pages (aka ``man pages``)
SYNOPSIS
    man [-acw|-h] [-C file] [-M path] [-m path] [-S srch] [[-s] section] name
    man [-k] [-C file] [-M path] [-m path] keyword ...
DESCRIPTION
    The man utility displays the BSD man pages entitled name.
    The options are as follows:
    -a      Display all of the man pages for a specified section and name
            combination. (Normally, only the first man page found is displayed.)
    -C      Use the specified file instead of the default configuration file.
            This permits users to configure their own man environment. See
            man.conf(5) for a description of the contents of this file.
/usr/share/man//cat1/man.0 20%
```

图 5-11 UNIX 操作系统

2. MS-DOS

最早的个人计算机操作系统为加里·基尔德(Gary Kildall)编写的 CP/M(Control Program / (for) MicroComputers),由于其较好的体系结构设计,因此称之为 8 位 CPU 时代主流的操作系统。受 CP/M 的影响,1980 年西雅图计算机公司的一名程序员蒂姆·帕特森花费 4 个月编写出 MS-DOS 的前身 86-DOS,之后微软将其收购。1981 年,微软发布了著名的 MS-DOS 操作系统,如图 5-12 所示。MS-DOS 是 DOS 系列操作系统中最有名的一个。

在 Windows 95 之前, DOS 是 IBM PC 和兼容计算机中最基本的配备, 而 MS-DOS 是个人计算机中最常用的 DOS 操作系统。



图 5-12 MS-DOS 操作系统

3. Mac OS

1984 年, 苹果公司发布了麦金塔电脑 (Macintosh), 与麦金塔一同发布的还有 Mac OS, 如图 5-13 所示。Mac OS 被认为是第一款带有图形用户界面 (GUI) 的操作系统。最早的图形用户界面概念出现在 20 世纪 70 年代, 施乐 PARC 的工程师正在打造一款通过鼠标的单击代替命令行操作的产品; 1979 年, 施乐 PARC 邀请乔布斯等人前去参观, 在施乐 PARC 会面结束之后, 乔布斯和他的团队将 GUI 变成现实, 最终发展出自己的图形化操作系统 Mac OS。如今, 许多图形界面最基本的设计要素和操作规则如下拉菜单、桌面图标、双击和拖曳等都源于苹果电脑。

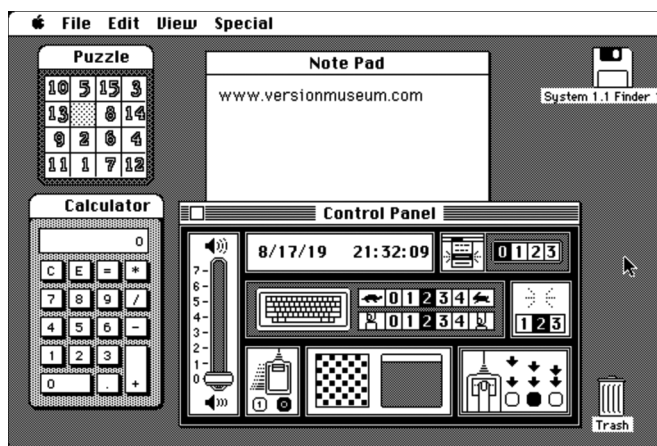


图 5-13 Mac OS 桌面环境

4. Windows

比尔·盖茨和乔布斯因麦金塔电脑而合作, 比尔·盖茨和他的团队为麦金塔开发 Word 和 Excel 等应用软件, 在此过程中他察觉图形界面将会是未来的方向。1983 年 11 月, 微软计划为 IBM 个人计算机开发操作系统。1985 年, Windows 正式问世。Windows v1.01 桌面环境如图 5-14 所示。初版的 Windows 基于 MS-DOS 进行开发, 因此和 MS-DOS 一样在功