

第一篇 操作系统的基本原理

第 1 章 操作系统概论

1.1 操作系统的定义

1. 计算机系统的组成

随着计算机技术的迅速发展，计算机系统的硬件和软件资源越来越丰富。从功能上，可把整个计算机系统划分为四个层次：硬件、操作系统、实用程序和应用程序，如图 1.1 所示。这四层的关系表现为一种单向调用关系，即外层的软件必须事先约定的方式调用内层软件或硬件提供的服务。通常把这种约定称为界面或接口（interface）。下面简单介绍一下各个层次的特点。

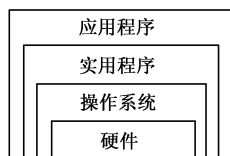


图 1.1 计算机系统的组成

（1）硬件

硬件层由计算机的硬件资源组成，它包括中央处理机（简称 CPU）、存储器和输入/输出设备。其中，存储器包括主存和辅助存储器（又叫外存，包括磁盘、磁带、光盘等）；输入设备包括卡片或纸带输入机、键盘、鼠标、图形扫描仪等；输出设备有卡片或纸带输出机、显示器、打印机、绘图机等。这种不加任何扩充的计算机称为裸机。一个裸机的功能即使很强，用户也很难使用，因为这些裸机的部件是通过执行机器指令来实现计算和输入/输出功能的。

（2）操作系统

操作系统是整个计算机系统的管理和控制中心。一个计算机如果没有操作系统，整个系统将无法工作。

（3）实用程序

实用程序层包括下面一些程序，且它们通常是驻留在磁盘上的。

- ① 各种语言编译程序。语言编译程序包括高级语言和汇编语言的编译和汇编程序。
- ② 文本编辑程序。文本编辑程序是用来建立和修改用户的源程序或其他文本数据的，这类程序种类繁多，功能各异，有面向行的和面向全屏幕的。
- ③ 调试排错程序。它用来帮助用户调试程序，从而方便地找出程序中的逻辑错误。
- ④ 连接装配和装入程序。连接装配程序把用户独立编译好的各目标程序连接装配成一个可执行的程序。而装入程序，则负责将一个可执行程序装入到主存运行。

还有一些实用程序，如标准过程和函数、系统诊断程序、文件加密/解密程序以及用户连网使用的 Internet 浏览器等。总之，这些实用程序为用户提供了良好的使用计算机系统环境。

(4) 应用程序

应用程序是计算机系统的最外层软件，它主要负责解决用户的实际问题。这些程序通常由用户或专门的软件公司编制。这类软件比较丰富，如各种数据库管理软件、计算机辅助设计软件、各种事务处理软件（如制表软件、Web 浏览器）等。

2. 操作系统的定义和设计目标

为了深入、全面地理解操作系统的含义，从以下两个方面来阐述：

① 从计算机系统设计者的角度看，操作系统是由一系列程序模块组成的一个大的系统管理程序，它依据设计者设计的各种管理和调度策略，对计算机的硬件和软件（程序和数据）资源进行管理和调度，合理地组织计算机的工作流程，从而提高资源的利用效率。由此可认为，操作系统是计算机软硬件资源的管理和控制程序。

② 从用户角度看，配上操作系统的计算机是一台比裸机功能更强、使用更方便简单的虚拟机。也即，它是用户与计算机系统之间的一个接口界面，用户通过它来使用计算机。它向用户及其程序提供了一个良好的使用计算机的环境。它使系统变得容易维护、安全可靠、容错能力强和更加有效。

从上述两个角度可总结出，设计操作系统的目标有两个：一个是使用户方便、简单地使用计算机系统，另一个就是使计算机系统能高效可靠地运转。故操作系统是现代计算机系统不可缺少的关键部件。计算机系统越复杂，操作系统越显得重要。为此，有必要了解操作系统的组成和功能，以便更好地利用计算机进行系统和应用开发。这正是本书要讨论的主题。

1.2 操作系统的形成与发展

1.2.1 顺序处理（手工操作阶段）

早期的计算机（20 世纪的 40 年代~50 年代中期），程序员直接与计算机硬件打交道，没有操作系统。计算机由 CPU、主存、某种类型的输入设备（卡片输入机）、一台打印机，以及装有显示灯、乒乓开关的操作控制台组成。用机器代码或汇编或高级语言编写的程序通过输入设备装入计算机，再由程序员或操作员从控制台上通过设置乒乓开关启动汇编或编译或装入程序运行等。程序处理过程中，若出现错误，则通过控制台上的指示灯指示错误产生的条件并停止运行。经修改后可再次运行。如果程序正常完成，将输出结果送打印机打印。然后，将计算机转让给另一个程序员使用。

早期的这种操作方式，操作员或程序员要花费大量时间操作计算机，导致系统昂贵资源的无效使用，这主要由下面两方面问题造成。

1. 人工负责计算机的调度

计算机各资源的使用是通过一张纸登记的。一般的做法是，各用户说明他大约使用计算机多长时间，由机房负责人为他安排上机时间。如果一个用户预约 1 小时，结果用 45 分钟完成了，剩余的 15 分钟被浪费了。另一方面，若用户程序运行过程中出现问题，由于要检查错误和解决问题，被迫暂停，等待下一次进行预约后才能运行。

2. 人工负责编排作业的运行顺序

程序，又叫作业。在操作系统中，通常把用户在一次算题过程中要求计算机所做工作的集合叫做一个作业。以现在执行一个作业为例，用户在计算机上进行算题时，通常要经历以下几步：

① 采用某种语言按算法编写源程序，将源程序通过某种手段（如卡片输入机等）送入计算机；

② 调用某语言的汇编或编译程序，对源程序进行汇编或编译，产生目标代码程序；

③ 调用链接装配程序，将目标代码及调用的各种库代码连接装配成一个可执行程序；

④ 装入可执行程序 and 运行时所需数据，运行该程序并产生计算结果。

由此可见，一个算题任务通常要经历建立、编译、连接装配和运行，才能得到计算结果。把这些相对独立的每一步骤叫做作业步。一个作业的各作业步之间总是相互联系的，在逻辑上是顺序执行的。下一作业步能否执行，完全取决于上一作业步是否成功完成。比如，若汇编或编译失败，则不可能进行连接装配。

一次运行可能只是编译一个程序。为此要将编译程序和一个用高级语言编写的源程序装入主存，保存被编译好的目标程序。然后连接和装入目标程序和公共库函数。其中，每一步都涉及安装和卸下磁带或卡片叠等。如果在一次处理中出现错误，用户不得不返回到编排序列的开始重新运行，因此，大量时间用在编排程序运行上了，导致系统效率极低。这种操作方式叫顺序处理。

1.2.2 简单的批处理系统

早期计算机造价昂贵，上述的那种人工调度和编排作业的方式浪费了大量的计算机时间。简单的批处理系统（Simple Batch System）正是在 20 世纪 50 年代后期到 60 年代中期，伴随第二代计算机的出现而研制成功的。

简单的批处理模式的中心思想是使用一个监控程序软件，各个用户将各自作业的卡片叠或纸带交给机房的操作员，再由操作员将这些作业的卡片或纸带按序成批地放在一个输入设备上，由监控程序自动控制输入设备将各个作业读入到磁带上。之后，监控程序按照顺序自动地把一个个作业装入内存进行处理。

监控程序常驻主存，工作非常简单，就是将 CPU 的控制权自动地从一个作业转换到另一个作业。很清楚，手工操作阶段的两个问题已得到圆满解决：一方面由监控程序处理调度问题，各作业以尽可能快的速度执行，从而不存在空闲的机器时间；另一方面由监控程序处理作业编排问题。为了使监控程序能代替用户完成对作业的编排控制，系统向用户提供了一套作业控制命令。每当用户提交作业时，将对作业的控制意图用作业控制卡或作业说明书的形式提交给监控程序。下面是监控程序使用的几个典型的控制卡。控制卡以“\$”开头，以区别其他一般的卡片。

控制卡	功能
\$JOB	启动一个新程序
\$FORTRAN	调用 FORTRAN 编译程序
\$LOAD	调用装入程序，装入要执行的程序
\$RUN	运行程序
\$END	作业结束卡

图 1.2 给出了以卡片形式提交的一个用 FORTRAN 语言编写的程序及程序所用数据，以及控制作业执行的控制指令卡。

为了执行这个作业，监控程序首先读入控制卡 \$FORTRAN，从外存磁带上装入 FORTRAN 编译程序。编译程序翻译紧跟其后的 FORTRAN 源程序为目标代码，并存入主存或外存磁带上。如果它存入主存，这个操作叫“编译、装入和执行”。如果它存入磁带上，只编译不装入执行，那么当要执行程序时，还需要读入一个 \$LOAD 卡。监控程序读入该卡片后，通过执行装入程序将目标程序及其调用的标准函数装入主存。接着读 \$RUN 卡，启动执行用户程序。在用户程序执行期间，用户程序的每个输入指令都引起读一张数据卡。一个用户作业的成功或错误完成，都引起监控程序扫描输入卡，直到遇到下一个作业的控制卡 \$JOB 为止。

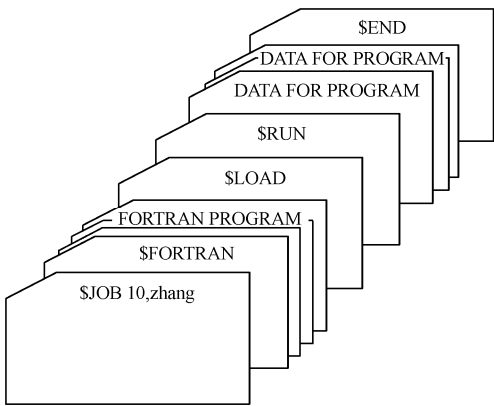


图 1.2 一个典型的输入作业的结构

当计算机运行中发生错误或意外时，监控程序通过控制台打字机输出信息向操作员报告。这种输出信息不仅比亮灯显示所表达的更为丰富，而且便于操作员理解。总之，用这种半自动方式控制计算机不仅提高了效率，而且方便了使用。这种简单的批处理在硬件结构上有两种不同的控制方式。

1. 早期的联机批处理

早期的联机批处理的硬件控制方式是：作业的输入、计算和输出都是在 CPU 直接控制下进行的。这样，在输入或输出过程中，主机的速度降低为慢速的输入或输出设备的速度。图 1.3 给出了联机批处理的模型。

2. 早期的脱机批处理

为了提高 CPU 的利用率，使 CPU 从慢速的外设控制中解脱出来，引入了早期的脱机批处理方式。该方式下，系统增设一台小型卫星机专门用来控制外部设备的输入/输出。其模型如图 1.4 所示。

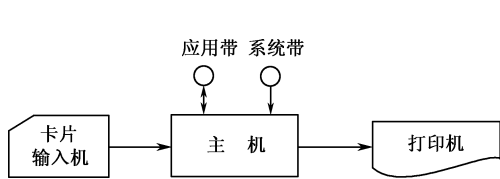


图 1.3 早期的联机批处理模型

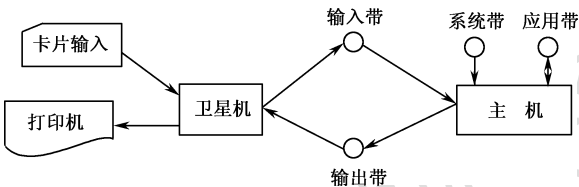


图 1.4 早期的脱机批处理模型

在这种模型中，小型卫星机的作用如下。

- ① 把卡片输入机上的作业逐个地记到输入磁带上，以便主机执行。
- ② 控制打印机输出由主机送入输出磁带上的作业的执行结果。

由此可见，采用这种脱机技术后，主机的所有输入/输出都是通过磁带进行的，而且主机与慢速外部设备可以并行，从而提高了主机运行效率。从 20 世纪 50 年代后期到 60 年代中期，脱机批处理运行得相当成功。脱机技术的实质是用快速的输入/输出设备代替慢速的设备。

1.2.3 多道成批处理系统

1. 多道程序设计 (Multiprogrammed System)

进入 20 世纪 60 年代中后期, 计算机的硬件有了突飞猛进的发展, 产生了硬件通道、中断和缓冲技术, 从而使得计算机在组织结构上发生了重大变革。原先以 CPU 为中心的体系结构, 转变为以主存为中心, 其结构模型如图 1.5 所示。

所谓通道，实际是一种比 CPU 速度较慢、价格较便宜的硬件。它是比小型卫星机更经济的、独立于 CPU 的、专门用于控制输入/输出设备的 I/O 处理机。通道连接着主存和外设，具有与主存直接交换数据的能力。当需要输入/输出时，CPU 只要向通道发一个命令，通道就独立地控制相应的外部设备完成指定的传输任务。通道通过中断机构向 CPU 报告其完成情况。这样，利用缓冲技术，通道控制输入可以更充分地并行执行，提高了 CPU 的利用率。

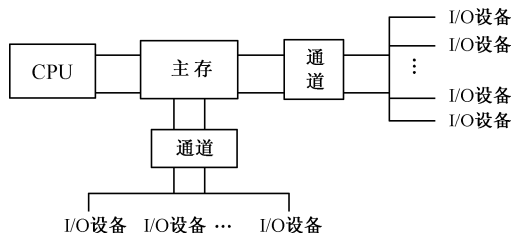


图 1.5 多道程序系统的计算机模型

简单的批处理系统提供了作业自动定序处理。通道和中断的出现,使 CPU 摆脱了对慢速设备的控制,从而提高了系统的处理效率。但由于主存只存放一个用户作业,当作业请求输入/输出时, CPU 空闲等待输入/输出完成。如果能在主存同时放多个用户作业,当一个作业等待数据传输时, CPU 转去执行其他作业,从而保证 CPU 与系统中的输入/输出设备并行操作。

这种在主存中同时存放多个作业，使之同时处于运行状态的程序设计方法，叫做多道程序设计。对于一个单处理机的系统来说，“多作业同时处于运行状态”是从宏观上理解的，其含义是指每个作业都已开始运行，但都尚未完成。而从微观上看，各个作业是串行执行的。在任何特定时刻，只有一个作业在处理机上运行。

引入多道程序设计技术的根本目的是提高 CPU 的利用率，充分发挥系统设备的并行性。这包括程序之间、CPU 与设备之间、设备与设备之间的并行操作。为了更好地理解引入多道程序的好处，下面用两个例子进行说明。

(1) 主存只有一道程序。这个程序每处理文件中的一个记录, 平均需要执行 100 条指令。假定读一个记录需 0.0015 秒, 执行 100 条指令需 0.0001 秒, 写一个记录需 0.0015 秒, 则

平均处理一个记录的时间： $0.0015+0.0001+0.0015=0.0031$ （秒）

处理一个记录 CPU 的利用率:

处理机执行 100 条指令时间/平均处理一个记录的时间=0.0001/0.0031=3.2%

由这个例子可看出，处理一个记录时，CPU 的利用率仅为 3.2%。也就是说，CPU 有

96.8%的时间在等待 I/O 设备进行记录的读和写，这显然是对 CPU 的极大浪费。

(2) 主存同时放三个程序。当主存容量足够大时，为提高 CPU 的利用率，应在主存放足够多的作业，以便当一个作业等待 I/O 时，处理机转去执行其他作业。假设一个计算机系统，它有 256K 主存（不包含操作系统）、一个磁盘、一个终端和一台打印机，主存装有三个作业，分别命名为 JOB1、JOB2、JOB3，它们对资源的使用情况如表 1.1 所示。

由表 1.1 可见，作业 2 主要使用终端进行作业的输入，作业 3 主要使用磁盘和打印机。JOB2 和 JOB3 需要较少的 CPU 时间。

对于简单批处理，这些作业将按顺序执行。作业 1 运行 5 分钟完成，作业 2 等待 5 分钟再用 15 分钟完成。20 分钟后，作业 3 开始执行。30 分钟后三个作业全部完成。

若采用多道程序设计技术，让三个作业并行运行。由于它们运行中几乎不同时使用同类资源，故三个程序可同时运行。在作业 1 进行计算的同时，作业 2 可在终端上进行输入/输出，作业 3 使用磁盘和打印机。这样，作业 1 仍需 5 分钟完成，但在作业 1 结束时，作业 2 已完成三分之一，而作业 3 已完成一半。这样，这三个作业在 15 分钟内将全部完成。显然整个系统处理效率明显提高。

为了更好地理解引入多道程序的好处，下面先引入衡量批处理计算机系统性能指标的几个重要概念：

- 资源利用率：指在给定时间内，系统中某一资源，如 CPU、存储器、外部设备等，实际使用时间或者实际使用容量所占的比率。显然，要提高资源利用率，就必须使资源尽可能忙碌。
- 吞吐量 (Throughput)：指单位时间内系统所处理的信息量。它通常以每小时或每天所处理的作业个数来度量（这里以小时作为度量单位）。
- 周转时间 (Turnaround Time)：指从作业进入系统到作业退出系统所经历的时间。即作业在系统中的等待时间加运行时间。而平均周转时间是指系统运行的几个作业周转时间的平均值。

依据上述定义的三个概念，分别计算上述三个作业单道和多道运行时，处理机和存储器等资源的利用率：

① 单道运行时，只有 JOB1 占有处理机，三个作业运行完成需要 30 分钟。

处理机的利用率为： $5 / (5 + 15 + 10) = 17\%$

存储器的利用率为： $(50 / 256 + 100 / 256 + 80 / 256) / 3 = 30\%$

② 多道时，15 分钟三个作业都完成，其中处理机使用 5 分钟。

处理机的利用率为： $5 / 15 = 33\%$

三个作业共享主存，主存利用率为： $(50 + 100 + 80) / 256 = 90\%$

三个作业单道运行和多道运行时的各资源利用率见表 1.2 中。

由表可知，多道程序运行，使得系统资源的利用率、吞吐量和作业的平均周转时间大大优于单道程序运行，系统性能的改善是明显的。因此，将多道程序设计技术用于批处理系统就构成了多道成批处理系统。

表 1.1 三个作业的执行特点

作 业 编 号	JOB1	JOB2	JOB3
类型	计算型	I/O 型	I/O 型
占用主存	50K	100K	80K
需磁盘情况	NO	NO	Yes
需终端情况	NO	Yes	NO
需打印机情况	NO	NO	Yes
运行时间	5 分钟	15 分钟	10 分钟

表 1.2 多道程序与单道程序时平均资源利用率情况对比

	单 道	三 道 作 业
处理机利用率	17%	33%
存储器利用率	30%	90%
磁盘利用率	33% = 10/30	67% = 10/15
打印机利用率	33% = 10/30	67% = 10/15
三个作业完成运行耗费的时间	30 分钟 = 5 + 15 + 10	15 分钟
吞吐量	6jobs / 小时 = 3 / 0.5	12jobs / 小时 = 3 / 0.25
平均周转时间	18 分钟 = (5 + 20 + 30) / 3	10 分钟 = (5 + 15 + 10) / 3

2. 多道程序设计技术的实现

与简单批处理系统相比，多道批处理的实现必须解决好以下三个问题：

（1）存储器的分配和存储保护。采用多道程序设计技术，主存储器为几道程序所共享。因此，必须提供存储分配手段，使各个程序在主存拥有自己的一个区域。同时各程序在运行时，硬件必须提供必要的存储保护手段，限制它们只能正确地访问自己所占的区域，以避免相互干扰和破坏。特别是不能破坏操作系统。否则，整个系统将无法运行。

现在的微型机中，有些系统，如 MS-DOS 没有存储器保护技术，致使计算机的病毒泛滥成灾，系统遭受严重破坏。

另外，随着程序道数的不断增加，出现了主存容量不够用的问题。为此，必须解决存储器的扩充技术。这就是以后将要介绍的覆盖与交换技术，以及虚拟存储技术等存储器管理技术。

（2）处理机的管理和调度。在单处理机系统中，系统中的多道程序都要竞争使用处理机，处理机管理的主要任务就是实施处理机的分配和调度，以便解决好多道程序之间的转接和有效的运行。现代的计算机系统虽然广泛采用多 CPU 技术，但使用最多的还是集中式处理系统，即系统中只有一个功能强大的主 CPU，并由它完成系统主要的处理任务。因此，处理机管理主要涉及的是对处理机的分配调度策略、实施具体的分配等问题。

（3）系统其他资源的管理和调度。采用多道程序设计技术，系统的资源为多道程序所共享。系统资源除了上面提到的存储器、处理机之外，还有外部设备资源和一些软件即文件资源。因此，系统也要解决好这些资源的管理和调度问题，以使各道程序都能有条不紊地运行下去。

多道程序设计技术的出现，标识着操作系统渐趋成熟，具备了相应的处理机管理、存储器管理、外部设备和文件管理等功能。

1.2.4 分时系统

批处理系统使用多道程序技术后卓有成效地提高了系统资源的利用效率。但这样的系统仍存在以下几个问题：

（1）不能直接控制作业运行。用户一旦把其作业提交给计算机系统，便失去了对作业的控制能力。对于许多作业，如事务处理和一些短小作业的修改，用户希望能提供与系统交互的能力，以便直接控制作业的运行。

（2）作业的周转时间太长。在批处理系统中，用户提交的作业通常要经几小时甚至几天的延迟才能得到所需的结果。这对于仅需要计算时间很短的作业是不利的。

现在，交互计算设施特别是微型计算机已经满足这种要求，但在 20 世纪 60 年代这种情况是不可能的，那时大多数计算机很庞大，价格非常昂贵，代之的是开发了分时系统 (Time-Sharing System)。

在分时系统中，一台计算机系统连接有若干台远近终端（通常终端是带有 CRT 显示的键盘输入设备），多个用户可以同时各自的终端上以交互方式使用计算机。分时系统又叫做多用户多任务操作系统。

1. 分时概念

所谓分时，在引入多道程序技术时就已经有了这个概念，它是以 CPU 与通道、通道与通道、通道与设备并行操作作为条件的。而并行操作又是以分时共享系统资源为基础的：CPU 与通道并行是通过分时使用主存和数据通路等来实现的；通道与通道并行是分时使用 CPU、主存及通道的公用控制部分；同一通道控制的多台 I/O 设备又是分时共享使用主存、通道等的。这种分时是分时使用硬件，属于硬件设计技术。

这里的分时，是指将 CPU 的单位时间（比如 1 秒钟）划分成若干个时间段，每个时间段称为一个时间片 (Time slice)，并按时间片轮流把 CPU 分配给各联机用户使用，这样，每个用户都能在很短时间内得到计算机的服务，彼此感觉不到别的用户存在，好像整个系统为他独占。这样的系统叫分时系统。

早期的分时系统之一是兼容的分时系统 (CTSS)，与后来的分时系统相比，CTSS 非常简单，它的操作控制很容易理解。当 CPU 的控制分配给一个交互用户时，用户的程序和数据装入主存，每 0.2s（时间片）产生一个时钟中断。每个时钟中断产生时，操作系统获得控制，将处理机分给另一个用户。这样，以规定的间隔，当前用户被抢占，另一个用户被装入。为了使被抢占用户以后恢复运行，它的程序和数据被写到磁盘后才装入新用户的程序和数据。利用交换技术，使各个作业在主存与外存之间换入换出，以保证向用户提供合理的响应时间。将多道程序技术应用于分时系统就是功能完善的多道分时系统。

2. 分时系统的特点

分时系统允许多个用户同时共享计算机，它具有以下几个特点：

- (1) 同时性。若干个终端用户可同时使用计算机。
- (2) 独立性。各用户之间彼此独立地占有一台终端工作，互不干扰。
- (3) 交互性。用户从终端键盘上输入各种控制作业的命令，系统响应和处理这些命令，并将处理结果输出显示，用户根据系统显示结果再继续输入。亦即采用一问一答形式控制作业运行。
- (4) 及时性。用户的请求能在较短时间内得到响应。分时系统的响应时间 (response time) 是指从用户发出终端命令到系统响应并开始进行应答所需的时间，是衡量分时系统的主要性能指标。它可以简单地表示为 $T=nq$ ，其中 T 代表响应时间， n 代表同时上机的用户数， q 代表时间片。通常的响应时间为 2~3s。显然，通过修改 n 和 q ，可以获得合理的响应时间。

上述的 4 个特点为程序设计人员提供了比较理想的开发环境，他们可以直接使用计算机，边开发，边调试，边思考，边修改，从而显著地提高了程序开发、调试的效率。

批处理系统与分时系统又称为作业处理系统。用户是以作业为单位提交系统进行处理

的。分时系统与批处理系统是为了适应不同需要而开发的：

- 批处理系统的自动化程度高，通过将各种不同类型的作业进行合理搭配，大大提高系统资源的利用效率。该系统适合处理的作业具有如下性质：比较成熟的、需要耗费较长处理时间的大型作业。
- 分时系统的目标是对用户请求的快速响应。该系统适合处理短小作业。因此，它广泛应用于各种事务处理，并为进行软件开发提供了一个良好环境。

1.2.5 实时系统

计算机不仅能用于科学计算、数据处理等方面，也广泛地用于工业生产的自动控制，军事上的飞机导航、导弹发射，各种票证的预订、查询，银行系统的借贷，以及情报检索等系统。通常把上述系统称为实时控制和实时信息处理系统，这两个系统统称为实时系统（Real-Time System）。

所谓实时，是指计算机对随机发生的外部事件能做出及时的响应和处理。这里的及时是指系统对特定输入做出反应所具有的速度，足以控制发出实时信号的那个设备。

实时系统不同于作业处理系统，它不以作业为处理对象，而以数据或信息作为处理对象，它不接收用户作业，只有几个由外部事件触发的任务。实时系统是一个专用系统，主要用于实时过程控制和实时信息处理。用于实时控制的计算机系统要确保在任何时候，甚至在满负荷时都能及时响应。因此，设计实时系统时，首先要考虑响应的实时性，其次才考虑资源的利用率。

实时操作系统能够在限定的时间内执行完所规定的功能，并能在限定的时间内对外部的异步事件做出响应。执行完规定的功能和响应外部异步事件所需时间的长短，是衡量实时操作系统实时性强弱的指标。有两种类型的实时系统：硬实时和软实时。在硬实时系统中，系统的所有可能的延迟是一定的。对于关键的任务，必须在指定时间范围内完成，否则，可能出现不可预知的错误和危险。在软实时中，即使任务没有在规定时间内完成，也还是允许的。例如，信息查询、多媒体和虚拟现实就属于这一类。

1. 实时系统的主要特点

（1）实时性。由于实时系统接收来自现场的事件，对这种事件的响应速度直接影响到现场过程控制的质量或服务的质量。与分时系统相比，实时系统对响应时间有更严格的要求。分时系统的响应时间通常是以人们能够接受的等待时间来确定的（2~3s），而实时系统则是以被控制过程或信息处理时能接受的延迟来确定的，通常可能是秒的数量级，也可能是毫秒级甚至微秒级。也即系统的正确性不仅依赖于计算的逻辑结果的正确性，而且依赖于结果产生的及时性。

（2）可靠性。实时系统要求硬件和软件措施有非常高的可靠性。因此实时系统要具有容错能力，往往采用双工机制。一台作为主机，用于实时现场控制或实时信息处理，另一台作为后备机与主机并行运行。一旦主机发生故障，后备机便立即代替主机继续工作，以保证系统不间断运行。

（3）确定性。确定性主要取决于系统响应中断的速度，其次取决于系统是否有足够的能力在要求的时间内处理完指定的请求。实时性通常与确定性密切相关。实时性是指系统内核应该保证系统尽可能快地对外部事件产生响应，而确定性是指系统对外部事件的响应性的

最坏时间是可以预知的。

2. 实时系统的功能

(1) 实时时钟管理。实时系统的主要设计目标是提供对实时任务进行实时处理的能力。通常，实时任务分两类：一类是定时任务，它是根据用户规定的时间启动该任务的执行，并按照规定循环周期重复启动执行该任务。另一类是延迟任务，这类任务是推迟一规定的时间后再执行。这两类任务的控制时间，是由实时时钟进行计时控制的。

(2) 简单的人机对话。由于实时系统是专用系统，其交互能力不及分时系统，它一般仅允许与系统中的特定实时任务进行有限制的交互能力，仅允许操作员通过终端访问有限的专用软件，不允许对现有系统软件进行修改。

(3) 过载处理。虽然实时系统设计时，考虑了系统对所有实时任务的实时处理能力，但由于被处理任务进入系统时带有很大的随机性，使得某一时刻系统中的任务数超过了它的处理能力，而产生过载。为此，系统应按照任务的紧急程度排成一个队列，优先处理更紧急的任务，以保证系统在即使出现过载时，仍能正常运行。

1.2.6 嵌入式系统

嵌入式系统(Embedded System)是面向用户、产品、应用的系统，可以定义为以应用为中心、以计算机技术为基础、软硬件可裁剪、适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。凡是计算机的主机嵌埋和应用系统或设备之中，不为用户所知的计算机应用方式，都是嵌入式应用。嵌入式系统运行的几乎都是实时操作系统。

1. 嵌入式系统的技术特点

嵌入式处理器的应用软件是实现嵌入式系统功能的关键。软件要求固化存储，具有高质量、高可靠性，高实时性是嵌入式操作系统的基本要求。制造工业、过程控制、通信、仪器仪表、汽车、船舶、航空、航天、军事装备、消费类产品等，均是嵌入式计算机的应用领域。

2. 嵌入式系统的功能

嵌入式系统是形式多样、面向特定应用的软硬件综合体，其硬件和软件都必须高效地进行量体裁衣式设计。嵌入式系统的运行环境和应用场合决定了嵌入式系统具有区别于其他操作系统的一些特点。

大多数嵌入式操作系统通常是一个多任务可抢占式的实时操作系统核心，只提供基本的功能，如任务的调度、任务之间的通信与同步、主存管理、时钟管理等。其他的应用组件，如网络功能、文件系统、图形用户界面(GUI)系统等均在用户态，以函数调用的方式工作。因而系统都是可裁剪的，用户可以根据需要选用相应的组件，构造自己的专用系统。

1.3 操作系统的功能、服务和特性

批处理系统、分时系统和实时系统是大、中、小型计算机上操作系统所具有的三种形式。这些机器的操作系统往往兼有批处理、分时处理和实时处理三者或其中两者的功能，而形成通用操作系统。如分时和批处理相结合，将分时作业作为前台任务，将批处理作业作为

后台任务，便是分时批处理系统。通用操作系统不仅能满足用户的特殊要求，而且能提高资源的利用率，因此得到广泛应用。

1. 操作系统的功能

操作系统提供了程序执行的环境。从资源管理的观点来看，操作系统的功能应包括：

(1) 处理机管理。在多道程序或多用户系统中，由于处理机数目远远少于运行的作业数，且一个作业可能包含多个算题任务，因此，中央处理机的管理和调度就成为关键问题。特别是在单处理机的情况下，多个程序的并行运行是宏观上的。微观上，处理机在一个时刻只能执行一个作业。因此，不同类型的操作系统将针对各种不同情况采用不同的调度策略，如先来先服务（FCFS）、优先级调度、分时轮转，来提高系统资源的利用率等。

(2) 存储器管理。存储器管理是指计算机的主存管理。主存是计算机硬件中除 CPU 之外的又一个宝贵的资源。如何对主存资源进行统一管理，使多个用户能分享有限的主存和方便存取在主存中的程序和数据，则是存储器管理的主要任务。其次要负责对用户存入主存的程序和数据提供存储保护，保证各用户程序和数据彼此不被破坏。另外，还要解决主存扩充，以便多用户方便地共享主存。

(3) 设备管理。设备管理涉及对系统中各种输入设备、输出设备等的管理和控制问题。这些设备是用户与计算机进行交互的硬件。由于这些设备种类繁多，操作特性各不相同，因此，使得对这些设备管理和控制变得十分复杂。设备管理的主要任务是负责为多用户运行提供方便的运行环境，其中包括分配设备，并按照用户要求控制实现设备的数据传输，完成实际的 I/O 操作。

(4) 文件管理。文件管理又称为信息管理或文件系统。现代计算机系统中，将程序、数据及各种信息资源（包括操作系统及各种实用程序等）组织成文件，长期保存在计算机的磁盘或磁带上。文件管理就是对这样复杂、庞大的软件资源进行存储、检索和保护，以便用户能方便、安全地访问它们。

2. 操作系统提供的服务

由操作系统的功能可总结出它提供的服务：

(1) 用户接口（User Interface, UI）。操作系统是用户与计算机系统之间的接口，用户通过操作系统使用计算机系统。操作系统向用户提供了两种接口方法。一种是命令行接口；另一种是程序级接口。操作系统通过这些接口了解用户的意图，以便更好地服务于用户。

(2) 执行程序。将用户程序装入内存，启动程序运行，并控制程序正常完成或错误终止。

(3) I/O 操作。为请求 I/O 的用户执行设备的启动、驱动和中断处理等。

(4) 文件系统操作。为用户提供操作文件所需的各种操作和功能。

(5) 通信服务。控制进程之间（即多道程序或多用户任务执行时）的同步和通信。

(6) 错误检测和处理。

此外，操作系统为了使用户很好地共享系统资源，保证系统高效运行，还提供如下服务。

(7) 资源分配。为用户的程序分配计算机的各种资源，以保证程序的正常运行。

(8) 记账。不断统计各个用户程序占有的系统资源情况，防止故障程序的运行，使系统有效地工作。

(9) 保护。在多用户多任务系统中，控制用户程序有限制地存取系统资源，以保证资

源状态的一致性。

3. 操作系统的特性

由上述的操作系统功能组成可见，操作系统是一个相当复杂的系统软件。特别是多道程序的引入，为分析和理解操作系统带来了非常大的困难。为了深入研究操作系统，有必要了解一下操作系统的特性，它主要表现在以下几个方面：

(1) 并发性 (Concurrency)。所谓并发性是指为了增强计算机系统的处理能力而采用的一种时间上重叠操作的技术。并发是指系统中存在着若干个逻辑上相互独立的程序或程序段，它们都同时处于活动状态，并竞争系统的各种资源，如 CPU、主存和硬盘等。在单处理机计算机系统中，这种并发执行是宏观上的概念。例如，系统中同时有三个程序在运行，它们可能以交叉方式在 CPU 上执行，也可能是一个在执行计算，一个在进行数据输入，另一个在进行计算结果的打印。为了使这些并发活动（又称为进程）能有条不紊地进行，操作系统必须有效地对其进行管理和控制。允许程序并发活动的系统称为多道程序系统 (MultiProgramming) 或多处理系统 (MultiProcessing)。

(2) 共享性 (Sharing)。支持系统并发性的物质基础是资源共享。资源若不能共享，多任务并发就不能实现；同样，若没有多任务并发，也就没有资源共享。资源共享是操作系统追求的主要目标之一。为了提高计算机系统的资源利用率，更好地共享系统资源，操作系统的各部分功能设计中采用了各种各样的分配调度算法。

(3) 虚拟性 (Virtualization)。为了便于用户程序共享计算机系统的各种资源，操作系统把这些资源的一个物理实体变为逻辑上的多个对应物。如 CPU 的分时使每个用户感觉都拥有一个 CPU；虚拟存储器技术使多个用户程序不必担心存储器的容量就可以很好地共享存储器运行。有关这个性能，以后随时可以理解到。

(4) 异步性 (Asynchronism)。由于系统资源的共享，有限的资源使并发进程之间产生相互制约关系。系统中的各个进程何时执行、何时暂停以及以怎样的速度向前推进、什么时候完成等都是不可预知的。异步性给系统带来潜在的危险，有可能导致系统产生与时间有关的错误。操作系统必须保证有效地防止这些错误的发生。

由于操作系统的并发性、资源共享性及虚拟性和异步性，使得系统变得复杂和不可确定。这些问题将在以后各章中加以研究。

1.4 操作系统的进一步发展

进入 20 世纪 70 年代中期以后，计算机系统结构发生了重大变化，微型计算机（又叫个人计算机）、多处理机相继出现和发展，使计算机出现了大发展、大普及，之后计算机网络、分布式系统、巨型机更是快速发展，促使操作系统技术也有了进一步的发展，产生了各具特点的操作系统。20 世纪 90 年代中期，诺基亚发布了将手机与 PDA 进行整合的 N9000 智能手机，从此面向智能手机的操作系统逐渐发展起来，形成了 Symbian、RIM、iOS 和 Android 等一系列移动操作系统。

1. 个人计算机操作系统

20 世纪 70 年代的个人计算机操作系统既不是多用户系统，也不是多任务系统，它不再

追求系统资源的最大利用，而是考虑极大地方便用户使用和最快的响应速度。它是一个单用户的交互式操作系统，MS-DOS 等就是该时代的代表。个人计算机结构简单，规模小，它以磁盘文件管理为主，配有简单的设备管理，并向用户提供了一组功能丰富的键盘操作命令。20 世纪 80 年代个人计算机上最流行的操作系统是 MS-DOS 和图形工作站上运行的 UNIX 系列的操作系统。20 世纪 80 年代之后，微软公司发布了一系列与 DOS 兼容的具有图形用户界面的新型操作系统，如 Windows NT、Windows 95/98、Windows ME、Windows 2000/XP、Windows Vista、Windows 7、Windows 8、Windows 10 等。

另一类重要的个人计算机操作系统是 UNIX 系列，UNIX 具有很多版本，如 AT&T 公司的 UNIX 系统 V、Sunsoft 公司的 Solaris、HP 的 HP-UX 等。虽然 UNIX 主要用于服务器，但也经常用在个人计算机上。由 Linus Torvalds 编写的基于 MINIX（1987 年发布的一个用于教育的 UNIX 类系统）的 Linux 系统备发展迅猛。由于 Linux 源代码公开，从而吸引了更多的软件开发参与者参与开发、改进，使得该系统具有了更大的发展前景，已经在个人计算机、服务器、嵌入式和移动设备领域广泛应用。

20 世纪 90 年代，多媒体技术已成为个人计算机的重要发展方向。多媒体计算机技术是计算机综合处理多种媒体信息（文本、图形、图像和声音），使多种信息建立逻辑连接，为用户提供一个具有交互性的、更为直观的集成环境，这样的系统要求处理机具有高速的信息处理能力，大容量的主存和海量的外存，高速大容量光缆传输，并为声音和图像的同步提供所需的实时多任务处理能力。总之，多媒体技术是 20 世纪 90 年代计算机的又一次革命，没有多媒体技术的计算机就不是真正的个人计算机。

2. 多处理机操作系统

现在的大多数个人计算机和 workstation 都只包含一个通用的处理机，任何时候只能运行一个程序。多处理机（Multi-Processors）系统就是由多个处理机组成的计算机系统，各处理机采用紧耦合方式进行连接，共享主存。这样的系统又叫做并行系统。

多处理机操作系统有两种模式：

（1）非对称多处理（ASymmetric MultiProcessing, ASMP）操作系统。通常指定一个处理机运行操作系统，其他处理机运行用户作业。运行操作系统的处理机为其他处理机分配和调度任务，这是一个主从模式。ASMP 操作系统特别适合在非对称硬件上运行，例如一个处理机带一个附属的协处理器或者两个并不共享所有可用主存的处理机一类的硬件。这种系统，一旦运行操作系统的处理机出现故障，整个系统就崩溃了。

（2）对称多处理（Symmetric MultiProcessing, SMP）操作系统。在对称多处理系统中，操作系统和用户程序可安排在任何一个处理机上运行，或者同时所有处理机上运行，各处理机共享主存和各种 I/O 设备。目前几乎所有的现代操作系统（如 Windows、Max OS X 和 Linux 等）都支持 SMP，例如 Windows 7 的专业版、企业版和旗舰版都以 SMP 方式支持两颗 CPU。对称多处理系统比单处理机具有更多的潜在的优势：

① 增加了系统的吞吐率。多个作业可以分配在任何一个处理机上执行，大大增加了系统的吞吐率。

② 增加了系统的可靠性。一个处理机的失效，只是性能的降低，不会影响整个系统。

与多处理器发展相关的另一个方向是多核（multicore）处理器。多核，又称单芯片多处理器（Chip MultiProcessor, CMP），是在一个 CPU 芯片内部集成多个计算核，每个计算核内

部具有一套完整、独立的执行部件，以及寄存器组和高速缓存等部件，不同计算核之间还有一些共享的高速缓存。这主要是因为随着 CPU 主频的提高，带来了 CPU 性能不能随着主频提高的现象，而且随着功率提高，散热问题越来越严重。因此，CPU 设计向更易于扩展的多核发展方向，例如 Intel 生产的 Core i7 处理器采用多核和超线程技术（HT），可以达到 4 核 8 线程或者 6 核 12 线程。图 1.6 所示为典型的具有三级缓存的 4 核处理器。

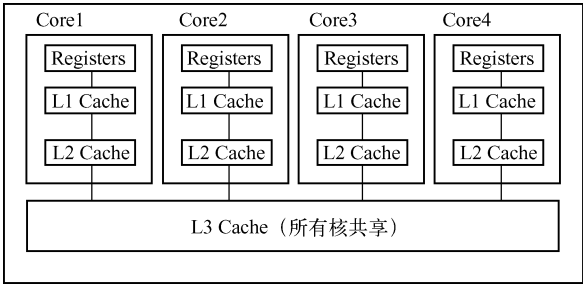


图 1.6 典型的具有三级缓存的 4 核处理器

多核处理器的设计可以更好地支持并行化的程序运行方式，应用程序可以采用多线程的并行编程模式或者多进程的模式来应用多核处理器的并行执行能力。同时多核处理器也给操作系统的设计引入了一些需要关注的问题，如操作系统对多线程的支持方式、多核处理器线程调度方法、多核间 Cache 一致性等。关于多核处理器操作系统设计的相关问题将在后续章节中具体阐述。

3. 网络操作系统

计算机网络是通过通信设施将地理上分散的具有自治能力的计算机系统连接起来的松耦合的系统。在计算机网络中的用户可以共享网络系统中的资源，彼此进行通信和信息交换，但要求通信双方必须清楚相互的位置。

网络操作系统是为计算机网络配置的操作系统，网络中的各台计算机配有各自独立的操作系统，网络操作系统把它们有机地联系起来，其主要功能是为网络中各台计算机提供通信和网络资源的共享。因此，网络操作系统除了具有常规操作系统所应具有的处理机管理、存储器管理、设备管理、文件管理的功能外，还具有网络管理的功能。它是在各种计算机操作系统之上按网络协议（如 TCP/IP）标准开发的软件。网络管理功能主要包括：

（1）提供高效、可靠的网络通信能力。除了支持终端与计算机之间的通信外，还应支持网络中各计算机之间的通信。

（2）提供多种网络服务。在网络协议控制下，各计算机之间可以协同工作，实现用户需要的网络服务。例如：

- ① 文件传输服务：用于将一个计算机上的文件传输到另一个计算机上，以便共享。
- ② 分时系统服务：使网络上的远程用户也能像本地用户一样，使用其上的分时系统。
- ③ 远程作业录入服务。
- ④ 远程打印服务。
- ⑤ 电子邮件服务等。

因此，用户可以利用上述服务设施，灵活地访问网络中各计算机上的文件系统、分时系统和批处理系统等，方便地共享网络中的各种软、硬件资源。

网络操作系统的结构模式有客户/服务器模式和对等模式两种。对于客户/服务器 (Client/Server) 模式, 系统有一个功能和资源配置完善的服务器, 其他计算机都向该服务器提出服务请求, 对于对等模式 (Peer-to-peer), 各个站点是对等的, 既可作为客户请求服务, 也可作为服务器, 向其他计算机提供服务。

4. 分布式操作系统

分布式计算机系统是由多个分散的计算机经互连网络连接而成的统一的计算机系统, 其中的各计算机既高度自治又相互配合, 能在整个系统范围内实现资源管理、资源共享、信息交换和协同执行任务。粗看起来, 分布式系统与计算机网络系统没有多大区别, 但就其实现功能来讲, 仍具有明显的区别:

(1) 计算机网络有国际标准化组织 (ISO) 制定的网络互连体系结构及一系列标准通信网络协议。而分布式系统没有制定标准协议。

(2) 分布式计算机系统是多机系统的一种新形式, 它强调资源、任务、功能和控制的全面分布。就资源分布而言, 既包括处理机、辅助存储器、输入/输出系统、通信接口等硬设备资源, 也包括程序、数据、文件等软件资源, 它们分布在各个物理上分散的场地, 各场地经互连网络相互通信, 构成统一的计算机系统。这种系统的工作方式也是分布的。分布的原则有两种:

① 任务分布是指把一个计算任务分成多个可并行执行的子任务, 分配给各场地协同完成。

② 功能分布是指把系统的总功能分划成若干子功能, 由各场地分别承担其中的一部分或几分子功能。

由此可见, 分布式系统要求连网的多机有一个统一的操作系统, 实现系统的统一操作性。为了把数据处理系统的多个通用部件合成一个具有整体功能的系统, 必须引入分布式操作系统。为了保持各机的自治性, 各处理机有自己的私有操作系统。对于系统中各物理资源的管理, 分布式操作系统与各机的私有操作系统之间, 不允许有明显的主从管理关系。

总之, 分布式计算机系统是由若干可分离、自治的, 但彼此间又相互通信、协同完成同一任务的小型或微型机, 用互连网络连成的一种新型计算机系统。它既能利用原单机软硬件功能进行分布处理, 又能互连成一个整体, 协同配合实现并行处理。其基本特征是:

① 软硬件结构上具有模块性; ② 工作方式上具有自治性; ③ 系统功能上具有协同并行性; ④ 对用户具有透明性; ⑤ 系统的容错性和坚固性。

分布式操作系统是负责分布式计算机系统资源分配和调度、任务划分、信息传输和控制协调工作的软件, 其目的是向用户提供一个使用方便、友好的分布式用机环境。同时, 尽可能提高整个系统的利用效率。

5. 虚拟化与云计算

虚拟化 (Virtualization) 与云计算 (Cloud Computing) 是最近一些年经常同时出现的计算机术语。虚拟化可以认为是实现云计算的基础技术, 主要是指将一台物理计算机变成多台虚拟计算机, 每台虚拟计算机 (简称虚拟机, Virtual Machine) 可以运行独立的操作系统和应用程序, 实现不同操作系统之间的隔离。虚拟化技术早在 20 世纪 60 年代就用在了 IBM 大型机上, 以便多个用户并发执行任务。现在, 虚拟化的概念已经被扩展到了很多的方面,

例如 CPU 虚拟化、存储虚拟化、网络虚拟化、应用虚拟化、桌面虚拟化等。虚拟化的核心思想就是通过虚拟机监视器（Virtual Machine Monitor, VMM）（VMM 也称为 hypervisor）产生虚拟机，并处理虚拟机对底层硬件资源的请求，让虚拟机获得虚拟化的硬件资源。当采用了虚拟化技术之后，在计算机硬件上就运行了多个操作系统，其中虚拟机中运行的操作系统称为客户机操作系统（GuestOS）。如果 VMM 运行在主机的操作系统上，则该操作系统称为宿主操作系统（HostOS）。根据 VMM 是否运行在硬件上，虚拟化模式大体上可以分为以下两类：

（1）裸金属(bare metal)虚拟化模型

裸金属虚拟化模型也称为 Type I 型。在该模型中，Hypervisor 直接运行在没有安装操作系统的物理硬件上，拥有最高的特权等级，直接管理底层硬件资源；Hypervisor 之上安装虚拟机，虚拟机上的 Guest OS 通过 Hypervisor 访问底层硬件资源。此种模型由 Hypervisor 直接管理硬件资源，性能损失较少，整体性能较好，因此几乎所有的企业级数据中心都倾向采用此种类型的虚拟化模型。Type I 型的虚拟化产品包括 VMware vSphereESXi、Microsoft Hyper-V、Citrix Xen Server 等。

（2）宿主(hosted)虚拟化模型

宿主虚拟化模型也称为 Type II 型。在该模型中，Hypervisor 需要运行在宿主操作系统之上，就像一个普通的用户进程一样。虚拟机运行在 Hypervisor 之上，Guest OS 对底层硬件资源的访问需要被 Hypervisor 拦截，然后转交给 Host OS 进行处理。因此 Guest OS 访问资源需要经过的路径更长，性能损失更严重。采用此种模型的虚拟化产品包括 VMware Workstation、Oracle VirtualBox 等。两种虚拟化模型如图 1.7 所示：

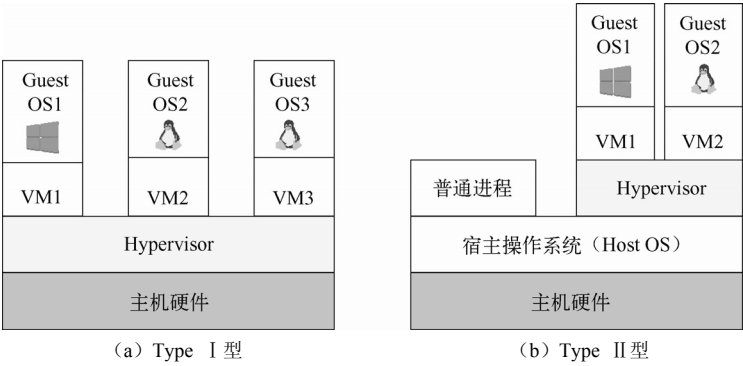


图 1.7 虚拟化模型

虚拟机监视器针对处理器、内存和 IO 设备可以采用不同的虚拟化技术。针对处理器来说，常用的虚拟化技术包括全虚拟化（Full Virtualization, FV）、半虚拟化（Paravirtualization, PV）和硬件辅助虚拟化（Hardware-Assisted Virtualization）三种。

全虚拟化是指 VMM 为虚拟机模拟了完整的底层硬件，虚拟机操作系统不知道自己是运行在虚拟化环境上，像普通操作系统一样访问底层硬件，这些资源的访问会被 VMM 捕获，然后经过二进制翻译等技术手段完成资源的访问，并返回给虚拟机操作系统。全虚拟化中的虚拟机操作系统不需要做修改，适用性较好，但是由于需要额外的指令翻译，性能会受到影响。

半虚拟化是指为虚拟机操作系统提供一个经过修改的硬件抽象，以 Hypercall（由 VMM 给虚拟机操作系统提供的类似系统调用的接口）的方式来调用底层硬件。要求虚拟机操作系统要修改一部分内核，与 VMM 配合工作，这样就省去了异常捕获和指令翻译的过程，性能损失比较少。虚拟化技术 Xen 主要就是采用这种半虚拟化的方式支持虚拟机运行。

硬件辅助虚拟化是指硬件厂商为了更好地支持虚拟化技术，进行了硬件层面的扩展，例如 Intel VT-x 和 AMD-V 两种技术，将原来处理器支持的四种特权等级（Ring 0、1、2、3）进行扩展，增加了一个新的特权等级，称为 Root mode，也有人称为 Ring -1。VMM 可以运行在 Root mode，Guest OS 仍然可以运行在 Ring 0。这样就不再需要通过 VMM 进行指令翻译等过程了，也不需要 Guest OS 做修改，性能损失也很少。目前主要的 Intel 和 AMD 处理器都支持虚拟化，很多虚拟化厂商的产品都已支持硬件辅助虚拟化模式。

各个软硬件厂商、研究机构都热衷于采用虚拟化技术，这主要是由于虚拟化技术在以下方面具有优势：

- 计算机硬件性能显著提升后，采用虚拟化技术能够使用较少的服务器支持更多的应用，进而提高资源利用率；
- 采用虚拟化技术能够比较容易地实现冗余系统，简化灾难恢复；
- 能够显著减少开发、测试环境的构建时间；
- 能够比较容易地实现应用环境的迁移和部署。

传统虚拟化技术需要在每个虚拟机中运行 Guest OS，这很好地实现了应用程序的隔离，但是也带来很大的性能损失。多数情况下，用户并不需要一个功能完整的 Guest OS，只是需要为应用程序提供良好的隔离环境，因此就有了容器（Container）技术。容器技术可以看作一种新型的轻量虚拟化技术。

1.5 用户与操作系统的接口

操作系统是用户与计算机系统之间的接口。用户在操作系统的帮助下，可以安全、可靠、方便、快速地使用计算机系统完成自己的各种任务。操作系统如何了解用户的意图，以便更好地服务用户呢？为此，操作系统向用户提供了使用计算机系统的接口，简称用户接口（User Interface，UI）。有两种接口方法：一种是提供给操作计算机的用户的操作接口，用户利用该接口来组织和控制作业的执行；另一种是提供给编程人员使用的低级接口——系统调用，编程人员在程序中利用该接口，向操作系统提出资源请求和一些功能服务。

1.5.1 用户与操作系统的操作接口

1. 命令解释程序（Command Interpreter）

操作接口是以命令行的形式出现的，通过命令解释程序解释执行。有些操作系统将命令解释程序作为操作系统内核的一部分，如 DOS 操作系统。有些操作系统把命令解释程序当做一个特殊的程序，当用户登入系统或启动一个作业时，命令解释程序被激活运行。UNIX 系统把它叫做 Shell。这种接口提供的是一组操作命令或作业控制语言。用户

使用这些命令，组织和控制作业的执行。命令解释程序的主要功能是获得用户的命令并解释执行。

命令解释程序有两种实现方法。

(1) 命令解释程序本身包含了执行这些命令的代码。这种情况下，由于每个命令都对应自己的执行代码，因此，提供的命令个数决定了命令解释程序的大小。如 DOS 操作系统的命令解释程序就是这样实现的。

(2) 由操作系统核心实现命令要求的功能。这种情况下，命令解释程序只是获得命令，并检查命令和参数的正确性，之后它调用实现命令的操作系统代码，完成命令的执行。这样命令解释程序较小，而且增加新的命令时，命令解释程序不受影响。UNIX 操作系统的 shell 命令解释程序就是这样实现的。

2. 图形用户接口 (GUI)

另一个用户友好的接口就是图形用户接口 (Graphic User Interface, GUI)。这是基于鼠标 (mouse) 的窗口和菜单系统。通过移动鼠标和单击按钮来定位屏幕上代表文件、程序和系统功能的映像或图标，用于运行一个程序、打开一个文件或目录或下拉一个命令菜单等。

究竟选择命令行还是 GUI 方式，这完全取决于个人的喜好。大多数 UNIX 系统的用户喜欢采用命令行接口方式，因为该系统提供了功能强大的 Shell 接口，而且命令方式在远程连接到服务器的时候往往更便捷。而 Windows 用户则喜欢采用 Windows 的 GUI 环境，不喜欢使用 DOS 的命令行接口。

3. 作业运行的控制命令

当用户使用批处理系统或分时系统时，如何控制其作业运行呢？也有以下两种控制方式。

(1) 脱机作业控制命令。这是批处理系统提供给用户对作业控制时使用的命令。当进入系统的批处理作业被作业调度选中时，系统将按照用户提供的作业控制命令控制作业的执行。

作业控制语言通常包含两种类型的语句。一类是作业及其申请资源的说明语句，另一类是实现作业控制和具体操作功能的执行性语句。它大致应反映如下几方面要求：① 作业标识、作业说明，以及调度要求的参数；② 资源申请（主存、外设、CPU 等）；③ 程序和数据的控制功能；④ 调试功能；⑤ 控制转移功能；⑥ 作业撤销等。

通常，这类语言是解释执行，而不是编译执行的。负责解释作业控制语言语句的程序叫做作业解释程序或命令解释程序。

(2) 交互式的作业控制命令。它提供给用户进行交互式作业或进程控制时使用的命令。实现这种方式的控制，系统提供了丰富的联机控制命令，又叫键盘命令，即 Shell 命令。

联机控制命令具体包括的功能如下：① 作业控制执行的命令（建立、修改、编译、连接、运行）；② 资源申请（各种外设的使用和重定向）；③ 文件的各种操作命令；④ 目录操作命令；⑤ 控制转移命令等。

每当一个新的用户登录系统，或开始一个新作业或进程时，命令行解释程序自动执行。等待用户输入命令并进行解释和执行。该方式下，系统的执行过程如图 1.8 所示。

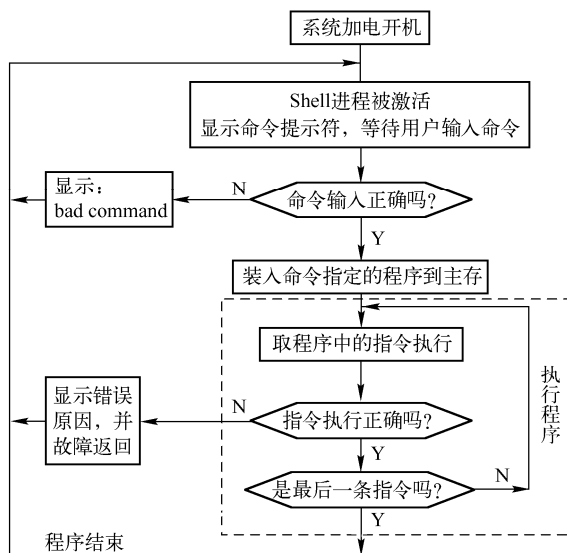


图 1.8 交互命令方式的程序执行过程

现代操作系统不再使用真正的批处理策略，但为了减少联机用户重复输入命令的负担，允许用户将一组复杂的操作系统命令组织成一个控制文件，然后批处理这个控制文件中的命令，使联机用户也可享受到自动批处理操作方式减轻用户负担的好处。

1.5.2 系统调用接口

系统调用（System Call）提供了用户程序与操作系统内核的接口。所谓系统调用，就是操作系统内提供的一些子程序。编程人员通过使用系统调用命令，向操作系统提出资源请求或获得系统的一些功能服务，以取得操作系统的服务。系统调用接口通常是通过库函数调用映射进入操作系统的功能子程序的。

1. CPU 的两种操作方式

由于计算机系统有许多用户，每个用户可能有若干个任务并发地在系统中执行。为了防止用户程序破坏操作系统或直接操作硬件，必须严格区分操作系统代码和用户代码。为此，许多处理机在处理机状态字（Processor Status Word, PSW）或者特定寄存器（例如 x86 处理器就是在 CS 寄存器中增加执行方式位来区分两种操作方式：用户态（又称目态）（User Mode）和核心态（Kernel Mode）。核心态又叫管理态（Supervisor Mode）或特权态（Privileged Mode）。例如 Intel 的 x86 架构处理器提供了 0 到 3 四个特权级，数字越小，特权越高，Linux 操作系统中主要采用了 0 和 3 两个特权级，分别对应的就是内核态与用户态。x86 处理器中的 CS 寄存器的低 2 位（Current Privilege Level, CPL）用来表示处理器当前工作的特权级。处理器工作的特权等级决定了所能执行的操作：

- （1）执行指令的特权性（指处理机能执行何种指令）；
- （2）存储器访问的特权性（指现行指令能访问存储器的哪些单元）。

为此，系统把 CPU 的指令集划分为特权指令（Privileged Instructions）和非特权指令。所谓特权指令是指关系系统全局的指令，如存取和操作 CPU 状态、启动各种外部设备、设置时钟时间、关中断、清主存、修改存储器管理寄存器、改变用户方式到核心方式和停机指

令等。这类指令只允许操作系统使用，不允许用户使用。

操作系统程序运行在核心态。在核心态下，允许执行处理机的全部指令集（包括特权和非特权指令），访问所有的寄存器和存储区。

用户程序运行在用户态。在用户态下，只允许执行处理机的非特权指令，访问指定的寄存器和存储区。如果在用户态下企图执行一条特权指令，CPU 就视其为非法指令，故障终止其运行。

2. 系统调用功能

为了满足用户程序请求操作系统服务或请求系统资源，操作系统提供了功能丰富的系统调用命令。用户程序通过调用操作系统的这些服务程序，控制程序的启动、执行和使用系统中的各种资源和提供的服务。通常系统调用命令包括：

（1）进程控制。包括程序的正常和异常终止、程序的装入和执行、进程的创建和删除、得到和设置进程的属性、分配和释放主存，以及进程之间同步和通信等。

（2）文件管理。包括创建和删除文件、打开和关闭文件、读写和移动文件、得到和设置文件属性等。

（3）设备管理。包括请求和释放设备、读写设备、得到和设置设备的属性，以及连接和卸下设备等。

（4）其他服务。包括得到和设置系统的时间和日期、发送和接收消息等。

对于不同的操作系统，它们所提供的系统调用命令的条数、调用格式和所完成的功能不尽相同，这正如不同型号的计算机有不同的指令系统一样。一般来说，操作系统可提供的系统调用有数十条，乃至数百条。

系统调用很像预定义的函数或子程序，一般用在汇编语言指令中。有些系统调用直接用在高级语言程序中。

系统调用命令可以看成是机器指令的扩充。因为从形式上看，执行一条系统调用命令就好像执行一条功能很强的机器指令，所不同的是机器指令是由硬件执行的，而系统调用命令是由操作系统核心解释执行的。但从用户来看，操作系统提供了系统调用命令之后，就好像扩大了机器指令系统，增强了处理机的功能。因此，呈现在用户面前的是一台功能强、使用方便的虚拟处理机。

系统调用常常以 API 的形式出现。常用的有两种 API，即 Windows 的 Win32 API 和基于 POSIX 系统的 POSIX API（包括 UNIX 和 Linux 等）。系统维护一个系统调用接口表，每个系统调用都有一个名字，每个名字用一个数字表示，该数字就是查找这个表的索引。

3. 系统调用的执行

从用户角度看，用户使用系统调用与一般的子程序调用非常相似，它们的执行会改变指令的执行流程，而且都可以共享和嵌套调用。一般的过程调用不涉及系统状态的转换，而系统调用的执行将使 CPU 的执行方式发生变化，将由用户态转换为核心态，执行相应的系统调用程序。完成系统调用功能后，CPU 再由核心态转换为用户态，返回用户程序继续执行。

虽然系统调用命令的具体格式因系统而异，但从用户程序进入系统调用程序的步骤及其执行过程来看，却大致相同。通常，用户必须向系统调用命令处理程序提供必要的参数，以便根据这些参数进行相应的处理。用户程序执行到系统调用命令时，硬件把它作为一个软

件中断对待，控制通过中断向量传递给操作系统的一个服务例程，改变用户方式到核心方式。这种方式又称为异常（Exception）或陷入（Trap）。

当执行一个系统调用时，它检查中断指令以决定出现了什么系统调用。所带参数决定了用户调用的服务类型并使用寄存器、堆栈（即一块主存区）进行参数传递。在验证参数的合法性后，去完成特定的系统调用功能。需要指出的是，这个被调用的系统子程序可能还要调用其他的子程序完成更基本的子功能，从而形成系统功能的嵌套调用。当系统调用执行完成时，系统要把执行是否成功，以及成功时的执行结果再回送给调用者。之后恢复处理机执行系统调用前的状态——用户态，返回到用户程序继续执行。系统调用的执行过程如图 1.9 所示。

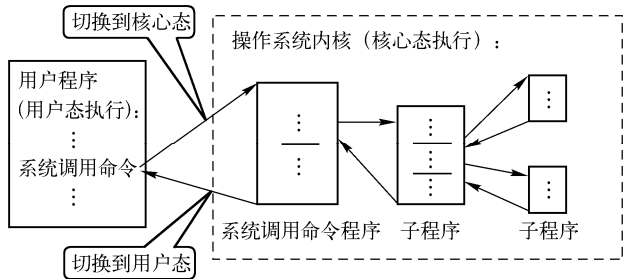


图 1.9 系统调用的执行过程

系统调用执行的一般过程可以概括为以下几步：

- ① 通过陷入（或异常）使系统切换到核心态，这需要硬件提供相应的支持。一旦 CPU 进入核心态，就可以按特权方式运行。
- ② 将程序计数器和处理机的当前状态存入任务的堆栈中。
- ③ 将系统调用号存入核心堆栈中。
- ④ 执行汇编代码来保存通用寄存器中的内容。
- ⑤ 调用相应的操作系统例程来完成系统调用。
- ⑥ 返回到用户方式。

1.6 操作系统的运行方式

由上一节的描述可以看出，操作系统通过两种模式的切换来保护自身免受应用程序代码的损害。由用户态切换到内核态主要发生在以下三种情况：中断、异常和系统调用。上一节介绍了系统调用的模式切换过程，本节简单描述中断和异常的基本原理。可以说操作系统内核代码的运行是通过中断或异常驱动和激活的。中断，又叫做外中断，是一些外部设备向 CPU 泄放的一个硬件信号，如各种外部设备的 I/O 完成中断、时钟中断等，它与 CPU 执行的指令无关，是异步发生的，分为可屏蔽或不可屏蔽处中断。当中断产生时，CPU 响应中断，转入相应的中断处理程序执行。

异常又叫做陷入，是执行程序自己产生的，是同步发生的，而且是不可屏蔽的。其产生的条件是：① 由于程序错误，如程序的非法操作码、地址越界、除数为 0，包括存储器管理中的页面失效等；② 由于程序请求操作系统服务和请求系统资源等使用系统调用。一旦产生，立即转操作系统处理。

操作系统内核代码运行在核心态，主要是在中断或异常发生时嵌入到用户进程中运行的。当中断或异常发生时，保护当前进程的运行现场信息，将 CPU 的控制权交给操作系统内核。操作系统根据中断和异常产生的原因，进行相应处理。处理完成后，或者恢复被中断进程的现场，或者另选一个进程运行。因此由内核态切换回用户态主要发生在以下四种情况：完成中断或者异常的处理、新进程或者线程启动、进程调度选择执行新的进程、操作系统向进程发送信号。中断、异常和信号等详细内容参见第 12 章。

1.7 操作系统的设计规范和结构设计

1. 操作系统的设计规范

一个高质量的操作系统应具有高效性、可靠性、易维护性、可移植性、安全性、可伸缩性和兼容性等特征。到目前为止，还没有一个统一的标准来衡量一个操作系统。从以下几个方面描述操作系统的设计规范：

(1) 系统效率：是操作系统的一个重要性能指标。体现系统效率的指标在前面已经陈述过，它包括资源利用率、吞吐量和周转时间及响应时间等。

(2) 系统可靠性：是指系统能发现、诊断和恢复硬件和软件故障的能力。通常用下面几个指标进行说明：

① 可靠性 R (Reliability)：通常用系统的平均无故障时间 (Mean Time Between Failures, MTBF) 来度量。它是指系统能正常工作的平均时间值。 R 越大，系统可靠性越高。

② 可维护性 S (Seviceability)：通常用平均故障修复时间 (Mean Time to Repair Fault, MTRF) 来度量。它是指从故障发生到故障修复所需的平均时间。 S 越小，可修复性越高。

③ 可用性 A (Availability)：是指系统运行的整个时间内，能正常工作的概率。它由下面的公式计算：

$$A = \text{MTBF} / (\text{MTBF} + \text{MTRF})$$

(3) 可移植性：是指把一个操作系统从一种硬件环境移植到另一种硬件环境时系统仍能正常工作的能力。移植时，代码修改量越少，效率越高。

(4) 可伸缩性：是指操作系统对添加软硬件资源的适应能力，尤其是指可能添加到硬件中的 CPU 资源的能力，也即操作系统可运行在不同种类的计算机上，从单处理器到多处理机的系统上。

(5) 兼容性：主要指软件的兼容性，是指操作系统执行为其他操作系统或为同一系统的早期版本所编写的软件的能力。

(6) 安全性：是指操作系统应具有一定的安全保护措施，如账号检查、系统接入检测、各用户资源分配和资源保护、用户的资源不受他人侵犯等。

2. 操作系统结构设计

操作系统是一个非常复杂的大型软件。为了使它的功能正确而又便于修改和维护，必须进行细心的工程设计。通常，首先应该划分成几个相对独立的任务，再将任务划分成许多小的模块。每一个模块都应该是一个很好定义的小系统。各个模块之间如何交互连接而构成一个性能良好的操作系统，这是下面要讨论的操作系统的结构设计问题。

(1) 单块式内核 (Monolithic kernel) 设计方式

单体式内核，有时也称宏内核 (Macrokernel)，是指整个操作系统的全部或者大部分模块运行在内核空间内，模块之间可以直接调用，不用进行内核态到用户态的转换。许多操作系统设计之初没有在结构划分上下很大工夫，而是先构造一个小的、具有有限功能的简单的操作系统内核，之后根据需要再逐步添加。显然，这样的系统设计方法本身并不够优雅，模块之间依赖较多，会造成系统复杂，不易维护。图 1.10 所示的 DOS 操作系统就是这样一个例子。

DOS 操作系统表面看划分了层次，但实际是个单块结构，不提供硬件保护。因为系统中的任何一层程序都相互调用，特别是用户的应用程序能够直接访问硬件。因此，易导致 DOS 系统病毒泛滥，进而引起整个系统崩溃。同时，由于操作系统各模块之间可以直接调用，使得系统内核模块之间的通信开销非常小，因而单体式操作系统内核的性能较好。UNIX、Linux 和早期的 Windows 采用了单体式内核设计方法。

(2) 层次式的结构设计

随着硬件功能的增强，操作系统可以划分成若干功能独立、更加合理的小模块，从而使操作系统能够获得对计算机的很好控制。通常按照应用的要求，采用自顶向下或者自底向上的方式进行分层设计。每一层都应该向上层提供独立于下层的软件接口，最终实现操作系统的功能。

UNIX 操作系统就是采用分层设计的典范。另外，它将 CPU 的执行模式分为用户态和核心态，从而保护操作系统不受用户程序的破坏，使系统具有健壮性和安全性。图 1.11 给出了 UNIX 操作系统大致分层的情况。

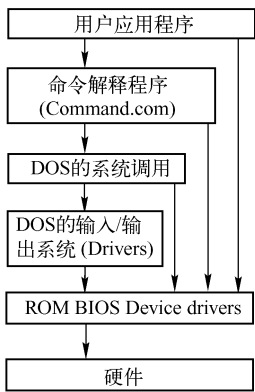


图 1.10 MS-DOS 分层结构

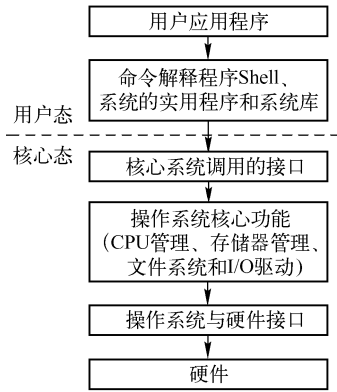


图 1.11 UNIX 系统分层结构

由于它的绝大部分代码用 C 语言编写，它的可移植性较好。POSIX 的操作系统标准就是依据 UNIX 操作系统标准制定的。

(3) 基于微内核 (Micro kernel) 的 client/server 的设计

随着操作系统功能的不断扩充，系统内核越来越庞大，管理和维护越来越困难。卡内基梅隆大学于 1980 年开发了一个基于微内核的操作系统 MACH，该系统的特点是操作系统内核只保留最基本的功能，如只提供进程 (线程) 调度、消息传递、存储器管理和设备驱动等。而将其他功能，包括各种 API、文件系统和网络等许多操作系统服务放在用户层以服务的形式实现。

client 程序和 server 服务都在用户空间运行。微内核的主要功能就是提供 client 程序和

server 服务之间的以消息传递方式的通信设施。如果一个 client 程序希望访问文件时，它必须与文件服务器进行交互。这是通过向微内核发消息的形式间接交互实现两者的请求和服务功能的，如图 1.12 所示。

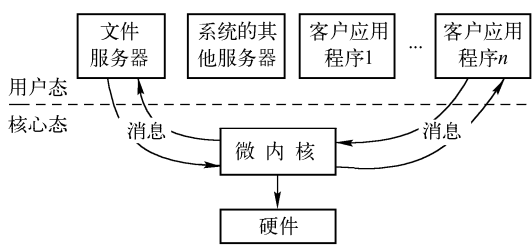


图 1.12 client/server 操作系统结构

微内核的好处是便于扩充操作系统的功能。这是因为新增加的功能放在用户空间，不需要修改操作系统内核。由于微内核是一个小内核，对操作系统的修改几乎没有什么影响。这样设计的最大好处是，使得操作系统很容易从一种硬件平台移植到另一种平台上。另外，微内核也更加安全和可靠。因为大多数服务是作为一个用户任务运行的，如果一个服务出现故障，操作系统的其他部分仍可以正常运行而不受影响。

微内核操作系统曾经被广泛研究，用于解决宏内核存在的诸多问题。例如 QNX、Minix、seL4 等都采用了微内核设计方法。

由于微内核将大量操作系统服务放在用户空间，消息传递增加了系统开销，使系统性能降低，早年计算机处理性能比较弱，所以这种设计方法没有被成熟操作系统广泛采用，但是随着硬件性能的提升，微内核设计方法也有可能再次得到发展，例如目前谷歌正在研制的 Fuchsia 操作系统内核 Zircon，就采用了微内核的设计方法。

(4) 混合结构操作系统内核设计

虽然在操作系统领域一直都存在着关于单体式结构和微内核结构设计方法之争，这就好像讨论 CISC 和 RISC 两种体系结构的优劣一样。实际上，为了更好地解决性能、安全性、可用性和可扩展性等问题，很多主流操作系统采用了混合结构，例如 Windows NT 之后的操作系统（Windows 8、Windows 10 等），以及 Mac OS X 都采用了混合结构。

1.8 小 结

本章重点讨论了操作系统的基本概念、分类、功能、提供的服务，以及操作系统的运行环境等。

● 操作系统的三个基本类型：

- ① 批处理系统。它是在解决人机矛盾和 CPU 与 I/O 设备速度不匹配的矛盾过程中形成的，其特点是提高系统资源的利用率，增加系统的吞吐量。其缺点是用户无法与作业交互，作业的平均周转时间长。
- ② 分时系统。它克服了批处理系统的缺点，使用户与其作业直接交互，其特点是具有同时性、独立性、交互性、及时性，但其系统资源利用率不如批处理系统高。
- ③ 实时系统。它的主要特点是计算机能够在限定的时间内执行完所规定的功能，并能

在限定的时间内对外部的异步事件做出响应。它具有实时性和确定性、高度的安全可靠。嵌入式系统实际是实时系统的一个分支。

- 从资源管理的观点来看，操作系统具有处理机管理、存储器管理、设备管理和文件管理等。为了实现这些管理，操作系统提供了多种服务。

- 多道程序的概念及特点：

多道程序指主存存放多个用户程序，使这些程序同时处于运行状态。其特点就是主存有多道、宏观上并行、微观上串行。

- 批处理操作系统的性能指标指资源利用率、系统吞吐率和作业的平均周转时间等。
- 操作系统的三个基本类型的特征是并发性、共享性和虚拟性等；关于操作系统的运行通常或者由中断和异常激活，或者作为一个独立的进程运行。
- 操作系统的进一步发展，形成了多种类型的操作系统：

① 微型机操作系统，从简单的单用户、单任务到复杂的多用户、多任务系统等多种类型。

② 多处理机 and 对称多处理操作系统，多处理机共享主存，提高系统的吞吐量和系统的可靠性。

- ③ 网络操作系统，将各计算机有机地联系起来，使各计算机间实现通信和资源共享。

④ 分布式操作系统，是经网络连接的多机统一的操作系统，它强调资源、任务、功能和控制的全面分布，以提高整个系统的资源利用效率。

- 用户与操作系统之间的接口有两种：

① 操作接口，提供给联机用户操作的是 Shell 操作命令和图形用户接口，提供给脱机用户操作的是作业控制语言。这些接口命令由系统提供的命令解释程序进行解释执行。

② 编程接口——系统调用。用户程序利用系统调用命令请求系统资源和得到系统核心的服务。

- 本章还对操作系统的运行环境、设计规范、结构设计进行了讨论，应了解操作系统的运行特点和组成结构。

习 题

1-1 如何定义一个操作系统？

1-2 早期操作系统设计的主要目标是什么？

1-3 何谓作业和作业步？

1-4 引入多道程序设计的主要目的是什么？什么是多道程序设计？它有哪些特点？

1-5 叙述操作系统的基本功能和提供的服务。操作系统具有哪些特性？衡量一个批处理操作系统有哪些性能指标？

1-6 解释概念：吞吐量、周转时间和响应时间。

1-7 批处理系统、分时系统和实时系统各有什么特点？各适合应用于哪些方面？

1-8 何谓用户与操作系统之间的接口？在用户与操作系统之间通常提供哪几种类型的接口？其主要功能是什么？

1-9 设计一个操作系统应遵循哪些规范标准？

1-10 常用的操作系统结构设计有哪些分类方法？

1-11 什么是对称多处理？它有什么好处？

1-12 应在核心态运行的指令是（ ）。

- A. 设置处理机运行状态 B. 读实时时钟 C. 设置实时时钟
D. 关中断 E. 改变内存页映射 F. 读操作系统的某些数据

1-13 为了实现系统保护，CPU 设计有多种状态，常用的是哪两种状态？各种状态下执行什么程序？什么时候发生状态转换？

1-14 操作系统的设计目标一般不关心（ ）。

- A. 安全性 B. 系统规模 C. 模块化 D. 高效性

1-15 批处理系统的主要缺点是（ ）。

- A. CPU 利用率低 B. 不能并发执行 C. 缺少交互性 D. 以上都不是

1-16 在分时系统中，当用户数为 100 时，为保证响应时间不超过 2 秒，系统设置的时间片长度应为（ ）。

- A. 50ms B. 100ms C. 10ms D. 20ms

1-17 下面关于并发性的定义中，描述正确的是（ ）。

- A. 并发性是指若干事件在同一时刻发生
B. 并发性是指若干事件在不同时刻发生
C. 并发性是指若干事件在同一时间间隔内发生
D. 并发性是指若干事件在不同时间间隔内发生

1-18 批处理系统提高了系统各种资源的利用率和系统吞吐量，因而缩短了作业的周转时间。这句话对吗？请解释。

1-19 在批处理系统中，有两个程序参与运行。其中程序 A 要做的工作依次为：计算 10 分钟，5 分钟 I/O1，计算 5 分钟，10 分钟 I/O2，计算 10 分钟；程序 B 要做的工作依次为：10 分钟 I/O1，计算 10 分钟，5 分钟 I/O2，计算 5 分钟，10 分钟 I/O2。假定单道和多道运行时都是 A 先运行。分别计算两种方式运行时，两个程序的周转时间和 CPU 的利用率。