**第一章 导论**

**1.计算机系统四个组成部分**

**计算机硬件、操作系统、系统程序、用户。**

**2.操作系统概念**

**操作系统是一组控制和管理计算机硬件和软件资源、合理地对各类作业进行调度，以及方便用户的程序集合。**

1. **操作系统的目标**

**执行用户程序，更容易地解决用户问题。**

**使计算机系统使用方便。**

**有效地使用计算机硬件。**

1. **两个视角**

**用户视角、系统视角**

**5.硬中断（interrupt）**

**interrupt是一种在系统内硬件设备产生，又称外部中断（相对于CPU）。**

**6.软中断（trap）**

**Trap是指令产生的中断，又称内部中断（相对于CPU）。**

**7.存储结构**

**主存：CPU可以直接访问的大型存储介质。**

**辅存：非易失性大存储容量的主存储器的扩展。（又称二级存储）**

**磁盘：最常用的辅存。磁盘表面逻辑上划分为磁道，再细分为扇区。磁盘控 制器决定设备和计算机之间的逻辑交互。**

**8.DMA（直接内存访问）**

**用于高速I/O设备，能够以接近内存速度传送信息。**

**设备控制器无需CPU干预直接将数据块从缓冲存储器（设备控制器中的存储）直接传输到主存，每个块只产生一个中断，而不是每个字节的一个中断（块设备）。**

**9.计算机系统**

**单处理器系统、多处理器系统、集群系统**

**10.多处理器系统（并行系统、紧耦合系统）**

**定义：多处理器系统有多个紧密通信的CPU，它们共享计算机总线，有时还有时钟、内存和外设等。**

**优点：增加吞吐量、规模经济、增加可靠性。**

**分类：**

**非对称多处理：每个处理器都有各自特定的任务。一个主处理器控制系统，其他处理器或者向主处理器要任务或做预先定义的任务。**

**对称多处理：每个处理器都要完成操作系统中的所有任务。所有处理器对等，处理器之间没有主-从关系。**

**11.集群系统**

**定义：与多处理器系统一样，集群系统将多个CPU集中起来完成计算任务。然而，集群系统与多处理器系统不同，它是由两个或多个独立的系统耦合起来的。集群计算机共享存储并通过局域网络连接或更快的内部连接。**

**12.操作系统结构**

**操作系统最重要的一点是要有多道程序处理能力。多道程序设计通过组织作业（编码或数据）使CPU总有一个作业在执行，从而提高了CPU的利用率。**

**13.操作系统的三种基本类型**

**Batch systems（批处理系统）**

**Time-sharingsystems（分时系统）**

**Real timesystems（实时系统）**

**14.批处理系统**

**工作方式：**

**用户将作业交给系统操作员，系统操作员将许多用户的作业组成一批作业(jobs)之后输入到计算机中，在系统中形成一个自动转接的连续的作业流，系统自动、依次执行每个作业。最后由操作员将作业结果交给用户。操作系统：自动将控制从一个任务转到下一个任务。**

**分类：单道批处理系统、多道批处理系统**

**批处理操作系统优点：作业流程自动化、效率高、吞吐量高。**

**批处理操纵系统缺点：无交互手段、调试程序困难。**

**15.分时系统- 交互式计算**

**分时系统（或多任务）是多道程序设计的延伸。**

**共享需要一种交互计算机系统，它能提供用户与系统之间的直接通信。响应时间短（通常小于一秒钟）。**

**由于每个动作或命令都较短，每个用户只需少量CPU时间，用户之间切换时间短，所以用户会感觉整个系统为自己所用。**

**16.实时系统**

**定义：实时操作系统是保证在一定时间限制内完成特定功能的操作系统。**

**分类：**

**硬实时系统：硬实时要求在规定的时间内必须完成操作，这是在操作系统设计时保证的（不按时间约束完成意味着失败）。（又称强实时系统）**

**软实时系统：软实时则只要按照任务的优先级，尽可能快地完成操作即可，（不按时完成意味着“不好”）。（又称准实时系统、若实时系统）。**

**17.操作系统的双重模式操作**

**为了区分操作系统代码和用户定义代码的执行，至少需要两种独立的操作模式：用户模式（用户态）、监督程序模式（管理模式、系统模式、特权模式、核心态）。操作系统保护能力实现的必要条件。**

**将能引起损害的机器指令作为特权指令。用户模式下想要执行特权指令，硬件不会执行，会认为是非法指令，并以陷阱的形式通知操作系统。**

**第二章 操作系统结构**

**操作系统服务**

**用户界面（一种是命令行界面；另一种是批界面，最为常用的是图形用户面）、程序执行、I/O操作、文件系统操作、通信、错误检测、资源分配、统计、保护和安全。**

**操作系统的用户界面：**

**命令解释程序（CLI）、图形用户界面（GUI）**

**命令解释程序主要作用：获取并执行用户指定的下一条命令。**

**系统调用（System Call）**

**操作系统内核提供一系列预定功能，通过一组称为系统调用的接口呈现给编程人员，系统调用把应用程序的请求传给内核，系统调用相应的内核函数完成所需的处理，将处理结果返回给应用程序。**

**向操作系统传递参数的三种方法：**

**通过寄存器来传递参数。**

**若参数数量比寄存器多，参数通常存在内存的块和表中，并将块的地址通过寄存器来传递。**

**参数也可以通过程序放在或压入堆栈中，并通过操作系统弹出。**

**系统程序分类：**

**文件管理、状态信息、文件修改、程序语言支持、程序装入和执行、通信。**

**操作系统设计和实现：**

**设计目标需求：用户目标和系统目标**

**机制和策略：机制决定如何做，策略决定做什么**

**操作系统结构**

**简单结构、分层方法、微内核、模块、虚拟机**

**简单结构**

**MS-DOS、原始的UNIX操作系统**

**分层方法**

**定义：操作系统分成若干层（级）。最底层（层0）为硬件，最高层（层N）为用户接口。（理想方法，困难在于如何划分层）**

**微内核**

**微内核方法将所有非基本部分从内核中移走，并将它们实现为系统或用户程序，这样得到了更小的内核。**

**微内核的主要功能是使客户程序和运行在用户空间的各种服务之间进行通信。**

**模块：**

**大多数现代操作系统按模块方式实现内核**

**采用面向对象的方法**

**每个核心组件是分开的**

**每部分与已知接口的其他部分通信**

**可以是动态加载方式，每部分根据需要加载到内核**

**总之，类似于层，但更灵活。**

**虚拟机**

**虚拟机（VirtualMachine）指通过软件模拟的具有完整硬件系统功能的、运行在一个完全隔离环境中的“完整”计算机系统。**

1. **进程**

**进程概念**

**进程是执行中的程序。进程还包括当前活动信息：通过程序计数器的值和处理器寄存器的内容等来表示。另外，进程内存影像包括进程堆栈段（包括临时数据，如函数参数、返回地址和局部变量）和数据段（包括全局变量），包括堆，是在进程运行期间动态分配的内存。**

**进程与程序是截然不同的两个概念**

**一个程序可以对应多个进程(Oneprogram can be several processes)**

**一个进程也可以由多个程序段共同完成一项任务（接力）**

**进程五个特征（而程序不具备）**

**动态性：是进程的最基本的特征，表现在进程由创建而产生，由调度而执行，因得不到资源而暂停执行，由撤销而消亡。**

**并发性：多个进程实体同存于内存中，能在一段时间内同时执行。**

**独立性：进程实体是一个能独立运行的基本单位，同时也是系统中独立获得资源和独立调度的基本单位。**

**异步性：指进程按各自独立的、不可预知的速度向前推进；或者说，进程按异步方式运行。**

**结构特征：从结构上看，进程实体由程序段、数据段以及进程控制块组成。三部分也称为进程影像。**

**进程状态**

**新建（new）：进程正在被创建。**

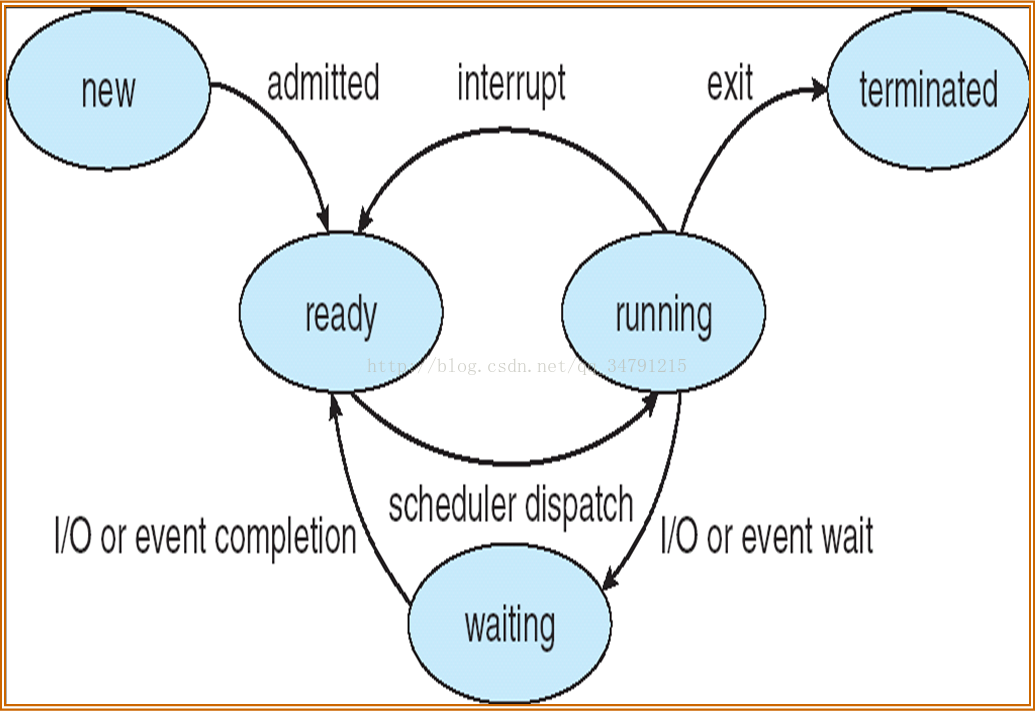
**运行（running）：指令正在被执行。**

**等待（waiting）：进程等待某个时间的发生（如I/O完成或收到信号）。**

**就绪（ready）：进程等待分配处理器。**

**终止（terminated）：进程完成执行。**

**进程状态图“”**

****

**进程控制块**

**每个进程在操作系统内对应一进程控制块（PCB，process control block）。**

**概念：**

**系统为了管理进程设置的一个专门的数据结构，用它来记录进程的外部特征，描述进程的运动变化过程。**

**要点：**

**系统利用PCB来控制和管理进程，所以PCB是系统感知进程存在的唯一标志**

**进程与PCB是一一对应的**

**通常进程队列是进程所对的PCB队列**

**操作系统通过PCB来感知进程的存在**

**PCB表：**

**系统把所有PCB组织在一起，并把它们放在内存的固定区域，就构成了PCB表。PCB表的大小决定了系统中最多可同时存在的进程个数，称为系统的并发度。**

**结构：**

**链接结构：同一状态进程的PCB组成一个链表，不同状态的进程对应多个不同的链表。（就绪链表、阻塞链表……）**

**索引结构：对具有相同状态的进程，分别设置各自的PCB索引表，表明PCB在PCB表中的地址。**

**...**

**进程调度**

**多道程序设计的目的是无论何时都有进程在运行，从而使CPU利用率达到最大化。**

**分时系统的目的是在进程之间快速切换CPU以便用户在程序运行时能与其进行交互。**

**为达到此目的，进程调度选择一个可用的进程到CPU上执行。单处理器系统从不会有超过一个进程在运行。如果有多个进程，那么余下的则需要等待CPU空闲并重新调度。**

**调度程序**

**长期调度程序（或作业调度程序）**

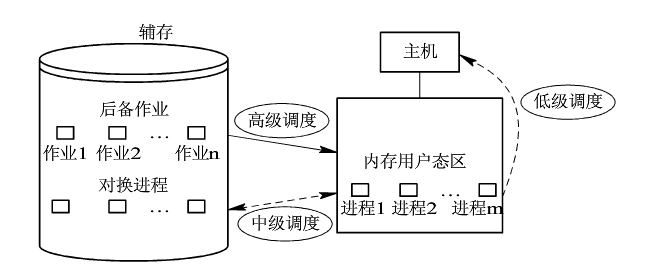
**短期调度层序（或CPU调度程序）**

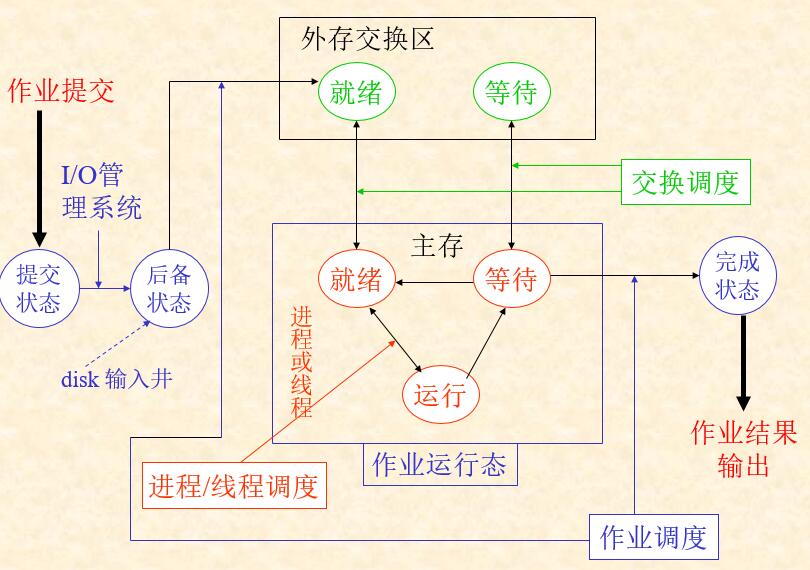
**这两个调度程序的主要差别是它们执行的频率。**

**短期调度程序必须频繁为CPU选择新进程。**

**长期调度程序控制多道程序设计的程度（内存中的进程数量）。**

**中期调度程序（又名交换）核心思想：调度程序根据一定策略临时把内存中的某个进程换出到外存或者把外存中的进程换入内存的过程。（一定程度上降低长期调度的设计或策略难度）。**





**短期调度执行的频率最高。短期调度在内存作业中选择就绪执行的作业，并为他们分配CPU。中期调度作为一种中等程度的调度程序，尤其被用于分时系统，一个交换方案的实施，将部分运行程序移出内存，之后，从中断处继续执行，从而改善进程组合或者因内存要求的改变引起了可用内存的过度使用而需要释放内存。长期调度确定哪些作业调入内存以执行. 它们主要的不同之处是它们的执行的频率。短期调度必须经常调用一个新进程，由于在系统中，长期调度处理移动的作业时，并不频繁被调用，可能在进程离开系统时才被唤起。**

**上下文切换**

**将CPU切换到另一个进程需要保存当前进程的状态并恢复另一个进程的状态，这一任务称为上下文切换。**

**进程操作**

**进程标识符（pid）**

**通过fork()系统调用，可创建新进程。**

**新进程通过复制原来的地址空间形成。**

**子进程返回0，父进程返回大于0，错误返回-1.**

**Fork()的功能**

**内核为子进程做一个父进程的上下文的拷贝（复制父进程的PCB作为子进程的PCB），子进程与父进程共享子进程创建之前父进程所有的资源，父进程和子进程在不同的地址空间上运行。**

**Fork()的几个要点**

**父子进程具有独立的内存空间。**

**父子进程资源的共享与分离：父进程中在fork之前创建的变量——先继承，后分离，子进程继承了父进程的私有变量，作为自己的私有变量；Fork之后各自创建的变量——完全分离。**

**子进程继承了父进程的所有资源，其中包括父进程的这些私有变量，但继承以后互相不能访问。**

**进程间通信有两种基本模式**

**共享内存：允许以最快的速度进行方便的通信，在计算机中可以达到内存的速度，等等。**

**消息传递：对于交换较少数量的数据很有用，因为不需要避免冲突，对于计算机间的通信，消息传递也比共享内存更易于实现，**