第一章

- 计算机模型：CPU(流水线/超标量，内核态/用户态、多线程)、存储、IO设备(设备控制器/IO设备)
- 寄存器/高速缓存/内存(RAM/ROM/EEPROM/flash memory/CMOS)/磁盘(磁道/柱面/扇区)
- PCI总线/ISA总线
- **基本特征**包括并发、共享、虚拟和异步
    - 并发是指两个或多个事件在同一时间间隔内发生。操作系统的并发性是指计算机系统中同时存在多个运行着的程序，因此它具有处理和调度多个程序同时执行的能力。
    - 互斥共享方式/同时访问方式
    - 虚拟是指把一个物理上的实体变为若干个逻辑上的对应物。在操作系统中利用了多种虚拟技术，分别用来实现虚拟处理器、虚拟内存和虚拟外部设备等。
    - 在多道程序环境下，允许多个程序并发执行，但由于资源有限，进程的执行不是一贯到底，而是走走停停，以不可预知的速度向前推进，这就是进程的异步性。
- 操作系统的**发展**（多道批处理、分时、实时 ）
    - 多道程序设计技术允许多个程序同时进入内存并运行。即同时把多个程序放入内存，并允许它们交替在CPU中运行，它们共享系统中的各种硬、软件资源。当一道程序因I/O请求而暂停运行时，CPU便立即转去运行另一道程序。
    - 在操作系统中采用分时技术就形成了分时系统。所谓分时技术就是把处理器的运行时间分成很短的时间片，按时间片轮流把处理器分配给各联机作业使用。若某个作业在分配给它的时间片内不能完成其计算，则该作业暂时停止运行，把处理器让给其他作业使用，等待下一轮再继续运行。
    - 为了能在某个时间限制内完成某些紧急任务而不需时间片排队，诞生了实时操作系统。这里的时间限制可以分为两种情况：如果某个动作必须绝对地在规定的时刻（或规定的时间范围）发生，则称为硬实时系统。例如，飞行器的飞行自动控制系统，这类系统必须提供绝对保证，让某个特定的动作在规定的时间内完成。如果能够接受偶尔违反时间规定，并且不会引起任何永久性的损害，则称为软实时系统，如飞机订票系统、银行管理系统。
- OS的**主要功能**----资源管理器和用户接口
    - 资源管理功能：处理机管理、存储器管理、设备管理、文件管理
    - 操作系统和用户之间的接口：用户接口：联机用户接口，脱机用户接口和图形用户接口；程序接口：该接口是为用户程序在执行中访问系统资源而设置的，它是由一组系统调用组成。

第二章

- **进程的概念**
    - 计算机中正在运行的程序实例；可以分配给处理器并由处理器执行的一个实体。
    - 由程序段、相关的数据段和PCB三个部分便构成了进程实体。进程实质是进程实体的一次执行过程
    - PCB中主要包括下述四方面的信息：进程标识符（内部标识符，外部标识符）、处理机状态进程、调度信息、进程控制信息
- **进程和程序区别：**

  （1）进程是一个**动态概念**，强调执行的过程，每个进程中包含了程序段和数据段两个部分，以及进程控制块PCB；而程序是一个**静态概念**，程序是指令的有序集合，无执行含义；

  （2）进程具有**并行特征**（独立性，异步性），程序则没有；

  （3）**一个进程可以执行多个程序（如Linux中通过exec调用），同一程序的多次执行将产生多个不同的进程。同一个程序的一次执行也可产生多个进程（如在程序中多次调用Linux中的fork）。**
- **进程的三种基本状态：就绪状态，执行状态，阻塞状态**
- **进程间通信方式：**
    - 管道（pipe）及有名管道（named pipe）： 管道可用于具有亲缘关系的父子进程间的通信，有名管道除了具有管道所具有的功能外，它还允许无亲缘关系进程间的通信。
    - 信号（signal）： 信号是在软件层次上对中断机制的一种模拟，它是比较复杂的通信方式，用于通知进程有某事件发生，一个进程收到一个信号与处理器收到一个中断请求效果上可以说是一致的。
    - 消息队列（message queue）： 消息队列是消息的链接表，它克服了上两种通信方式中信号量有限的缺点，具有写权限得进程可以按照一定得规则向消息队列中添加新信息；对消息队列有读权限得进程则可以从消息队列中读取信息。
    - 共享存储（shared memory）： 可以说这是最有用的进程间通信方式。它使得多个进程可以访问同一块内存空间，不同进程可以及时看到对方进程中对共享内存中数据得更新。这种方式需要依靠某种同步操作，如互斥锁和信号量等。
    - 信号量（semaphore）： 主要作为进程之间及同一种进程的不同线程之间得同步和互斥手段。
    - 套接字（socket）；这是一种更为一般得进程间通信机制，它可用于网络中不同机器之间的进程间通信，应用非常广泛。
- **线程之间的同步通信：**
    - 信号量 二进制信号量 互斥信号量 整数型信号量 记录型信号量
- **进程的特征：**
    - 动态性：进程是程序的一次执行，它有着创建、活动、暂停、终止等过程，具有一定的生命周期，是动态地产生、变化和消亡的。动态性是进程最基本的特征。
    - 并发性：指多个进程实体，同存于内存中，能在一段时间内同时运行，并发性是进程的重要特征，同时也是操作系统的重要特征。引入进程的目的就是为了使程序能与其他进程的程序并发执行，以提高资源利用率。
    - 独立性：指进程实体是一个能独立运行、独立获得资源和独立接受调度的基本单位。凡未建立PCB的程序都不能作为一个独立的单位参与运行。
    - 异步性：由于进程的相互制约，使进程具有执行的间断性，即进程按各自独立的、 不可预知的速度向前推进。异步性会导致执行结果的不可再现性，为此，在操作系统中必须配置相应的进程同步机制。
    - 结构性：每个进程都配置一个PCB对其进行描述。从结构上看，进程实体是由程序段、数据段和进程控制段三部分组成的。
- **线程的基本概念：** 线程，有时称为轻量级进程，是CPU使用的基本单元；它由线程ID、程序计数器、寄存器集合和堆栈组成。它与属于同一进程的其他线程共享其代码段、数据段和其他操作系统资源（如打开文件和信号）。
- 线程的作用：
    - 比进程更加轻量
    - 共享同一个地址空间和所有可用数据的能力
    - 加快程序速度
    - 并行可能
- **线程与进程的联系区别**
    - 都有**并发性**
    - 线程是进程的一部分，所以线程有的时候被称为是轻权进程或者**轻量级**进程。
    - 系统在运行的时候会为每个进程分配不同的内存区域，但是不会为线程分配内存（线程所使用的资源是它所属的进程的资源），线程组只能共享资源。那就是说，出了CPU之外（线程在运行的时候要占用CPU资源），计算机内部的软硬件资源的分配与线程无关，线程只能**共享**它所属进程的资源。
    - 进程是系统所有**资源分配**时候的一个基本单位，线程是**独立调度**的基本单位，拥有一个完整的虚拟空间地址，**独立存在**，而线程必须依赖程序无法独立存在
- 线程基本状态：就绪、运行、阻塞、终止
- 线程的实现可以分为两类：**用户级线程**(User-LevelThread, ULT)和**内核级线程**(Kemel-LevelThread，KLT)。
- 用户级线程
    - 优点：
        - 可以在不支持线程的操作系统中实现。
        - 线程的调度不需要内核直接参与，控制简单。
        - 允许每个进程定制自己的调度算法，线程管理比较灵活。这就是必须自己写管理程序，与内核线程的区别
        - 创建和销毁线程、线程切换代价等线程管理的代价比内核线程少得多。
        - 线程能够利用的表空间和堆栈空间比内核级线程多。
    - 缺点
        - 同一进程中只能同时有一个线程在运行，如果有一个线程使用了系统调用而阻塞，那么整个进程都会被挂起。另外，页面失效也会产生同样的问题。
        - 资源调度按照进程进行，多个处理机下，同一个进程中的线程只能在同一个处理机下分时复用
- 内核级线程
    - 优点：
        - 当有多个处理机时，一个进程的多个线程可以同时执行。
    - 缺点
        - 由内核进行调度。
- 竞争条件/临界区
- **互斥**：亦称间接制约关系。当一个进程进入临界区使用临界资源时，另一个进程必须等待, 当占用临界资源的进程退出临界区后，另一进程才允许去访问此临界资源。
- **同步**：亦称直接制约关系，它是指为完成某种任务而建立的两个或多个进程，这些进程因为需要在某些位置上协调它们的工作次序而等待、传递信息所产生的制约关系。进程间的直接制约关系就是源于它们之间的相互合作。
- 忙等待的互斥方法：
    - 锁变量(产生竞争条件)
    - 自旋锁(等待时间长)
    - 双标志法先检查(产生竞争条件)
    - 双标志法后检查(饥饿状态)
    - Peterson解法
    - 屏蔽中断(仅针对单CPU)j
    - TSL指令
- **生产者消费者问题**
    - Sleep-Wakeup模式
    - **信号量/互斥量**
    - 信号量是Dijkstra提出的用于解决进程同步的有效工具。信号量是一个数据结构以及对其的操作。除初始化外，仅能通过两个标准的原子操作down(S)和up(S)来访问。两个语句在执行到一半的时候不能被中断。down对信号量减一，若信号量为0，则进程将睡眠。up对信号量加一，并且唤醒一个进程
    - 互斥量mutex，用于任何时刻只有一个进程读写相应内存
    - **条件变量：** 条件变量允许线程由于一些未达到的条件而阻塞
    - **管程：** 当一个进程调用管程过程时，该过程中前几个指令将检查管程中是否有其他活跃进程，以保障管程中只有1个进程
    - 屏障
- **调度**：在多道程序系统中，进程的数量往往多于处理机的个数，进程争用处理机的情况就在所难免。处理机调度是对处理机进行分配，就是从就绪队列中，按照一定的算法（公平、高效）选择一个进程并将处理机分配给它运行，以实现进程并发地执行
- 调度条件：创建进程/进程退出/进程阻塞/I/O中断
- 进程调度方式：非剥夺调度方式，又称非抢占方式/剥夺调度方式，又称抢占方式。
- **调度的基本准则**：
    - CPU利用率
    - 系统吞吐量
    - 周转时间
    - 等待时间
    - 响应时间
- **操作系统调度算法**
    - 先来先服务(FCFS)调度算法
    - 短作业优先(SJF)调度算法
    - 高响应比优先调度算法
    - 最短剩余时间优先
    - 轮转调度
    - 优先级调度
    - 多级反馈队列调度算法

第三章

- **地址空间**：是一个进程可用于寻址内存的一套地址集合
- 基址寄存器/界限寄存器
- 内存覆盖：由于程序运行时并非任何时候都要访问程序及数据的各个部分（尤其是大程序），因此可以把用户空间分成一个固定区和若干个覆盖区。将经常活跃的部分放在固定区，其余部分按调用关系分段。首先将那些即将要访问的段放入覆盖区，其他段放在外存中，在需要调用前，系统再将其调入覆盖区，替换覆盖区中原有的段。
- **内存交换**：把处于等待状态（或在CPU调度原则下被剥夺运行权利） 的程序从内存移到辅存，把内存空间腾出来，这一过程又叫换出；把准备好竞争CPU运行的程序从辅存移到内存，这一过程又称为换入。第2章介绍的中级调度就是采用交换技术。
- 空闲内存管理：
    - 首次适应(First Fit)算法：空闲分区以地址递增的次序链接。分配内存时顺序查找，找到大小能满足要求的第一个空闲分区。
    - 最佳适应(Best Fit)算法：空闲分区按容量递增形成分区链，找到第一个能满足要求的空闲分区。
    - 最坏适应(Worst Fit)算法：又称最大适应(Largest Fit)算法，空闲分区以容量递减的次序链接。找到第一个能满足要求的空闲分区，也就是挑选出最大的分区。
    - 邻近适应(Next Fit)算法：又称循环首次适应算法，由首次适应算法演变而成。不同之处是分配内存时从上次查找结束的位置开始继续查找
- **虚拟内存**：每个程序拥有自己地址空间，空间被分割成多个块，每一块称作一页。每一页有连续地址范围，被映射到物理内存，但不是所有页都必须在内存中才能运行程序。当被引用的一部分在内存中时，进行映射。当引用部分不在内存中时，由系统装入物理内存并重新执行指令
- 页面/页框/页表/快表TLB
- 在/不在位-保护位-访问位-修改位-高速缓存禁止位
- **多级页表**/倒排页表
- **页面置换算法**
    - 最优算法OPT
    - NRU最近未使用(未访问未修改-未访问已修改-已访问未修改-已访问已修改)
    - FIFO先进先出(不使用栈，被访问元素不会放在栈顶)
    - 第二次机会
    - LRU最近最少使用算法(使用栈，被访问元素被放在栈顶)
    - NFU最久未使用(计数器-淘汰计数器小值)
    - 老化算法(计数器-R位移位)
    - 时钟算法
    - 工作集算法(未访问超时-未访问未超时-已访问未超时)
    - 工作集时钟算法(未修改未访问超时-未修改未访问未超时-已修改未访问超时-已修改未访问未超时-未修改已访问未超时-已修改已访问未超时)
- 局部分配和全局分配策略/负载控制
- 请求调页/预先调页/页面大小/清除策略/内存映射文件
- 分段

第四章

- 普通文件/目录/ASCII文件/二进制文件
- **主引导记录MBR**/引导块/超级块/**空闲空间管理**/**i节点**/**根目录**/文件和目录
- 连续分配/链表分配/内存链表分配/**i节点**
- 文件属性在目录项/**文件属性在i节点**/文件名存储
- 共享文件/**硬链接 /软连接**
- 日志文件系统
- **虚拟文件系统VFS/POSIX接口/VFS接口**
- 空闲空间管理/**位图/链表**
- 块大小/磁盘配额/文件系统备份/选择备份/增量转储/压缩/物理转储/逻辑转储
- 文件系统一致性/块一致性/文件一致性
- 块高速缓存/块提前读
- **查找文件/user/ast/mbox**
    - **文件系统定位根目录，在UNIX系统中，根目录i节点存放在磁盘固定位置，i节点含有根目录的块地址。从这个i节点，系统可以定位根目录的块地址。**
    - **接下来，系统在根目录的块地址中查找路径的第一个分量user，获取user目录的i节点号。由i节点号定位i节点的位置，这一步很容易，因为每个i节点在磁盘上都有固定的位置，找到i节点后，读取i节点的块地址，找到ast分量。**
    - **找到ast分量后，获取i节点名，找到i节点地址，找到块地址。**
    - **找到mbox 分量**，**找到mbox文件i节点名，找到文件i节点地址，里面有mbox文件各个块的地址**

第五章

- 块设备/字符设备
- 设备控制器(适配器)/驱动器/控制寄存器/数据缓存区：把串行的位流转为字节块
- I/O端口/**内存映射I/O**：有些体系结构的CPU（如，PowerPC、m68k等）通常只实现一个物理地址空间（RAM）。在这种情况下，外设I/O端口的物理地址就被映射到CPU的单一物理地址空间中，而成为内存的一部分。此时，CPU可以象访问一个内存单元那样访问外设I/O端口，而不需要设立专门的外设I/O指令。这就是所谓的“内存映射方式”
- **主机与外设数据交换方式：程序查询、中断、DMA、通道**
- **直接存储器存取DMA**：以磁盘为例
    - 没有DMA：控制器从磁盘驱动器串行地、一位一位读块，直到整个块信息放入控制器内部缓存器。接着计算校验和，然后控制器产生中断，由操作系统从控制器的缓存器中一个字节一个字节读取块信息，并放入内存中
    - CPU通过设置DMA控制器寄存器程序，DMA控制器知道将什么数据传送到哪个地方。DMA控制器还要向磁盘控制器发出命令，从控制器缓存中读取块信息。
    - 周期窃取/突发窃取/飞跃窃取
- **中断控制器**
- 中断处理程序/设备驱动程序/设备无关I/O操作系统软件/用户级IO软件
- **RAID磁盘阵列**
    - 0级：条带
    - 1级：条带+冗余备份
    - 2级：字位+位奇偶校验码
    - 3级：字位+字奇偶校验码
    - 4级：条带+奇偶校验码
    - 5级：条带+均匀分布奇偶校验码
- **磁盘臂调度算法**
    - 先来先服务FCFS
    - 最短寻道优先
    - 电梯算法

第六章

- **死锁**：如果一个进程集合中每个进程都在等待只能由该进程集合中的其他进程才能引发的事件，那么该进程集合就是死锁的
- 资源死锁及条件：互斥、不可抢占、占有和等待条件、环路等待
- 处理死锁策略：鸵鸟算法、死锁检测（EACR）和恢复(抢占、回滚、杀死)、死锁避免（银行家算法）、死锁预防

CPU与外设之间的数据传送方式主要有程序传送方式、中断传送方式和直接存储器存取DMA传送方式