1. 进程通信
2. 概述

进程通信就是各个进程之间交换信息的过程，P,V操作和信号量的使用只能传递信号，进程之间还需要传递大批量的数据，由于数据量远比信号量大，因此需要引入高级通信机制。

1. 进程通信类型
2. 低级和高级通信

根据交换信息量的多少和效率的高低，分为低级方式和高级方式。P,V操作和管程都属于低级方式，管道通信和信箱通信都属于高级方式。低级通信只能传递状态和整数值，如信号量和管程。高级通信能够传送大量数据，如共享内存，消息传递，共享文件（管道）。

1. 直接通信和间接通信

直接通信，信息由发送方直接传递给接收方，如管道。间接通信，将收发双方进程之外的共享数据结构作为通信中转，实现通信，如消息队列，发送方和接收方的数目是任意的。

1. 共享内存

UNIX系统中使用，系统在内存中指定一个区域作为共享存储区，建立一张段表来管理，各个进程可以申请其中的一个存储段，在申请时提供关键字。若申请的存储区已经被其他进程占用了，系统会向申请进程返回关键字，该存储区就连接到了进程的逻辑地址空间，此后进程就可以直接存取共享存储区中的数据了。若申请的存储段尚未分配，则系统会分配存储段，并在段表中加入该进程的信息。一个进程可以申请多个存储段，

1. 消息传递
2. 原理

每个消息都由消息头和消息体组成，系统中有一定数量的消息缓冲区，它是进程通讯的一个基本单位，每当发送进程要发送消息时，使用Send原语将消息从发送区复制到消息缓冲区，挂在接受进程的消息队列末尾，如果该接收进程因等待消息而处于阻塞状态，则将其唤醒，当接受进程要读取消息时，使用接受原语Receive()从消息队列头中取走一个消息放到自己的接受区中。

消息队列是临界资源，因此在PCB中设置了一个互斥的信号量mutex，每当有进程要进入消息队列时，执行P操作，退出消息队列时，执行V操作。

1. 消息传递方式
2. 直接通信

利用send原语和receive原语实现通信。

1. 间接通信

利用信箱作为中介进行消息传递，信箱缓存一定数量的消息，每个信箱用标识符加以区分，由信箱头和信箱体组成，头部存放控制信息，一个信箱可以被多个进程共享，这就实现了消息的广播发送。

1. 管道

管道通信是一种共享文件模式，基于文件系统，连接于两个通信进程之间，以FIFO的方式实现消息的单向传送。管道是一个特殊文件，在内核中通过文件描述符表示。在UNIX系统中，管道的创建使用pipe()实现的，管道创建完毕后，返回两个分别用于读，写操作的文件描述符fd[0], fd[1]，读管道时调用read()，利用fd[0]从管道中读取字节。如果进程要实现双向通信，则需要定义两个管道，只适用于父子进程的通信。

另一种命名管道，有自己的名字和访问权限，可以用于不相关进程之间的通信，进程通过管道的名字获取管道。

1. 线程
2. 概述

线程也叫轻量级进程，是可以执行的实体单元，CPU调度的基本单位。

1. 多线程

多线程指的是操作系统支持在单个进程中执行多个线程的能力，多线程环境中，进程是保护单位和资源分配单位，进程中的每个线程共享所属进程内的主存和其他资源。进程中包含PCB，进程的用户地址空间，以及每个线程独有的线程控制块和用户栈区（核心栈，用户栈），在控制块中包含有寄存器映像，优先级等，保证线程运行环境的独立性。

1. 线程的优势

创建和销毁线程的开销非常小；

线程间的切换速度非常快；

通信效率高，同一进程中的线程共享统一地址空间，通信时不需要借助内核功能；

并发度高，对进程个数有限制，但是对于线程个数是没有限制的；

1. 线程的状态
2. 创建

进程创建的过程中也创建了线程，该线程可以继续创建其他线程。

1. 阻塞

线程等待某一事件的发生。

1. 解除阻塞

等待的事件已发生，线程移入就绪队列。

1. 终止

线程完成任务，释放占用点的寄存器和栈空间。

1. 线程的实现机制

实现机制由用户级线程和核心级线程。

1. 用户级线程（ULT）

指的是由每个进程通过进程中的线程表来管理线程，核心感觉不到线程的存在，CPU在两个线程切换时不用进入到核心态执行，节省了用户态与核心态之间切换的开销。用户级线程的管理机制可以运行在各种操作系统中，灵活，但是当线程执行系统调用时，整个进程都被阻塞，不能充分利用CPU。

1. 核心级线程（KLT）

内核线程，运行在核心中，通过核心来管理，核心中包括进程表和线程表，核心可以调度一个进程中的多个线程同时运行，当某线程发生阻塞时，可以调度其他线程执行。优先是发送了多CPU的并行能力，缺点是核心与用户切换代价大。

1. 线程与进程的比较
2. 线程使用的调度程序可以是用户程序，也可以是系统程序。线程的调度必须在运行的进程中才能进行。
3. 线程与进程一样具有并发性，可在多个CPU之间并行执行。
4. 同一进程中的线程共享进程的资源和状态，存在于进程的同一地址空间中。线程具有寄存器和栈。