

# 进程和线程之间有什么区别？

进程：并发执行的程序在执行过程中管理和分配资源的基本单位，是动态的概念。

线程：是进程更小的执行单元，独立运行的基本单位。

地址空间：

线程共享本进程的地址空间，而进程之间是独立的地址空间。

资源：

线程共享本进程的资源如内存、I/O、cpu等，不利于资源的管理和保护，而进程之间的资源是独立的，能很好的进行资源管理和保护。

健壮性：

多进程要比多线程健壮，一个进程崩溃后，在保护模式下不会对其他进程产生影响，但是一个线程崩溃整个进程都死掉。

执行过程：

每个独立的进程有一个程序运行的入口、顺序执行序列和程序入口，执行开销大。

但是线程不能独立执行，必须依存在应用程序中，由应用程序提供多个线程执行控制，执行开销小。

可并发性：

两者均可并发执行。

切换时：

进程切换时，消耗的资源大，效率高。所以涉及到频繁的切换时，使用线程要好于进程。同样如果要求同时进行并且又要共享某些变量的并发操作，只能用线程不能用进程。

其他：

线程是处理器调度的基本单位，但是进程不是。

## 进程间有哪些通信方式？

1.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。

2.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。

3.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。

#### 4.共享内存（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。

#### 5.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。

#### 6.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。

## 简述 socket 中 select 与 epoll 的使用场景以及区别， epoll 中水平触发以及边缘触发有什么不同？

select==>时间复杂度 $O(n)$

它仅仅知道了，有I/O事件发生了，却并不知道是哪那几个流（可能有一个，多个，甚至全部），我们只能无差别轮询所有流，找出能读出数据，或者写入数据的流，对他们进行操作。所以select具有 $O(n)$ 的无差别轮询复杂度，同时处理的流越多，无差别轮询时间就越长。

epoll可以理解为event poll，不同于忙轮询和无差别轮询，epoll会把哪个流发生了怎样的I/O事件通知我们。所以我们说epoll实际上是**事件驱动（每个事件关联上fd）**的，此时我们对这些流的操作都是有意义的。（复杂度降低到了 $O(1)$ ）

1、表面上看epoll的性能最好，但是在连接数少并且连接都十分活跃的情况下，select和poll的性能可能比epoll好，毕竟epoll的通知机制需要很多函数回调。

2、select低效是因为每次它都需要轮询。但低效也是相对的，视情况而定，也可通过良好的设计改善

LT(level triggered)是缺省的工作方式，并且同时支持block和no-block socket.在这种做法中，内核告诉你一个文件描述符是否就绪了，然后你可以对这个就绪的fd进行IO操作。如果你不作任何操作，内核还是会继续通知你的。

ET(edge-triggered)是高速工作方式，只支持no-block socket。在这种模式下，当描述符从未就绪变为就绪时，内核通过epoll告诉你。然后它会假设你知道文件描述符已经就绪，并且不会再为那个文件描述符发送更多的就绪通知，直到你做了某些操作导致那个文件描述符不再为就绪状态了

## Linux 进程调度中有哪些常见算法以及策略？

先来先去服务 短作业(进程)优先调度算法SJ(P)F 轮转法 多级反馈队列算法

## 操作系统如何申请以及管理内存的？

内存管理程序通过映射机制把用户程序的逻辑地址映射到物理地址。当用户程序运行时，如果发现程序中要用的虚地址没有对应的物理内存，就发出了请求页要求。如果有空闲的内存可供分配，就请求分配内存(于是用到了内存的分配和回收)，并把正在使用的物理页记录在缓存中(使用了缓存机制)。如果没有足够的内存可供分配，那么就调用交换机制；腾出一部分内存。另外，在地址映射中要通过TLB(翻译后援存储器)来寻找物理页；交换机制中也要用到交换缓存，并且把物理页内容交换到交换文件中，也要修改页表来映射文件地址。进程和线程、进程间及线程通信方式、共享内存的使用实现原理。