

操作系统

配置在计算机硬件上的第一层软件，负责管理硬件设备，并为用户和应用程序提供一个简单的接口，以使用户和应用程序使用硬件设备

进程和线程

进程

进程控制块(PCB)

进程可以提高资源利用率和吞吐量

多线程 OS 中进程是一个可拥有资源的基本单位，且不再是一个可执行实体。不过更改进程的状态，该进程的所有线程状态也会改变

线程

线程控制块(TCP)

线程是比进程更小的基本单位，线程使 OS 具有更好的并发性，可以减少程序在并发时付出的时空间。属于同一进程的所有线程都具有相同地址空间，线程除了自己的少量资源，还共享他所属的进程所拥有的资源。

传统 OS 进程作为调度和分派的基本单位，能够独立运行。但每次调度都需要切换上下文，开销较大。

引入线程的 OS 线程作为调度和分派的基本单位，能够独立运行。线程切换代价远低于进程，同一进程中的线程切换不会引起进程切换，从一个进程中的线程切换到另一个进程中的线程会引起进程切换。

并发性

进程间能并发执行，同一进程中的线程能并发执行，不同进程中的线程也能并发执行。

进程的三种基本状态

- 就绪(ready)状态,进程分配到除 CPU 外的必要资源,只要再获得 CPU 便可立即执行
- 执行(running)状态,进程获得 CPU 后其程序“正在执行”这一状态
- 阻塞(block)状态,进程由于发生某事件(如 IO 请求、申请缓冲区等)而无法继续执行
- 创建状态,进程所必需的资源尚不能得到满足,创建工作未完成此时为创建状态
- 终止状态,进程工作完成,或是出现了无法克服的错误或是被 OS、父进程所终止时就会进入终止状态

并行与并发

并行是指同一时刻有多个线程在多个处理机上同时执行。

并发是指同一时刻只有一个线程能占用处理机,但是多个线程快速交替使用处理机,宏观上来看就是多个线程同时执行的效果,微观上来说并不是同时执行。

僵尸进程和孤儿进程是什么?

- 孤儿进程:父进程退出了,而它的一个或多个子进程还在运行,那这些子进程都会成为孤儿进程。孤儿进程将被 init 进程(进程号为 1)所收养,并由 init 进程对它们完成状态收集工作。
- 僵尸进程:子进程比父进程先结束,而父进程又没有释放子进程占用的资源,那么子进程的进程描述符仍然保存在系统中,这种进程称之为僵尸进程。

进程通信

- 管道
 - 匿名管道:半双工的通信方式,数据只能单向流动,而且只能在具有亲缘关系的进程间使用。进程的亲缘关系通常是指父子进程关系。匿名管道
 - 有名管道:半双工的通信方式,但是它允许无亲缘关系进程间的通信。
- 信号:通知接收进程某个事件已经发生。
- 消息队列:以消息链表的形式出现,消息队列克服了信号传递信息少、管道只能承载无格式字节流以及缓冲区大小受限等缺点。
- 共享内存:映射一段能被其他进程所访问的内存,这段共享内存由一个进程创建,但多个进程都可以访问。最快的进程间通信方式,常配合其他通信方式使用。

- 信号量：信号量是一个计数器，可以用来控制多个进程对共享资源的访问。
- 套接字：适用于异地计算机通信以及本地同一台计算机内部通信

进程调度算法

- **FCFS 先来先服务**：从就绪队列中选择一个 最先进入该队列的进程 为之分配资源，使它立即执行并一直执行到完成或发生某事件而被阻塞放弃占用 CPU 时再重新调度。
- **SJB 短作业优先**：从就绪队列中选出一个 估计运行时间最短的进程 为之分配资源，使它立即执行并一直执行到完成或发生某事件而被阻塞放弃占用 CPU 时再重新调度。
- **RR 时间轮转片**：每个进程被分配一个时间片，表示该进程允许运行的时间。
- **P 优先级调度**：为每个进程设置优先级，依次出队最高优先级的进程 为之分配资源，优先级相同的进程采用 FCFS 调度。
- 多级反馈队列：

什么是资源分配，分配的资源指的是什么？

每个进程创建时都需要为其分配资源，这些资源包括处理机、内存、外部设备使用权。

死锁？死锁产生的原因？ 如果解决死锁的问题？

死锁，是指多个进程在运行过程中因争夺资源而造成的一种僵局，当进程处于这种僵持状态时，若无外力作用，它们都将无法再向前推进。

产生死锁的原因：

- 竞争资源
 - 产生死锁中的竞争资源之一指的是竞争不可剥夺资源（例如：系统中只有一台打印机，可供进程 P1 使用，假定 P1 已占用了打印机，若 P2 继续要求打印机打印将阻塞）
 - 产生死锁中的竞争资源另外一种资源指的是竞争临时资源（临时资源包括硬件中断、信号、消息、缓冲区内的消息等），通常消息通信顺序进行不当，则会产生死锁

- 进程间推进顺序非法

若 P1 保持了资源 R1，P2 保持了资源 R2，系统处于不安全状态，因为这两个进程再向前推进，便可能发生死锁。例如，当 P1 运行到 P1: Request (R2) 时，将因 R2 已被 P2 占用而阻塞；当 P2 运行到 P2: Request (R1) 时，也将因 R1 已被 P1 占用而阻塞，于是发生进程死锁

产生死锁的必要条件：

- 互斥条件：进程要求对所分配的资源进行排它性控制，即在一段时间内某资源仅为一进程所占用。
- 请求和保持条件：当进程因请求资源而阻塞时，对已获得的资源保持不放。
- 不剥夺条件：进程已获得的资源在未使用完之前，不能剥夺，只能在使用完时由自己释放。
- 环路等待条件：在发生死锁时，必然存在一个进程——资源的环形链。

预防死锁的方法：(银行家算法)

- 资源一次性分配：一次性分配所有资源，这样就不会再有请求了（破坏请求条件）
- 只要有一个资源得不到分配，也不给这个进程分配其他的资源（破坏请保持条件）
- 可剥夺资源：即当某进程获得了部分资源，但得不到其它资源，则释放已占有的资源（破坏不可剥夺条件）
- 资源有序分配法：系统给每类资源赋予一个编号，每一个进程按编号递增的顺序请求资源，释放则相反（破坏环路等待条件）

虚拟内存

我们在开启很多占内存的软件后，这些软件占用的内存可能远远超出了电脑本身具有的物理内存，通过虚拟内存可以让程序拥有超过系统物理内存大小的可用内存空间。虚拟内存为每个进程提供了一个连续完整的私有内存空间。

虚拟内存是一种使用硬盘空间来扩展内存的技术。虚拟内存定义了一个连续的虚拟地址空间，并且把内存扩展到了硬盘空间