并发处理

进程（Process）

是计算机中程序关于某数据集合上的一次运行活动，是系统进行资源分配和调度的基本单元，是操作系统结构的基础。进程是一个执行中的程序

进程的三态模型：运行、就绪、阻塞

进程的五态模型：新建态、活跃就绪/静止就绪、运行、活跃阻塞/静止阻塞、终止态

1. 新建态:对应于进程刚刚被创建时没有被提交的状态,并等待系统完成创建进程的所有必要信息
2. 终止态:进程已结束运行,回收除进程控制块之外的其他资源,并让其他进程从进程控制块中收集有关信息。
3. 活跃就绪:是指进程在主存并且可被调度的状态。
4. 静止就绪(挂起就绪):是指进程被对换到辅存时的就绪状态,是不能被直接调度的状态,只有当主存中没有活跃就绪态进程,或者是挂起就绪态进程具有更高的优先级,系统将把挂起就绪态进程调回主存并转换为活跃就绪。
5. 活跃阻塞:是指进程已在主存,一旦等待的事件产生便进入活跃就绪状态。
6. 静止阻塞:进程对换到辅存时的阻塞状态,一旦等待的事件产生便进入静止就绪状态。

线程

线程是进程中的一个实体,是被系统独立调度和分派的基本单位,线程自己不拥有系统资源,只拥有一点儿在运行中必不可少的资源但它可与同属一个进程的其它线程共享进程所拥有的全部资源。

一个线程可以创建和撤消另一个线程,同一进程中的多个线程之间可以并发执行。

线程是程序中一个单一的顺序控制流程。进程内一个相对独立的、可调度的执行单元,是系统独立调度和分派 CPU 的基本单位指运行中的程序的调度单位。

在单个程序中同时运行多个线程完成不同的工作,称为多线程。

协程

协程是一种用户态的轻量级线程,协程的调度完全由用户控制。协程拥有自己的寄存器上下文和栈。协程调度切换时,将寄存器上下文和栈保存到其他地方,在切回来的时候,恢复先前保存的寄存器上下文和栈,直接操作栈则基本没有内核切换的开销,可以不加锁的访问全局变量,所以上下文的切换非常快。

区别

线程与进程

1. 线程是进程内的一个执行单元,进程内至少有一个线程,它们共享进程的地址空间,而进程有自己独立的地址空间
2. 进程是资源分配和拥有的单位,同一个进程内的线程共享进程的资源
3. 线程是处理器调度的基本单位但进程不是
4. 二者均可并发执行
5. 每个独立的线程有一个程序运行的入口、顺序执行序列和程序的出口,但是线程不能够独立执行,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

协程和线程

1. 一个线程可以多个协程,一个进程也可以单独拥有多个协程
2. 线程进程都是同步机制,而协程则是异步
3. 协程能保留上一次调用时的状态,每次过程重入时,就相当于进入上一次调用的状态

例子

1. 单进程单线程:一个人在一个桌子上吃菜
2. 单进程多线程:多个人在同一个桌子上一起吃菜
3. 多进程单线程:多个人每个人在自己的桌子上吃菜

同步阻塞

多进程模式

1. 创建一个 socket
2. 进入 while 循环,阻塞在进程 accept 操作上,等待客户端连接进入主进程在多进程模型下通过 fork 刨建子进程
3. 收到数据后服务器程序进行处理然后使用 send 向客户端发送响应

4. 当客户端连接关闭时,子进程/线程退出并销毁所有资源。主进程/线程会回收掉此子进程/线程。

多线程模式

1. 多线程模型下可以创建子线程
2. 子进程/线程创建成功后进入 while 循环,阻塞在 recv 调用上,等待客户端向服务器发送数据
3. 收到数据后服务器程序进行处理然后使用 send 向客户端发送响应
4. 当客户端连接关闭时,子进程/线程退出并销毁所有资源。主进程/线程会回收掉此子进程/线程。

缺点

1. 这种模型严重依赖进程的数量解决并发问题
2. 启动大量进程会带来额外的进程调度消耗

异步非阻塞

现在各种高并发异步 IO 的服务器程序都是基于 epoll 实现的

IO 复用异步非阻塞程序使用经典的 Reactor 模型,Reactor 顾名思义就是反应堆的意思,它本身不处理任何数据收发。只是可以监视一个 socket 句柄的事件变化

reactor 模型

Add:添加一个 SOCKET 到 Reactor

Set:修改 SOCKET 对应的事件,如可读可写

Del:从 Reactor 中移除

Callback:事件发生后回调指定的函数

常见 reactor 模型

1. Nginx:多线程 Reactor
2. Swoole:多线程 Reactor+多进程 Worker

微信二维码扫一扫咨询大神进阶课程内容

微信号码 若兰：zqf19907493857

QQ 咨询联系 妮妮：194210485