# 并发编程模型

## 进程和线程的区别和联系

一个程序至少有一个进程,一个进程至少有一个线程。

1. 根本区别：进程是操作系统资源分配的基本单位，而线程是任务调度和执行的基本单位;
2. 开销方面：每个进程都有独立的代码和数据空间（程序上下文），程序之间的切换会有较大的开销；线程可以看做轻量级的进程，同一类线程共享代码和数据空间，每个线程都有自己独立的运行栈和程序计数器（PC），线程之间切换的开销小。
3. 所处环境：在操作系统中能同时运行多个进程（程序）；而在同一个进程（程序）中有多个线程同时执行（通过CPU调度，在每个时间片中只有一个线程执行）
4. 内存分配方面：系统在运行的时候会为每个进程分配不同的内存空间, 进程在执行过程中拥有独立的内存单元；

而对线程而言，除了CPU外，系统不会为线程分配内存（线程所使用的资源来自其所属进程的资源），线程组之间只能共享资源。

1. 包含关系：没有线程的进程可以看做是单线程的，如果一个进程内有多个线程，则执行过程不是一条线的，而是多条线（线程）共同完成的；线程是进程的一部分，所以线程也被称为轻权进程或者轻量级进程。

## java线程有OS(Linux)线程的关系

|  |  |  |
| --- | --- | --- |
| pthread\_t \*thread | 传出参数，调用之后会传出被创建线程的id | 定义 pthread\_t pid; 继而取地址获取&pid |
| const pthread\_attr\_t \*attr | 线程属性，关于线程属性是linux的知识 | 在学习pthread\_create函数的时候一般传NULL，保持默认属性 |
| void \*(\*start\_routine) (void \*) | 线程的启动后的主体函数，相当于java当中的run方法 | 需要你定义一个函数，然后传函数名即可 |
| void \*arg | 前面参数主体函数的参数 | 如果没有可以传NULL |

通过C++来调用Linux的一个线程：

**#include** <pthread.h>*//头文件***#include** <stdio.h>  
pthread\_t pid;*//定义一个变量，接受创建线程后的线程id  
//定义线程的主体函数***void**\* thread\_entity(**void**\* arg){  
 printf(**"i am new Thread!"**);  
}  
*//main方法，程序入口，main和java的main一样会产生一个*进程*，继而产生一个main*线程**int** main() {  
 *//调用操作系统的函数*pthread\_create*创建线程，注意四个参数* pthread\_create(&pid,NULL,thread\_entity,NULL);  
 *//usleep是睡眠的意思，那么这里的睡眠是让谁睡眠呢？  
 //为什么需要睡眠？如果不睡眠会出现什么情况*  usleep(**100**);  
 printf(**"main\n"**);  
}

这就说明了系统内核也是支持多线程；

thread.start()方法调用的核心是：

**private native void** start0();

是一个native 本地方法

java调用的是start0(),这个本地方法肯定是一个C或者C++文件

JDK包含的内容：

1. 一系列sun公司提供的java库
2. 本地C文件(作用：1、实现调用操作系统的函数‘、；2、调用JVM的代码hotspot；)
3. C++写的hotspot虚拟机这个项目；

本地方法C文件调用的应该是hotspot虚拟机的代码，而hotspot的c++代码来调用os的函数；

重点 - java调用start方法来开启线程：

java通过start()方法开启一个线程，就是通过hotspot代理最终调用的os的pthread\_create()方法来创建一个新线程；所以可以说JAVA的线程和操作系统OS的线程是一一对应的，即java对线程的操作方法都可以在OS上找到对应的操作；

start()是来调用系统的函数的，所以start()方法不能写实现，而是通过调用对应的run()方法来实现相应的逻辑；

C语言调用java方法

java调用java方法：通过class对象来new 对应的java对象，然后调用它的方法；

C语言调用也是一样的，通过JVM找到对应的class文件，然后new对象来调用方法：

1. 定义变量jclass clas对象；
2. clas = (\*env)->FindClass(envOld,“Test”); //Test是对于java文件名；
3. 定义一个变量来承接对象jobject obj；jmethod cid对应java的方法；
4. 找到class之后调用cid = (\*envOld)->GetMethodID(envOld,cls,”<init>”,”()v”)

这里调用的方法<init>为java的构造方法，()v则说明是无参的；

这里的cid的c是constructor 构造方法的缩写；

1. 然后获取实例化对象：

obj = (\*envOld)->NewObject(envOld,clas,cid);

1. 调用方法：

java为什么不通过本地C文件直接调用OS的相关函数：

假如直接调用，则这个C文件会被十分频繁的调用去与OS函数进行交互；所以它调用了hotspot的c++的javaThread，而他去通过对应的方法与OS进行交互；