

进程与线程

江学强, PB16120100

1. 进程是什么？

首先需要叙述一下进程和程序之间的关系，用一句话概括，可以理解为进程是执行中的程序。我们知道程序（可执行程序）必须要被加载到内存中才能被执行，我们可以理解此时就创建了一个相应的进程，这个进程占据部分计算机的资源进行执行。从这个叙述也可以看的出来程序和进程并非是一一对应的关系，因为同一个程序执行产生的进程往往不同（至少往往在系统层面是不同的）。

进程在执行过程中有不同的状态：新建状态（new）是指此进程刚被创建，执行状态（running）指此进程正在被处理器执行，等待状态（waiting）指进程正在等待一些条件达到以继续执行（例如等待 I/O 设备输入输出的完成），预备状态（ready）指进程可以被继续执行但是此时处理器正被占用，终止状态（terminated）指此进程已经结束。

2. 操作系统如何管理进程？

现在的计算机基本上在同一时刻会有很多不同的进程在执行，操作系统要对进程进行管理就必须能够对这些进程进行区分，所以每一个进程被创建之时都会在操作系统空间创建一个进程控制块（PCB），PCB 中记录一个能够唯一确定进程的编号（PID），这可以用来区分不同的进程。此外，PCB 中也保存着很多其余必要的信息，例如进程状态、当前执行指令的地址、寄存器信息、CPU 调度信息、内存管理信息、I/O 状态信息以及一些记录。

PCB 在进程管理中非常关键，我们知道处理器在同一时刻只能执行一个进程，但计算机上同一时刻事实上是有很多进程的，计算机是通过不停地从不同的进程之间跳转实现这样的功能的。在这种跳转过程中需要 PCB 提供很多必要的信息。

3. 与进程相关有哪些操作？

显然系统必须要提供必要的操作来管理进程，这些必要的操作有获取进程的编号（PID），创建进程，终止进程，执行进程等。

获取进程编号很简单，操作系统提供了相关的系统调用可以直接获取一个进程的 PID。

创建进程也是通过系统调用来完成，但是需要注意进程中可以产生其他进程，也就是有父进程与子进程，创建其他进程的进程称为父进程，被创建的进程称为子进程。以 Linux 操作系统为例，Linux 系统下的所有进程呈树结构，根进程为 init，是被创建的第一个进程。在父进程中创建子进程的系统调用是 fork()。需要注意的是调用 fork() 系统调用后，父进程和子进程都在执行同一个程序（子进程从 fork（）之后执行）。

执行进程是通过 exec 系列的系统调用实现，如果只是在一个进程中通过调用 exec 执行其他进程，则被调用的进程结束后不会返回调用进程，此外调用进程的跟执行相关的资源都会被释放。如果想要保留原来的进程，就需要先创建子进程再在子进程中执行其他进程中，但这会出现问题，就是无法确定原来进程和被 exec 执行的进程的执行顺序，如果想要被执行的进程先执行，就需要在调用进程中显式调用 wait（）系统调用。

4. 什么是线程以及为什么需要多线程？

维基百科上给出的线程的概念是：线程是操作系统能够进行运算调度的最小单位。它被包含在进程之中，是进程中的实际运作单位。一条线程指的是进程中一个单一顺序的控制流，一个进程中可以并发多个线程，每条线程并行执行不同的任务。

从线程的概念可以看出线程是包含在进程中的，一个进程中可以包含多个线程，这就是所谓的多线程任务。这些同一个进程中的进程共享了进程的很多资源，这比所有的线程都创建单独的进程节省了很多资源和创建的时间。事实上多线程还有其他的好处，比如多线程的响应性更好，多线程分担了任务；同一进程中的线程进行通信更加容易（这是显而易见的），以及前面讲到的资源和时间的节省。

5. 多线程模型

线程可以分为用户线程和内核线程，根据这两种线程之间的关系可以总结出一些线程的模型：多对一模型：多个用户线程被映射成一个内核线程；一对一模型：每一个用户线程对应一个内核线程；多对多模型：多个用户线程对应到多个内核线程。

6. 进程和线程的区别与联系

进程是指在内存中运行的程序，独立占有计算资源，但是线程是进程内相对独立的计算单元，并不独立占用内存；线程必须依附进程存在，一个进程中可以有多个线程。

参考文献格式：

[1] Abraham Silberschatz. 操作系统概念. 高等教育出版社, 2007.3.