进程与线程的对比

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 进程 | 线程 |
| 不同点 | 含义 | 进程是资源分配的最小单位。每个进程空间是完全独立的。（资源主要指进程空间） | 可以被调度的最小单位。进程内部的所有线程，共享进程资源。 |
| 函数 | fork(),wait(),exit(),  exec() | Pthread\_creat,pthread\_join,  pthread\_exit,pthread\_detach() |
| 使用场合 | 只有在加载新程序时，才会涉及多线程 | 当程序设计多线任务时，才会使用多线程 |
| 资源保护 | 因为进程空间的独立性，资源保护天然就很到位。  对于进程来说，因为进程空间的独立性，因此进程资源的保护很到位，反倒是进程间共享数据很困难，因此OS提供了管道、消息队列等进程间的通信机制。 | 对于线程来说，由于它们共享进程空间，因此线程之间可以通过全局变量很容易地实现数据共享。不过这种容易反而造成了数据被篡改的可能性大大提高。因此相对于进程，线程反而要注重怎样去保护资源（数据）被篡改。 |
| 相同点 | 不管是进程还是线程，OS都会为它们各自创建一个task\_struct变量，用于存放进程和线程管理信息。 | | |
| 备注 | 当我们的程序涉及多线任务时，一定要使用多线程来实现，千万不要使用多进程来实现，否则程序的运行将占用大量的系统资源。  只有当你的程序有执行新程序的需求时，程序才会涉及到多进程。 | | |