



链滴

# linux内核源码情景分析 - 进程与进程调度（一）

作者: [xtestw](#)

原文链接: <https://ld246.com/article/1456469935055>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

### 进程四要素：

- 

- 有一段程序供其执行。（代码段）

- 进程**专用**的**系统**堆栈空间

- 内核中有一个task\_struct的数据结构，此数据结构是内核进程调度的基本单位。

- 独立的存储空间，即**专有**的**用户**空间。（私：包括数段和程序堆栈）



linux系统对于进程和线程（此线程不同于用户空间同一进程实现的线程）的判定就是是否具有四条规则，第四条规则强调的是专有的用户空间。如果是共用用户空间，则是用户线程，如果一点用户空间都没有的话，就是内核线程。

intel在i386新增了一个段，叫“任务状态段”TSS。是一个104字节的数据结构，记录任务关键的状态信息。用于任务切换和恢复（linux中任务等同于进程）

每个进程都有一个task\_struct结构体和用于系统空间堆栈的存储空间。内核分配了连续的2物理页（8192字节，8k），其中底部1k是用于task\_struct结构体，上面7k则是系统堆栈（有小限制，不是很大）。

进程状态分3个，在task\_struct的结构体中有一个state反应当前运行状态：

- 

- state = - 1, 表示unrunable

- state = 0, 表示runable

- 

state > 0, stopped:

- 

- TASK\_INTERRUPTIBLE 1

- TASK\_UNINTERRUPTIBLE 2

- TASK\_ZOMBIE 4

- TASK\_STOPPED 8



进程包含3个重要队列：

- 

- 宗族关系队列，进程关系，指向父进程和最新子进程的指针，指向兄弟进程的指针。

- pid映射的队列，通过hash计算，在此队列中找一个下标，顺着此下标查找进程。

- 用于遍历所有进程的线性队列。init\_task进程为队首。








### 进程创建，执行，消亡

fork/clone/vfork

- 

- fork创建的进程是有自己的用户空间的，与父进程不共用资源，但是复制所有资源。

- clone部分资源以指针方式复制（与父进程共享部分资源）

- vfork全部以指针方式复制（与父进程共享全部资源），就是线程。



exeve()函数可以用来执行以文件方式存在的可执行程序映象。

子进程创建后，父进程3个选择：

- 

- 子进程先父进程“去世”，父进程会收到一个信号。

- 父进程等待子进程结束。linux提供了2个系统调用，wait3()&&wait4(), wait3等待任何一个子程去世，wait4等待某个特殊的子进程去世。

- 父进程先子进程去世，自行结束。



其实就是同步和异步两种选择。