

前缀的状态是运行状态(用于 state 字段),而EXIT前缀的状态是退出状态(用于 exit\_state 字段);三是有几种状态是复合状态;四是排序规则主要按历史沿革递增,未根据内在含义分组。

因此需要总结归纳以方便理解。根据 state 和 exit\_state 的取值,进程状态可以分成“活动时状态”和“死亡后状态”两大类。

**活动时:** 进程的运行状态(state 字段)为新创生(TASK\_NEW)<sup>1</sup>、可运行(TASK\_RUNNING)、可中断可杀死的睡眠(TASK\_INTERRUPTIBLE)、不可中断可杀死的睡眠(TASK\_KILLABLE)、不可中断不可杀死的睡眠(TASK\_UNINTERRUPTIBLE)、不可中断不可杀死的空闲睡眠(TASK\_IDLE)、暂停式睡眠(TASK\_PARKED)<sup>2</sup>、正在唤醒(TASK\_WAKING)、停止(TASK\_STOPPED)、跟踪(TASK\_TRACED)等10种状态之一;退出状态(exit\_state 字段)一律为0。

**死亡后:** 进程的运行状态(state 字段)为死亡(TASK\_DEAD);退出状态(exit\_state 字段)为死亡(EXIT\_DEAD)、僵尸(EXIT\_ZOMBIE)或跟踪僵尸(EXIT\_TRACE)。如果一个进程优雅地死亡,退出状态应当是EXIT\_DEAD;如果死亡后有关数据结构没有销毁,退出状态就是EXIT\_ZOMBIE(普通僵尸进程,通常是由于父进程没有或尚未收尸)或EXIT\_TRACE(被跟踪的僵尸进程,通常其父进程是调试跟踪器)。

进程的主要状态变化如图5-1所示,通常进程活动时的状态以可运行(即TASK\_RUNNING,包括就绪态和运行态)和睡眠等待(主要包括TASK\_INTERRUPTIBLE和TASK\_UNINTERRUPTIBLE)居多。一般来说,可运行的进程组织在运行队列中,而睡眠等待的进程根据不同的情况组织在专门针对某类事件的等待队列中(不等待任何事件的“纯睡眠”进程不进入任何等待队列,但一定会离开运行队列)。有一系列睡眠函数可以让进程从可运行状态变更为睡眠状态,也有另外一系列唤醒函数可以让进程从睡眠状态变更为可运行状态(如果睡眠的进程没有进入等待队列,则不需要专门的唤醒操作)。

除了“睡眠”和“唤醒”,在进程管理里面还有“冻结”和“解冻”的概念。冻结(freeze)就是一种非主动的强制睡眠,内核有一个“全局冰箱”,用全局变量system\_freezing\_cnt的状态来记录冰箱的状态(初始值为0)。当内核需要冻结进程时,就增加system\_freezing\_cnt的值,然后给进程发送一个伪信号,进程收到伪信号后调用try\_to\_freeze(),根据system\_freezing\_cnt是否非零来决定要不要进入不可中断的睡眠,如果进入了睡眠,就称为进程被冻结。解冻(thaw)是冻结的逆操作,其作用就是减少system\_freezing\_cnt的值并唤醒冰箱中被冻结的进程。普通进程总是可冻结的;内核线程默认是不可冻结的,但是可以通过set\_freezable()来设置成可冻结;可冻结的内核线程通过显式调用try\_to\_freeze()来试图进入冻结状态。冻结和解冻主要用于系统级睡眠(STR和STD,见第8章电源管理解析)。

1 新创生(TASK\_NEW)是从Linux-4.8版本内核开始引入的一种状态,标识一个正在创建中的或者刚创建尚未开始运行的进程。

2 暂停式睡眠用于内核线程,类似于TASK\_INTERRUPTIBLE,用来防止一些竞争条件。