第 3 部分总结与回顾

至此，第3部分讲述的一个新进程的诞生，就全部结束啦！恭喜你又渡过了一道难关！

整个第3部分，用前4回的内容讲述了**进程调度机制**，又用后3回内容讲述了 fork 函数的全部细节。我们一起来回顾一下。

进程调度机制

前4回内容循序渐进地讲述了进程调度机制的设计思路和细节。

- **第 21 回 | 第 3 部分全局概述**
- **第 22 回 | 从内核态切换到用户态**
- **第 23 回 | 如果让你来设计进程调度**
- **第 24 回 | 从一次定时器滴答来看进程调度**

进程调度的始作俑者，就是那个每 10ms 触发一次的定时器滴答。

而这个滴答将会给 CPU 产生一个**时钟中断**信号。

而这个中断信号会使 CPU 查找中断向量表，找到操作系统写好的一个时钟中断处理函数 do_timer。

do_timer 会首先将当前进程的 counter 变量减1，如果 counter 此时仍然大于 0，则就此结束。

但如果 counter 等于 0 了，就开始进行进程的调度。

进程调度就是找到所有处于 RUNNABLE 状态的进程，并找到一个 counter 值最大的进程，把它丢进 switch_to 函数的入参里。

switch_to 这个终极函数，会保存当前进程上下文，恢复要跳转到的这个进程的上下