

内核fork.c函数中的部分代码分析Linux认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022__E5_86_85_E6_A0_B8fork_c103_645066.htm 在内核代码 2.6.15.5

中/kernel/fork.c第1255 - 1261中有如下代码：1. p = copy_process(clone_flags, stack_start, regs, stack_size, parent_tidptr, child_tidptr, pid). 2. if (!IS_ERR(p)) { 3. struct completion vfork. 4. if (clone_flags & CLONE_VFORK == CLONE_VFORK). 7. }

为了方便描述我在这段代码上加了行号。第一行首先通过copy_process()函数完成具体的进程创建工作，返回值类型为task_t类型。第2行用函数IS_ERR()分析copy_process()的返回值是否正确。如果正确则执行第3 - 7行代码。这里分析一下接下来这几行代码：struct completion vfork. //定义struct completion 类型的变量 vfork. 关于struct completion的定义如下：struct completion { unsigned int done; wait_queue_head_t wait; } 第4行判断clone_flags中是否有CLONE_VFORK标志。如果有则执行下面的代码：p->vfork. init_completion(&p->done = 0). init_waitqueue_head(&p->wait). } 这个函数的作用是在进程创建的最后阶段，父进程会将自己设置为不可中断状态，然后睡眠在等待队列上(init_waitqueue_head()函数就是将父进程加入到子进程的等待队列)，等待子进程的唤醒。100Test 下载频道开通，各类考试题目直接下载。详细请访问

www.100test.com