

linux进程介绍之forkvsfork PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/145/2021\\_2022\\_linux\\_E8\\_BF\\_9B\\_E7\\_A8\\_c103\\_145342.htm](https://www.100test.com/kao_ti2020/145/2021_2022_linux_E8_BF_9B_E7_A8_c103_145342.htm) 在fork之后，子进程和父进程都会继续执行fork调用之后的指令。子进程是父进程的副本。它将获得父进程的数据空间，堆和栈的副本，这些都是副本，父子进程并不共享这部分的内存。也就是说，子进程对父进程中的同名变量进行修改并不会影响其在父进程中的值。但是父子进程又共享一些东西，简单说来就是程序的正文段。正文段存放着由cpu执行的机器指令，通常是read-only的。由于在fork之后我们常常都是跟个exec在后面，所以为了提高效率，很多的实现并不完全复制数据段和堆、栈，而是采用写时复制，有点类似于某些cache与内存数据的同步方法。另一种提高效率的方法就是使用vfork，vfork最早起源于2.9BSD，它与fork的不同就在于它并不将父进程的地址空间完全复制到子进程中，因为子进程会立即调用exec.vfork出来的子进程是在父进程的空间中运行的，它的存在就是为了exec调用，所以它不需要复制这些东西，因为复制了也没有用。如果这时子进程修改了某个变量，这将影响到父进程。vfork与fork的另一区别是：vfork保证子进程先运行，在它调用exec或exit后父进程才可能调度运行。而fork的父子进程运行顺序是不定的，它取决于内核的调度算法。所以，fork的时候，程序代码被复用了我指的程序代码就是由cpu执行的机器指令部分，这与有多少个进程在运行无关，即使是频繁执行的程序在存储器中也只需一个副本，而且它在执行期可能是read-only的。当然，如果你exec了，那就是另一码事了。另外，父进程中

的数据空间和堆、栈可能会产生副本，具体情况要看你使用的是vfork还是fork.fork会产生副本，而vfork则共享这部分内存。 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](http://www.100test.com)