进程创建



系统允许一个进程创建新进程,新进程即为子进程,子进程还可以创建新的子进程,形成进程树结构模型。

```
#include <sys/types.h>
#include <unistd.h>
pid_t fork(void);
```

返回值:

- 成功: 子进程中返回 O, 父进程中返回子进程 ID
- 失败:返回 -1

失败的两个主要原因:

- 1. 当前系统的进程数已经达到了系统规定的上限,这时 errno 的值被设置 为 EAGAIN
- 2. 系统内存不足, 这时 errno 的值被设置为 ENOMEM



内核区 Linux kernel pid=10086 环境变量 命令行参数 栈空间 pid=10089在栈空间里面 num=9 用户区 共享库 堆空间 .bss .data .text 受保护的地址(0~4k)



Linux kernel pid=10089 环境变量 命令行参数 栈空间 pid=0在栈空间里面 num=9 共享库 堆空间 .bss .data .text

受保护的地址 (0~4k)

fork以后,子进程 的用户区数据和父 进程一样,内核区 也会拷贝过来,但 是pid在栈空间不 一样,而对于num 来说一样的,其互 不影响



内核区 用户区 栈空间 num=10 共享库 堆空间

Linux的fork()使用是通过写时拷贝 (copy-on-write)实现的,写时拷贝 是一种可以推迟甚至避免拷贝数据 的技术, 内核此时并不复制整个进 程的地址空间,而是让父子进程共 享同一个地址空间, 只用在需要写 入的时候才会复制地址空间,从而 使得各个进程拥有各自的地址空 间;也就是说,资源的复制是在需 要写入的时候才会进行, 在此之 前,只有以只读方式共享。 注意: fork之后的父子进程共享文 件,fork产生的子进程与父进程相 同的文件描述符指向相同的文件 表, 引用计数增加, 共享文件偏移

指针。

内核区 用户区 栈空间 num = 12共享库 堆空间



使用 GDB 调试的时候, GDB 默认只能跟踪一个进程, 可以在 fork 函数调用之前, 通过指令设置 GDB 调试工具跟踪父进程或者是跟踪子进程, 默认跟踪父进程。

设置调试父进程或者子进程: set follow-fork-mode [parent (默认) | child]

设置调试模式: set detach-on-fork [on | off] 默认为 on, 表示调试当前进程的时候, 其它的进程继续运行, 如果为 off, 调试当前进程的时候, 其它进程被 GDB 挂起。

查看调试的进程: info inferiors

切换当前调试的进程: inferior id

使进程脱离 GDB 调试: detach inferiors id





THANKS



关注【牛客大学】公众号 回复"牛客大学"获取更多求职资料