

# linux kill -9 杀不掉的进程

kill -9 发送SIGKILL信号给进程，将其终止，但对于以下两种情况不适用

- 1.该进程是僵尸进程（STAT z），此时进程已经释放所有的资源，但是没有被父进程释放。僵尸进程要等到父进程结束，或者重启系统才可以被释放。
- 2.进程处于“核心态”，并且在等待不可获得的资源，处于“核心态”的资源默认忽略所有信号。只能重启系统。

kill 只能杀死处于用户状态的进程。

下面是一个自测试例子：

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <errno.h>

int main(int argc ,char *argv[])
{
    pid_t pid;

    pid = fork();
    if(pid<0){
        perror("fork");
        return -1;
    }else if(pid==0){
        printf("I am a child\n");
        exit(0);
    }else{
        while(1){
            sleep(100);
        }
    }
    return 0;
}
```

由于父进程没有退出，所以执行`ps -aux | grep "z"`，可以查看进程的状态，发现如下（绿色标出部分）

```
root    1937  0.0  0.1 389000 4336 ?      Sl  09:12  0:00 /usr/bin/gnome-keyring-
daemon --daemonize --login
root    4447  0.0  0.0 112680  964 pts/2   R+   10:16  0:00 grep --color=auto z
[root@localhost linux_test]# ps -aux | grep "[zZ]"
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.1 128164 6828 ?        Ss   09:07   0:01 /usr/lib/systemd/systemd --
switched-root --system --deserialize 21
root    1937  0.0  0.1 389000 4336 ?      Sl  09:12  0:00 /usr/bin/gnome-keyring-
daemon --daemonize --login
root    4385  0.0  0.0    0    0 pts/0   Z+  10:16  0:00 [a.out] <defunct>
root    4455  0.0  0.0 112680  976 pts/2   S+   10:17  0:00 grep --color=auto [zZ]
```

从以上信息 可以得到该进程的进程号是4385

此时的解决方法有两种

《1》 `cat /proc/4385/status` 找到该子进程对应的父进程，将其父进程杀死

State: Z (zombie)

Tgid: 4385

Ngid: 0

Pid: 4385

PPid: 4384

执行`kill -9 4384` 如果父进程也杀不死，那就只能执行重启了

《2》 重启