

# 孤儿进程

维基百科，自由的百科全书

在操作系统领域中，**孤儿进程**指的是在其父进程执行完成或被终止后仍继续运行的一类进程。

## 目录

### 解决办法

“收养”

进程组

远程调用的情况

“孤儿进程”的应用

参见

参考

## 解决办法

### “收养”

在类UNIX操作系统中，为避免孤儿进程退出时无法释放所占用的资源而僵死，任何孤儿进程产生时都会立即为系统进程init或systemd自动接收为子进程，这一过程也被称为“收养”（英语：re-parenting）<sup>[1]</sup>。在此需注意，虽然事实上该进程已有init作为其父进程，但由于创建该进程的进程已不存在，所以仍应称之为“孤儿进程”。

### 进程组

因为父进程终止或崩溃都会导致对应子进程成为孤儿进程，所以也无法预料一个子进程执行期间是否会被“遗弃”。有鉴于此，多数类UNIX系统都引入了进程组以防止产生孤儿进程：在父进程终止后，用户的Shell会将父进程所在进程组标为“孤儿进程组”，并向终止的进程下属所有子进程发出SIGHUP信号，以试图结束其运行，如此避免子进程继续以“孤儿进程”的身份运行<sup>[2]</sup>。

### 远程调用的情况

RPC过程中也会产生孤儿进程。例如，若客户端进程在发起请求后突然崩溃，且对应的服务器端进程仍在运行，则该服务器端进程就会成为孤儿进程。这样的孤儿进程会浪费服务器的资源，甚至有耗尽资源的潜在危险，但也有对应的解决办法<sup>[3]</sup>：

- 终止机制：强制杀死孤儿进程（最常用的手段）；
- 再生机制：服务器在指定时间内查找调用的客户端，若找不到则直接杀死孤儿进程；
- 超时机制：给每个进程指定一个确定的运行时间，若超时仍未完成则强制终止之。若有需要，亦可让进程在指定时间耗尽之前申请延时。

## “孤儿进程”的应用

除此之外，用户也可能会刻意使进程成为孤儿进程，以使之与用户会话脱钩，并转至后台运行。这一做法常应用于启动需要长时间运行的进程，也即守护进程<sup>[4]</sup>。另外，UNIX命令nohup也可以完成这一操作<sup>[5]</sup>。

## 参见

---

- 僵尸进程
- 守护进程

## 参考

---

- Robert Love. Linux Kernel Development. Novell Press.
  - GNU C Library Manual. GNU Project. |chapter=被忽略 (帮助)
  - Er. Vivek Sharma, Er. Manish Varshney, Shantanu Sharma. Design and Implementation of Operating System. Laxmi Publication Pvt. Ltd.
  - 杜华. 8.2.1 实现守护进程的步骤. Linux编程技术详解. 人民邮电出版社. 2007.
  - nohup(1) - Linux man page.
- 

取自 “<https://zh.wikipedia.org/w/index.php?title=孤儿进程&oldid=47978574>”

---

本页面最后修订于2018年1月24日 (星期三) 13:50。

本站的全部文字在知识共享 署名-相同方式共享 3.0协议之条款下提供，附加条款亦可能应用。（请参阅使用条款）  
Wikipedia®和维基百科标志是维基媒体基金会的注册商标；维基™是维基媒体基金会的商标。  
维基媒体基金会是按美国国内税收法501(c)(3)登记的非营利慈善机构。