参考资料

1. fork()函数详解

<https://blog.csdn.net/weixin_51609435/article/details/124849719>

1. wait()函数及僵尸进程、孤儿进程介绍

<https://blog.csdn.net/m0_46251547/article/details/124784761>

1. Linux环境下printf输出问题

<https://sauron7i.blog.csdn.net/article/details/121294374>

1. 关于Linux输入输出缓冲区的问题

<https://blog.csdn.net/qq_49959794/article/details/115420196>

<https://blog.csdn.net/XITMan/article/details/107534915>

<https://blog.csdn.net/TYnfd/article/details/119683293>

1. 换行符对缓冲区的刷新作用：

<https://blog.csdn.net/qq_37284607/article/details/112470262>

<https://blog.csdn.net/qq_43701555/article/details/93632292>

printf输出语句存在”缓冲区”的问题：在Linux系统中，printf并不会马上将数据直接输出到屏幕，而是将输出结果暂存在缓冲中，当缓冲区满、刷新缓冲区或程序结束时，缓冲区的内容才会进行输出。刷新缓冲区的方式有两种：fllush(stdout)或使用换行符 \n 刷新缓冲区。事实上，我们熟悉的函数getchar()也使用到了缓冲区：用户输入的字符被存在缓冲区中，直到用户按下回车键。

1. Linux父子进程变量地址相同问题

<https://blog.csdn.net/weixin_45636061/article/details/124610215>

1. system()函数介绍

<https://blog.csdn.net/weixin_42645653/article/details/121587379?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522166633664516800180696932%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request_id=166633664516800180696932&bi>

1. exec族函数介绍及举例

<https://blog.csdn.net/weixin_44575952/article/details/125565152?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522166633785416782412537527%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=166633785416782412537527&biz_id=0&utm>

1. 孤儿进程、僵尸进程的观察

孤儿进程是指，若子进程还在运行，但是父进程已经退出，那么该进程会成为孤儿进程，并且将会被init进程所收养。为了观察到孤儿进程，我们要保证父进程在子进程之前退出，例如，可以在子进程中，加入语句sleep(10),使其休眠10s，以便让父进程先于子进程退出，子进程成为孤儿进程。可以通过在终端执行命令ps-aux查看孤儿进程，或getppid()查看父进程号。如果子进程退出，但是父进程并没有用wait(NULL)函数回收，那么子进程的PCB仍然在系统中，我们称这样的进程为僵尸进程。为了可以观察到僵尸进程，我们要让子进程退出后的一段时间内，父进程没有对其进行回收。

实验中，可以在父进程中的wait()语句前加入sleep(20)让其休眠20s，这样子进程就会有20s的僵死状态。在Linux系统中，我们可以用命令 ps - ef | grep PID 来查看当前的进程的状态；也可以用命令 ps - ef | grep defunct直接列出僵尸进程(defunt进程即为僵尸进程)。

例如，在一个终端中，我们执行图1所示程序，运行结果如图2所示，可以看到子进程号为3669，此时再打开另一个终端，在20s内分别键入ps - ef | grep 3669和 ps - ef | grep defunct，运行结果为图3 所示。从图3可以看到，在执行ps - ef | grep defunct后，系统显示PID号为3669的进程为僵尸进程。执行ps - ef | grep 3669后，其状态也为defunct，即为僵尸进程。在20s过后，我们再次执行ps - ef | grep defunct，结果如图4，此时子进程已经被父进程通过wait()回收，不再是僵尸进程。



图1 观察僵尸进程的程序

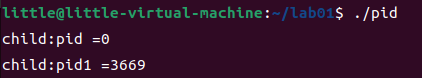
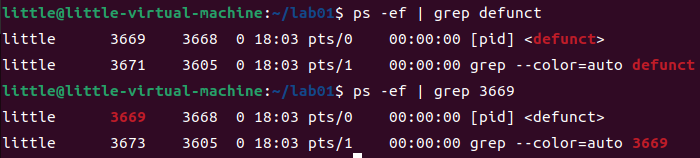


图2程序执行结果



IMG_256图3 执行ps - ef | grep 3669 和 ps - ef | grep defunct结果

图4 20s后执行ps - ef | grep defunct结果

1. Linux虚拟内存介绍

<https://zhuanlan.zhihu.com/p/352366788>

1. 如何创建线程

<https://blog.csdn.net/qq_44824574/article/details/110672780?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522166642173916800180666268%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=166642173916800180666268&biz_id=0&utm_med>

1. undefined reference to ‘phread\_join’问题的解决

<https://blog.csdn.net/weixin_44653395/article/details/100079447>

1. 使用信号量实现线程同步

<https://blog.csdn.net/hhaijiuwo/article/details/119568497>

1. 使用原子操作实现线程同步

<https://blog.csdn.net/piaopiaopiaopiaopiao/article/details/104407515>

1. 获取线程PID、TID

<https://www.cnblogs.com/lakeone/p/3789117.html>

1. 在线程实验中，虽然引用了<pthread.h>头文件，但还是报错。

<https://blog.csdn.net/weixin_42676396/article/details/123993146>

pthread 库不是 Linux 系统默认的库，连接时需要使用静态库，编译时增加一个参数即可:gcc -o pro7 pro7.c -lpthread

1. 软中断通信中，SIGINT与SIGQUIT中断信号的不同

SIGINT和SIGQUIT都是强制结束前台所有进程，不同点是SIGQUIT文件生成了core文件。core文件是分析内存错误的有用的文件，可以用这个命令查看core文件的位置：cat /proc/sys/kernel/core\_pattern。

<https://blog.csdn.net/weixin_39714164/article/details/111950203>

<https://blog.csdn.net/m0_37865510/article/details/86292453>

<https://blog.csdn.net/weixin_44713496/article/details/123831251>

<https://blog.csdn.net/lys07962000/article/details/60774406>

1. 进程间的管道通信

<https://blog.csdn.net/skyroben/article/details/71513385>

1. Linux中write函数详解

<https://blog.csdn.net/Arrany/article/details/123345928>

1. 管道通信的底层实现

创建一条管道完成下述核心工作：①分配一个隶属于root文件系统的磁盘和内存索引结点inode；②在系统打开文件表中分别分配一读管道文件表项和一写管道文件表项；③记录偏移量filedes[0]和filedes[1]分别指向系统打开文件表的读和写管道文件表项。

<https://blog.csdn.net/TyearLin/article/details/123552781>

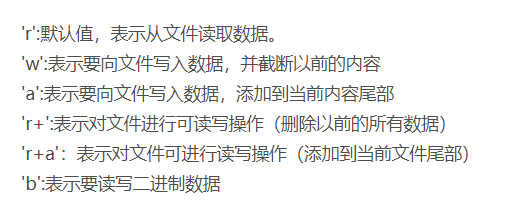
<https://blog.csdn.net/Once_fish_/article/details/124234132>

1. 如何创建一个指定大小(size)的txt文件？

用命令*fsutil file createnew [filename].txt [size]* 来创建指定大小的文件，填充filename以及size。

1. 用文件模拟磁盘意味着我们要一次性以二进制读写512byte的数据，那么如何实现以二进制数据读写，而不是将读取的字节转化为二进制？

文件有多种使用标识，将fopen()函数的mode字段设置成”rb+”即可，就可以实现以二进制的方式读写数据。



1. 用文件模拟磁盘意味着我们要找到指定的位置进行读写，那我们如何将读写指针移动到我们所想的位置进行读写操作？

可以用函数：fseek ( FILE \*fp, long offset, int origin )，设置offset这一项，就可以将读写指针移动到距离文件起始位置offset字节处(origin=0)。

1. 文件系统实验中，关于文件访问时间和修改时间的显示，C语言的相关函数。

<https://blog.csdn.net/u012229282/article/details/79598287>

<https://blog.csdn.net/weixin_42298373/article/details/117083379>

1. 创建一个Linux内核模块

https://zhuanlan.zhihu.com/p/446783415

1. 清除内存写保护位才可对系统调用进行操作。

<https://zhuanlan.zhihu.com/p/446784403>

1. KASLR(内核地址空间布局随机化)机制会造成每次启动时内核空间中内容的地址随机偏移,所以地址不同，关闭KASLR即可。

<https://blog.csdn.net/zy_zhengyang/article/details/95451484>