

Linux程序设计入门系统进程的详细介绍 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/252/2021_2022_Linux_E7_A8_8B_E5_BA_c103_252952.htm

1、进程的概念 Linux操作系统是面向多用户的.在同一时间可以有許多用户向操作系统发出各种命令.那么操作系统是怎么实现多用户的环境呢? 在现代的操作系统里面,都有程序和进程的概念.那么什么是程序,什么是进程呢? 通俗的讲程序是一个包含可以执行代码的文件,是一个静态的文件.而进程是一个开始执行但是还没有结束程序的实例.就是可执行文件的具体实现. 一个程序可能有許多进程,而每一个进程又可以有許多子进程.依次循环下去,而产生子孙进程. 当程序被系统调用到内存以后,系统会给程序分配一定的资源(内存,设备等等)然后进行一系列的复杂操作,使程序变成进程以供系统调用.在系统里面只有进程没有程序,为了区分各个不同的进程,系统给每一个进程分配了一个ID(就象我们的身份证)以便识别. 为了充分的利用资源,系统还对进程区分了不同的状态.将进程分为新建,运行,阻塞,就绪和完成五个状态. 新建表示进程正在被创建,运行是进程正在运行,阻塞是进程正在等待某一个事件发生,就绪是表示系统正在等待CPU来执行命令,而完成表示进程已经结束了系统正在回收资源. 关于进程五个状态的详细解说我们可以看《操作系统》上面有详细的解说。

2、进程的标志 上面我们知道了进程都有一个ID,那么我们怎么得到进程的ID呢?系统调用getpid可以得到进程的ID,而getppid可以得到父进程(创建调用该函数进程的进程)的ID. #include <unistd> pid_t getpid(void). pid_t getppid(void).进程是为程序服务的,而程序是为了用户服务的.

系统为了找到进程的用户名,还为进程和用户建立联系.这个用户称为进程的所有者.相应的每一个用户也有一个用户ID.通过系统调用getuid可以得到进程的所有者的ID.由于进程要用到一些资源,而Linux对系统资源是进行保护的,为了获取一定资源进程还有一个有效用户ID.这个ID和系统的资源使用有关,涉及到进程的权限.通过系统调用geteuid我们可以得到进程的有效用户ID.和用户ID相对应进程还有一个组ID和有效组ID系统调用getgid和getegid可以分别得到组ID和有效组ID。

```
#include <unistd.h> #include <sys/types.h> uid_t getuid(void).
uid_t geteuid(void). gid_t getgid(void). gid_t getegid(void). 有时候
我们还会对用户的其他信息感兴趣(登录名等等),这个时候我们
可以调用getpwuid来得到。 struct passwd { char *pw_name. /*
登录名称 */ char *pw_passwd. /* 登录口令 */ uid_t pw_uid. /* 用
户ID */ gid_t pw_gid. /* 用户组ID */ char *pw_gecos. /* 用户的真
名 */ char *pw_dir. /* 用户的目录 */ char *pw_shell. /* 用户
的SHELL */ }. #include <pwd.h> #include <sys/types.h> struct
passwd *getpwuid(uid_t uid).下面我们学习一个实例来实践一下
上面我们所学习的几个函数: #include <unistd.h> #include
<pwd.h> #include <sys/types.h> #include <stdio.h> int
main(int argc,char **argv) { pid_t my_pid,parent_pid. uid_t
my_uid,my_euid. gid_t my_gid,my_egid. struct passwd *my_info.
my_pid=getpid(). parent_pid=getppid(). my_uid=getuid().
my_euid=geteuid(). my_gid=getgid(). my_egid=getegid().
my_info=getpwuid(my_uid). printf("Process ID:%ld ",my_pid).
printf("Parent ID:%ld ",parent_pid). printf("User ID:%ld ",my_uid).
printf("Effective User ID:%ld ",my_euid). printf("Group ID:%ld
```

```
",my_gid). printf("Effective Group ID:%ld ",my_egid): if(my_info) {  
printf("My Login Name:%s " ,my_info->pw_name). printf("My  
Password :%s " ,my_info->pw_passwd). printf("My User ID :%ld  
",my_info->pw_uid). printf("My Group ID :%ld ",my_info->  
> pw_gid). printf("My Real Name:%s " ,my_info->pw_gecos).  
printf("My Home Dir :%s " , my_info->pw_dir). printf("My Work  
Shell:%s " , my_info->pw_shell). } }3、进程的创建 创建一个进  
程的系统调用很简单.我们只要调用fork函数就可以了.
```

#include <unistd.h> pid_t fork(). 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