<http://blog.chinaunix.net/uid-122937-id-3047193.html>

**1.进程状态概述**  
       在linux中通过task\_struct结构体来描述进程的PCB,我们可以在include/linux/sched.h中看到对进程task\_struct的定义和进程状态的描述。  
  
在include/linux/sched.h 中我们可以看到Linxu中进程状态的具体实现：  
       #define TASK\_RUNNING                   0  
       #define TASK\_INTERRUPTIBLE         1  
       #define TASK\_UNINTERRUPTIBLE    2  
       #define TASK\_ZOMBIE                     4  
       #define TASK\_STOPPED                   8  
其中：  
       TASK\_RUNNING，运行状态：进程正在运行或在运行队列中等待运行 。  
       TASK\_INTERRUPTIBLE，可中断等待状态：进程正在等待某个事件完成（如等待数据到达）。等待过程中可以被信号或定时器唤醒。  
       TASK\_UNINTERRUPTIBLE，不可中断等待状态：进程正在等待某个事件完成并且等待中不可以被信号或定时器唤醒，必须一直等待到事件发生。  
       TASK\_ZOMBIE，僵死状态：进程已终止，但进程控制块尚未注销，进程描述符依然存在，直到父进程调用wait()函数后释放。  
       TASK\_STOPPED，挂起/停止状态：进程因为收到SINSTOP,SIGSTP,SIGTIN,SGIOU信号后停止运行或者该进程正在被跟踪。  
  
**2.进程的基本信息获取**  
主要通过读取进程控制块PCB中的信息。  
(1)getpid()  
功能：用来获取目前进程的进程标识。  
定义函数：pid\_t getpid(void)  
返回值：返回当前进程的进程识别号。  
头文件：#include   
  
(2)getppid()  
功能：用来获取目前进程的父进程标识。  
定义函数：pid\_t getppid(void)  
返回值：返回当前进程的父进程识别号。  
头文件：#include   
  
(3) getuid()  
功能：获得进程的实际用户ID  
定义函数：pid\_t getuid(void)   
返回值：返回进程的实际用户ID  
              实际用户ID（uid) ：标识运行该进程的用户  
例如：一个普通用户A，运行了一个程序，而这个程序是以root 身份来运行的，则程序运行时就具有root 权限。此时，实际用户ID时A用户的ID，而有效用户ID是root用户ID  
头文件：#include   
  
(4) geteuid()  
功能：获得进程的有效用户ID  
定义函数：pid\_t geteuid(void)   
返回值：返回进程的有效用户ID  
头文件：#include   
  
(5)getpgid()  
功能：用来获得参数pid指令进程所属于的组识别号，若参数为0，则返回当前进程的组识别码。  
定义函数：pid\_t getpgid(pid\_t pid)  
返回值：执行成功则返回正确的组识别码，若有错则返-1，错误原因存在于errno中。  
头文件：#include   
  
(6)getpgrp()  
功能：用来获得目前进程所属于的组识别号，等价于getpgid(0)。  
定义函数：pid\_t getpgrp(void)  
返回值：执行成功则返回正确的组识别码。  
头文件：#include   
  
(7)getpriotity(void)  
功能：用来获得进程，进程组和用户的进程执行优先权。  
定义函数：int getpriority(int which,int who)  
参数含义：  
which:  
PRIO\_PROCESS   who为进程的识别码  
PRIO\_PGRP     who为进程的组识别码  
PRIO\_USER     who为用户识别码  
返回值：执行成功则返回当前进程的优先级(-20--20),值越小优先级越高。若出错则返-1，原因在errno中。  
头文件：#include   
  
**3.代码举例**  
**3.1 获取当前进程的PID信息**

1. #include <stdio.h>
2. #include <unistd.h>
3. #include <sys/resource.h>
4. int main(int argc,char \*\*argv)
5. {
6. printf("This process's pid is:%d",getpid());
7. printf("/nThis process's farther pid is:%d",getppid());
8. printf("/nThis process's group pid is:%d",getpgid(getpid()));
9. printf("/nThis process's group pid is:%d",getpgrp());
10. printf("/nThis process's priority is:%d/n",getpriority(PRIO\_PROCESS,getpid()));
11. return 0;
12. }