<http://blog.csdn.net/legendmaker/article/details/10494803>

在linux中查看线程数的三种方法  
最大线程数：cat /proc/sys/kernel/threads-max  
  
1、top -H  
手册中说：-H : Threads toggle  
加上这个选项启动top，top一行显示一个线程。否则，它一行显示一个进程。  
   
2、ps xH  
手册中说：H Show threads as if they were processes  
这样可以查看所有存在的线程。  
   
3、ps -mp <PID>  
手册中说：m Show threads after processes  
   
这样可以查看一个进程起的线程数。  
  
1.根据进程号进行查询：  
  
# pstree -p 进程号  
  
# top -Hp 进程号  
  
2.根据进程名字进行查询：  
  
# pstree -p `ps -e | grep server | awk '{print $1}'`  
  
# pstree -p `ps -e | grep server | awk '{print $1}'` | wc -l  
  
   
  
这里利用了管道和命令替换，  
  
关于命令替换，我也是今天才了解，就是说用``括起来的命令会优先执行，然后以其输出作为其他命令的参数，  
  
上述就是用 ps -e | grep server | awk '{print $1}' 的输出（进程号），作为 pstree -p 的参数  
  
管道和命令替换的区别是：  
  
管道：管道符号"|"左边命令的输出作为右边命令的输入  
  
命令替换：将命令替换符"``"中命令的输出作为其他命令相应位置的参数  
  
 1. cat /proc/${pid}/status  
  
2. pstree -p ${pid}  
  
3. top -p ${pid} 再按H  
  
cat /proc/`ps -e | grep server | awk '{print $1}'`/status | grep Threads | awk '{print $2}'  
  
  
一个进程同时拥有的线程数目有最大限制吗？？  
刚才查到一个，给大家分享：  
3)线程总数问题  
  
LinuxThreads将每个进程的线程最大数目定义为1024，但实际上这个数值还受到整个系统的总进程数限制，这又是由于线程其实是核心进程。  
  
在kernel 2.4.x中，采用一套全新的总进程数计算方法，使得总进程数基本上仅受限于物理内存的大小，计算公式在kernel/fork.c的fork\_init()函数中：  
        max\_threads = mempages / (THREAD\_SIZE/PAGE\_SIZE) / 8  
         
在i386上，THREAD\_SIZE=2\*PAGE\_SIZE，PAGE\_SIZE=2^12（4KB），mempages=物理内存大小/PAGE\_SIZE，对于256M的内存的机器，mempages=256\*2^20/2^12=256\*2^8，此时最大线程数为4096。  
  
但为了保证每个用户（除了root）的进程总数不至于占用一半以上物理内存，fork\_init()中继续指定：  
  
    init\_task.rlim[RLIMIT\_NPROC].rlim\_cur = max\_threads/2;  
    init\_task.rlim[RLIMIT\_NPROC].rlim\_max = max\_threads/2;  
     
这些进程数目的检查都在do\_fork()中进行，因此，对于LinuxThreads来说，线程总数同时受这三个因素的限制