守护进程：一直运行直至系统关闭的后台运行程序，该程序用于周期性执行某种任务或者等待处理某些事件。

守护进程（Daemon）是一种运行在后台的特殊进程，独立于控制终端并且周期性的执行某种任务或等待处理某些发生的事件。它从被执行的时候开始运转，直到整个系统关闭才退出。

守护进程常常在系统引导装入时启动，在系统关闭时终止

进程的运行依附于当前的终端。这个终端被称为这些进程的控制终端。当控制终端被关闭的时候，相应的进程都会自动关闭，守护进程脱离于终端并且在后台运行

在[linux](http://baike.baidu.com/view/1634.htm" \t "_blank)或者[unix](http://baike.baidu.com/view/8095.htm" \t "_blank)操作系统中，守护进程（Daemon）是一种运行在后台的特殊进程，它独立于控制终端并且周期性的执行某种任务或等待处理某些发生的事件。由于在linux中，每个系统与用户进行交流的界面称为终端，每一个从此终端开始运行的进程都会依附于这个终端，这个终端被称为这些进程的控制终端，当控制终端被关闭的时候，相应的进程都会自动关闭。但是守护进程却能突破这种限制，它脱离于终端并且在后台运行，并且它脱离终端的目的是为了避免进程在运行的过程中的信息在任何终端中显示并且进程也不会被任何终端所产生的终端信息所打断。它从被执行的时候开始运转，直到整个系统关闭才退出。

**中文名 外文名**

守护进程 Daemon

创建流程：

**进程1**

创建子进程，父进程退出

**进程2**

在子进程中创建新会话

**进程3**

改变当前目录为根目录

守护进程，也就是通常说的Daemon进程，是[Linux](http://baike.baidu.com/view/1634.htm" \t "_blank)中的后台服务进程。它是一个生存期较长的进程，通常独立于控制终端并且周期性地执行某种任务或等待处理某些发生的事件。守护进程常常在系统引导装入时启动，在系统关闭时终止。Linux系统有很多守护进程，大多数服务都是通过守护进程实现的，同时，守护进程还能完成许多系统任务，例如，作业规划进程crond、打印进程lqd等（这里的结尾字母d就是Daemon的意思）。

1、守护进程：Linux中的后台服务进程。它是一个生存期较长的进程，通常独立于控制终端并且周期性的执行某种任务或等待处理某些发生的事件，守护进程常常在系统启动时开始运行，在系统关闭时终止。

制作守护进程的步骤：

1.创建子进程，父进程退出

2.子进程创建新会话

3.子进程切换工作目录

4.修改文件权限掩码

5.关闭无用的文件描述符

创建流程：

1. 创建子进程，父进程退出

这是编写守护进程的第一步。由于守护进程是脱离控制终端的，因此，完成第一步后就会在Shell终端里造成一程序已经运行完毕的假象。之后的所有工作都在子进程中完成，而用户在Shell终端里则可以执行其他命令，从而在形式上做到了与控制终端的脱离。

在Linux中[父进程](http://baike.baidu.com/view/8507578.htm" \t "_blank)先于子进程退出会造成子进程成为孤儿进程，而每当系统发现一个孤儿进程时，就会自动由1号进程（init）收养它，这样，原先的子进程就会变成[init进程](http://baike.baidu.com/view/2965338.htm" \t "_blank)的子进程。

1. 在子进程中创建新会话

这个步骤是创建守护进程中最重要的一步，虽然它的实现非常简单，但它的意义却非常重大。在这里使用的是[系统函数](http://baike.baidu.com/view/710031.htm" \t "_blank)setsid，在具体介绍setsid之前，首先要了解两个概念：进程组和会话期

[进程组](http://baike.baidu.com/view/1545127.htm)：是一个或多个进程的集合。进程组有进程组ID来唯一标识。除了进程号（PID）之外，进程组ID也是一个进程的必备属性。每个进程组都有一个组长进程，其组长进程的进程号等于进程组ID。且该进程组ID不会因组长进程的退出而受到影响。

[会话周期](http://baike.baidu.com/view/6621448.htm)：会话期是一个或多个进程组的集合。通常，一个会话开始于用户登录，终止于用户退出，在此期间该用户运行的所有进程都属于这个会话期。

接下来就可以具体介绍setsid的相关内容：

（1）setsid函数作用：

setsid函数用于创建一个新的会话，并担任该会话组的组长。调用setsid有下面的3个作用：

1.让进程摆脱原会话的控制

2.让进程摆脱原进程组的控制

3.让进程摆脱原控制终端的控制

那么，在创建守护进程时为什么要调用setsid函数呢？由于创建守护进程的第一步调用了fork函数来创建子进程，再将[父进程](http://baike.baidu.com/view/8507578.htm" \t "_blank)退出。由于在调用了fork函数时，子进程全盘拷贝了父进程的会话期、进程组、控制终端等，虽然父进程退出了，但会话期、进程组、控制终端等并没有改变，因此，这还不是真正意义上的独立开来，而setsid函数能够使进程完全独立出来，从而摆脱其他进程的控制。

1. 改变当前目录为根目录

这一步也是必要的步骤。使用fork创建的子进程继承了父进程的当前工作目录。由于在进程运行中，当前目录所在的文件系统（如“/mnt/usb”）是不能[卸载](http://baike.baidu.com/view/386432.htm" \t "_blank)的，这对以后的使用会造成诸多的麻烦（比如系统由于某种原因要进入单用户模式）。因此，通常的做法是让"/"作为守护进程的当前工作目录，这样就可以避免上述的问题，当然，如有特殊需要，也可以把当前工作目录换成其他的路径，如/tmp。改变工作目录的常见函数是chdir。

1. 重设文件权限掩码

文件权限掩码是指屏蔽掉文件权限中的对应位。比如，有个文件权限掩码是050，它就屏蔽了[文件组](http://baike.baidu.com/view/701355.htm" \t "_blank)拥有者的可读与可执行权限。由于使用fork函数新建的子进程继承了父进程的文件权限掩码，这就给该子进程使用文件带来了诸多的麻烦。因此，把文件权限掩码设置为0，可以大大增强该守护进程的灵活性。设置文件权限掩码的函数是umask。在这里，通常的使用方法为umask(0)。

1. 关闭文件描述符

同文件权限码一样，用fork函数新建的子进程会从[父进程](http://baike.baidu.com/view/8507578.htm" \t "_blank)那里继承一些已经打开了的文件。这些被打开的文件可能永远不会被守护进程读写，但它们一样消耗系统资源，而且可能导致所在的文件系统无法卸下。

在上面的第二步之后，守护进程已经与所属的控制终端失去了联系。因此从终端输入的字符不可能达到守护进程，守护进程中用常规方法（如printf）输出的字符也不可能在终端上显示出来。所以，[文件描述符](http://baike.baidu.com/view/1303430.htm" \t "_blank)为0、1和2 的3个文件（常说的输入、输出和报错）已经失去了存在的价值，也应被关闭。通常按如下方式关闭文件描述符：

===============================

for(i=0;i<MAXFILE;i++)

close(i);

===============================

1. 守护进程退出处理

当用户需要外部停止守护进程运行时，往往会使用 kill命令停止该守护进程。所以，守护进程中需要

编码来实现kill发出的signal信号处理，达到进程的正常退出。

===============================

signal(SIGTERM, sigterm\_handler);

void sigterm\_handler(int arg)

{

\_running = 0;

}

===============================

这样，一个简单的守护进程就建立起来了。

实现守护进程的完整实例（每隔10ms在/tmp/dameon.log中写入一句话）：

=====================================================================

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<fcntl.h>

#include<sys/types.h>

#include<unistd.h>

#include<sys/wait.h>

#include<sys/stat.h>

#define MAXFILE 65535

void sigterm\_handler(int arg);

volatile sig\_atomic\_t \_running = 1;

int main()

{

pid\_t pc,pid;

int i,fd,len,flag = 1;

char \*buf="this is a Dameon\n";

len = strlen(buf);

pc = fork(); //第一步

if(pc<0){

printf("error fork\n");

exit(1);

}

else if(pc>0)

exit(0);

pid = setsid(); //第二步

if (pid < 0)

perror("setsid error");

chdir("/"); //第三步

umask(0); //第四步

for(i=0;i<MAXFILE;i++) //第五步

close(i);

signal(SIGTERM, sigterm\_handler);

while( \_running )

{

if( flag ==1

&&(fd=open("/tmp/daemon.log",O\_CREAT|O\_WRONLY|O\_APPEND,0600))<0)

{

perror("open");

flag=0;

exit(1);

}

write(fd,buf,len);

close(fd);

usleep(10\*1000); //10毫秒

}

}

void sigterm\_handler(int arg)

{

\_running = 0;

}

===============================================================================

linux守护进程列表

amd：自动安装NFS（[网络文件系统](http://baike.baidu.com/view/544339.htm)）守侯进程

apmd:高级电源治理

Arpwatch：记录日志并构建一个在LAN接口上看到的以太网地址和ip地址对数据库

Autofs：自动安装治理进程automount，与NFS相关，依靠于NIS

Bootparamd：引导参数服务器，为LAN上的[无盘工作站](http://baike.baidu.com/view/51922.htm" \t "_blank)提供引导所需的相关信息

crond：linux下的计划任务

Dhcpd：启动一个DHCP（动态IP地址分配）服务器

Gated：[网关](http://baike.baidu.com/view/807.htm" \t "_blank)路由守候进程，使用动态的OSPF[路由选择协议](http://baike.baidu.com/view/1031393.htm)

Httpd：WEB服务器

Inetd：支持多种[网络服务](http://baike.baidu.com/view/1279152.htm" \t "_blank)的核心守候程序

Innd：Usenet新闻服务器

Linuxconf：答应使用本地WEB服务器作为[用户接口](http://baike.baidu.com/view/1309799.htm" \t "_blank)来配置机器

Lpd：[打印服务器](http://baike.baidu.com/view/634855.htm" \t "_blank)

Mars-nwe：mars-nwe文件和用于Novell的打印服务器

Mcserv：Midnight命令[文件服务器](http://baike.baidu.com/view/566090.htm" \t "_blank)

named：DNS服务器

netfs：安装NFS、Samba和NetWare[网络文件系统](http://baike.baidu.com/view/544339.htm)

network：激活已配置网络接口的[脚本程序](http://baike.baidu.com/view/69991.htm" \t "_blank)

nfs：打开NFS服务

nscd：nscd(Name Switch Cache daemon)服务器，用于NIS的一个支持服务，它高速缓存用户口令和组成成员关系

portmap：RPC portmap治理器，与inetd类似，它治理基于RPC服务的连接

postgresql：一种SQL[数据库服务器](http://baike.baidu.com/view/32944.htm)

routed：路由守候进程，使用动态RIP[路由选择协议](http://baike.baidu.com/view/1031393.htm)

rstatd：一个为LAN上的其它机器收集和提供系统信息的守候程序

ruserd：[远程用户](http://baike.baidu.com/view/1541147.htm" \t "_blank)定位服务，这是一个基于RPC的服务，它提供关于当前记录到LAN上一个机器日志中的用户信息

rwalld：激活rpc.rwall服务进程，这是一项基于RPC的服务，答应用户给每个注册到LAN机器上的其他终端写消息

[rwho](http://baike.baidu.com/view/2131257.htm)d：激活rwhod服务进程，它支持LAN的rwho和ruptime服务

sendmail：[邮件服务器](http://baike.baidu.com/view/54241.htm" \t "_blank)sendmail

smb：Samba文件共享/打印服务

snmpd：本地简单网络治理候进程

squid：激活代理服务器squid

syslog：一个让系统引导时起动syslog和klogd[系统日志](http://baike.baidu.com/view/1270588.htm" \t "_blank)守候进程的脚本

xfs：X Window字型服务器，为本地和远程X服务器提供字型集

xntpd：网络时间服务器

ypbind：为NIS（网络信息系统）客户机激活ypbind服务进程

yppasswdd：NIS口令服务器

ypserv：NIS主服务器

gpm：管鼠标的

identd：AUTH服务，在提供用户信息方面与finger类似