

## 02 守护进程的作用、用途、父进程标时的特点

使进程直接成为 daemon 进程

方法一：

执行 fork 函数生成子进程，若子进程生成成功，则父进程推出，使子进程被 1#进程接管

```
pid = fork();

if(pid == -1)
{
    printf("fork error\n");
    exit(1);
}
else if(pid)
{
    exit(0);
}
```

执行 setsid 函数，使得子进程重新创建新的会话，使进程完全独立

```
if(setsid() == -1)
{
    printf("setsid error\n");
    exit(1);
}
```

Make 后执行

```
[root@vm-linux-centos7 02]# make
gcc -c -o test2.o test2.c
gcc -o test2 test2.o
[root@vm-linux-centos7 02]# ls
Makefile test2 test2.c test2.o
[root@vm-linux-centos7 02]# ./test2
[root@vm-linux-centos7 02]# 1650275
1650275

[root@vm-linux-centos7 02]# 1650275
1650275
1650275
```

Ps -el | grep test 查看

```
1 S      0 15652      1 0 80    0 - 1053 hrttime ?      00:00:00 test2
```

方法二：

调用库函数 daemon 函数生成守护进程

```
daemon(0, 1);
```

第一个参数为 0 设置守护进程运行目录为根目录，第二个参数为零设置输入输出为/dev/null

重新 make 后达到同样的效果

```
[root@vm-linux-centos7 02]# make
gcc -c -o test2.o test2.c
gcc -o test2 test2.o
[root@vm-linux-centos7 02]# ./test2
[root@vm-linux-centos7 02]# 1650275
1650275
1650275
1650275
1650275
1650275
1650275
1650275
1650275
1650275
```

在不同终端使用 `ps -ef | grep test` 查看 test 相关进程，第二列为进程标识，第三列为父进程标识

```
[root@vm-linux-centos7 02]# ps -ef | grep test
root      15729      1  0 12:09 ?        00:00:00 ./test2
root      15731  15146  0 12:09 pts/0    00:00:00 grep --color=auto test
[root@vm-linux-centos7 02]#
```

退出终端后 test2 仍在运行，再次登陆后不再打印信息

若要始终可以查看到输出结果，可将结果输出到文件，printf 后需要刷新输出缓存

```
[root@vm-linux-centos7 ~]# ./test2 > output.txt
```

可随时使用 `tail -f [文件名]` 循环读取文件中的结果