

# linux中使用pthread\_kill函数测试线程是否存活

 [blog.csdn.net/echoisland/article/details/6398081](http://blog.csdn.net/echoisland/article/details/6398081)

☰ 分类：

**pthread\_kill：**

别被名字吓到，pthread\_kill可不是kill，而是向线程发送signal。还记得signal吗，大部分signal的默认动作是终止进程的运行，所以，我们才要用signal()去抓信号并加上处理函数。

```
int pthread_kill(pthread_t thread, int sig);
```

向指定ID的线程发送sig信号，如果线程代码内不做处理，则按照信号默认的行为影响整个进程，也就是说，如果你给一个线程发送了SIGQUIT，但线程却没有实现signal处理函数，则整个进程退出。

pthread\_kill(threadid, SIGKILL)也一样，杀死整个进程。

如果要获得正确的行为，就需要在线程内实现signal(SIGKILL,sig\_handler)了。

所以，如果int sig的参数不是0，那一定要清楚到底要干什么，而且一定要实现线程的信号处理函数，否则，就会影响整个进程。

OK，如果int sig是0呢，这是一个保留信号，一个作用是用来判断线程是不是还活着。

我们来看一下pthread\_kill的返回值：

成功:0

线程不存在：ESRCH

信号不合法：EINVAL

所以，pthread\_kill(threadid,0)就很有用啦。

```
int kill_rc = pthread_kill(thread_id,0);
```

```
if(kill_rc == ESRCH)
```

```
printf("the specified thread did not exists or already quit/n");
```

```
else if(kill_rc == EINVAL)
```

```
printf("signal is invalid/n");
```

```
else
```

```
printf("the specified thread is alive/n");
```

上述的代码就可以判断线程是不是还活着了。

**版权声明：**转载时请以超链接形式标明文章原始出处和作者信息及**本声明**

<http://xufish.blogbus.com/logs/40545082.html>

使用pthread\_kill函数检测一个线程是否还活着的程序，在linux环境下gcc编译通过，现将代码贴在下面：

```
/****** pthread_kill.c *****/
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <pthread.h>
```

```
#include <errno.h>
```

```

void *func1()/*1秒钟之后退出*/
{
    sleep(1);
    printf("线程1 (ID : 0x%x) 退出。/n",(unsigned int)pthread_self());
    pthread_exit((void *)0);
}

void *func2()/*5秒钟之后退出*/
{
    sleep(5);
    printf("线程2 (ID : 0x%x) 退出。/n",(unsigned int)pthread_self());
    pthread_exit((void *)0);
}

void test_pthread(pthread_t tid) /*pthread_kill的返回值：成功 (0) 线程不存在 (ESRCH) 信号不合法 (EINVAL) */
{
    int pthread_kill_err;
    pthread_kill_err = pthread_kill(tid,0);

    if(pthread_kill_err == ESRCH)
        printf("ID为0x%x的线程不存在或者已经退出。/n",(unsigned int)tid);
    else if(pthread_kill_err == EINVAL)
        printf("发送信号非法。/n");
    else
        printf("ID为0x%x的线程目前仍然存活。/n",(unsigned int)tid);
}

int main()
{
    int ret;
    pthread_t tid1,tid2;

    pthread_create(&tid1,NULL,func1,NULL);
    pthread_create(&tid2,NULL,func2,NULL);

    sleep(3);/*创建两个进程3秒钟之后，分别测试一下它们是否还活着*/

    test_pthread(tid1);/*测试ID为tid1的线程是否存在*/
    test_pthread(tid2);/*测试ID为tid2的线程是否存在*/

    exit(0);
}

```

编译：gcc -o pthread\_kill -lpthread pthread\_kill.c

运行：./pthread\_kill

////////// 运行结果 //////////

线程1 (ID : 0xb7e95b90) 退出。

ID为0xb7e95b90的线程不存在或者已经退出。

ID为0xb7694b90的线程目前仍然存活。