[Linux进程间通信——使用消息队列](http://blog.csdn.net/ljianhui/article/details/10287879)

一、什么是消息队列

消息队列提供了一种从一个进程向另一个进程发送一个数据块的方法。每个数据块都被认为含有一个类型，接收进程可以独立地接收含有不同类型的数据结构。我们可以通过发送消息来避免命名管道的同步和阻塞问题。但是消息队列与命名管道一样，每个数据块都有一个最大长度的限制。

Linux用宏MSGMAX和MSGMNB来限制一条消息的最大长度和一个队列的最大长度。

二、在Linux中使用消息队列

1、msgget函数

该函数用来创建和访问一个消息队列。它的原型为：

1. **int** msgget(key\_t key, **int** msgflg);

与其他的IPC机制一样，程序必须提供一个键来命名某个特定的消息队列。msgflg是一个权限标志，表示消息队列的访问权限，它与文件的访问权限一样。msgflg可以与IPC\_CREAT做或操作，表示当key所命名的消息队列不存在时创建一个消息队列，如果key所命名的消息队列存在时，IPC\_CREAT标志会被忽略，而只返回一个标识符。

它返回一个以key命名的消息队列的标识符（非零整数），失败时返回-1.

2、msgsnd函数

该函数用来把消息添加到消息队列中。它的原型为：

1. **int** msgsend(**int** msgid, **const** **void** \*msg\_ptr, **size\_t** msg\_sz, **int** msgflg);

msgid是由msgget函数返回的消息队列标识符。

msg\_ptr是一个指向准备发送消息的指针，但是消息的数据结构却有一定的要求，指针msg\_ptr所指向的消息结构一定要是以一个长整型成员变量开始的结构体，接收函数将用这个成员来确定消息的类型。所以消息结构要定义成这样：

1. **struct** my\_message{
2. **long** **int** message\_type;
3. /\* The data you wish to transfer\*/
4. };

msg\_sz是msg\_ptr指向的消息的长度，注意是消息的长度，而不是整个结构体的长度，也就是说msg\_sz是不包括长整型消息类型成员变量的长度。

msgflg用于控制当前消息队列满或队列消息到达系统范围的限制时将要发生的事情。

如果调用成功，消息数据的一分副本将被放到消息队列中，并返回0，失败时返回-1.

3、msgrcv函数

该函数用来从一个消息队列获取消息，它的原型为

1. **int** msgrcv(**int** msgid, **void** \*msg\_ptr, **size\_t** msg\_st, **long** **int** msgtype, **int** msgflg);

msgid, msg\_ptr, msg\_st的作用也函数msgsnd函数的一样。

msgtype可以实现一种简单的接收优先级。如果msgtype为0，就获取队列中的第一个消息。如果它的值大于零，将获取具有相同消息类型的第一个信息。如果它小于零，就获取类型等于或小于msgtype的绝对值的第一个消息。

msgflg用于控制当队列中没有相应类型的消息可以接收时将发生的事情。

调用成功时，该函数返回放到接收缓存区中的字节数，消息被复制到由msg\_ptr指向的用户分配的缓存区中，然后删除消息队列中的对应消息。失败时返回-1.

4、msgctl函数

该函数用来控制消息队列，它与共享内存的shmctl函数相似，它的原型为：

1. **int** msgctl(**int** msgid, **int** command, **struct** msgid\_ds \*buf);

command是将要采取的动作，它可以取3个值，

  IPC\_STAT：把msgid\_ds结构中的数据设置为消息队列的当前关联值，即用消息队列的当前关联值覆盖msgid\_ds的值。

  IPC\_SET：如果进程有足够的权限，就把消息列队的当前关联值设置为msgid\_ds结构中给出的值

  IPC\_RMID：删除消息队列

buf是指向msgid\_ds结构的指针，它指向消息队列模式和访问权限的结构。msgid\_ds结构至少包括以下成员：

1. **struct** msgid\_ds
2. {
3. uid\_t shm\_perm.uid;
4. uid\_t shm\_perm.gid;
5. mode\_t shm\_perm.mode;
6. };

成功时返回0，失败时返回-1.

三、消息队列与命名管道的比较

消息队列跟命名管道有不少的相同之处，通过与命名管道一样，消息队列进行通信的进程可以是不相关的进程，同时它们都是通过发送和接收的方式来传递数据的。在命名管道中，发送数据用write，接收数据用read，则在消息队列中，发送数据用msgsnd，接收数据用msgrcv。而且它们对每个数据都有一个最大长度的限制。

与命名管道相比，消息队列的优势在于：

1、消息队列也可以独立于发送和接收进程而存在，从而消除了在同步命名管道的打开和关闭时可能产生的困难。

2、同时通过发送消息还可以避免命名管道的同步和阻塞问题，不需要由进程自己来提供同步方法。

3、接收程序可以通过消息类型有选择地接收数据，而不是像命名管道中那样，只能默认地接收。