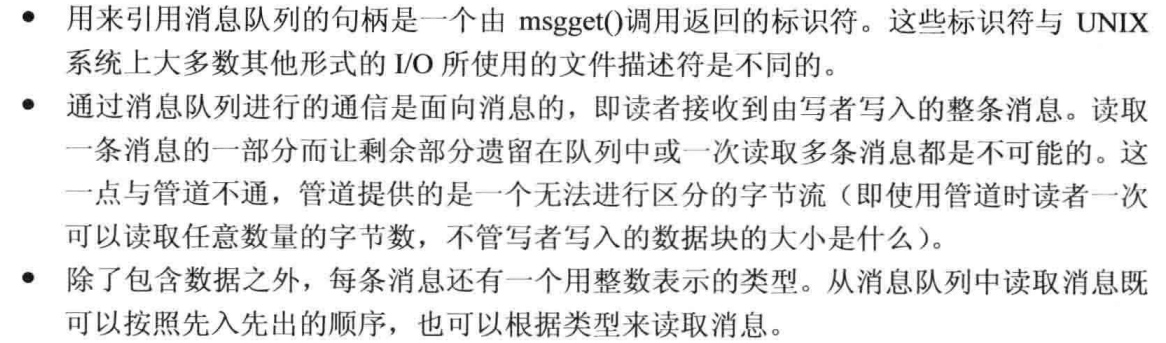
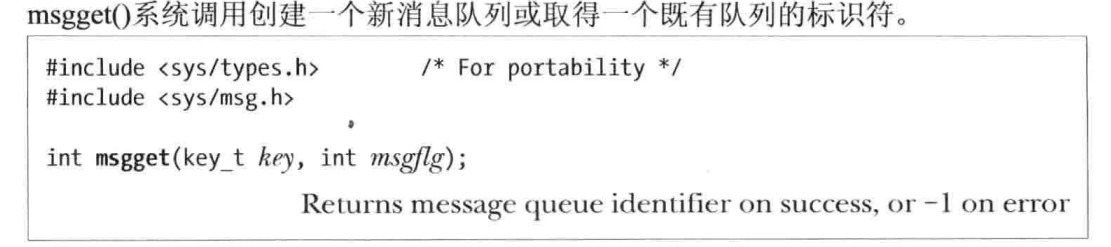
第四十六章

1. System V IPC消息队列：
2. 消息队列以消息的形式交换数据，与FIFO有类似的地方，但是也有区别：

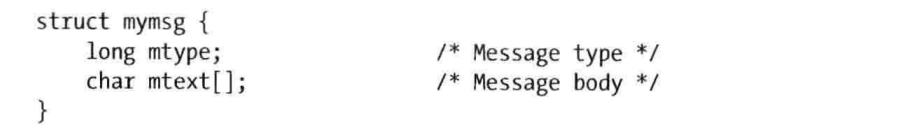


1. 创建或打开一个消息队列：

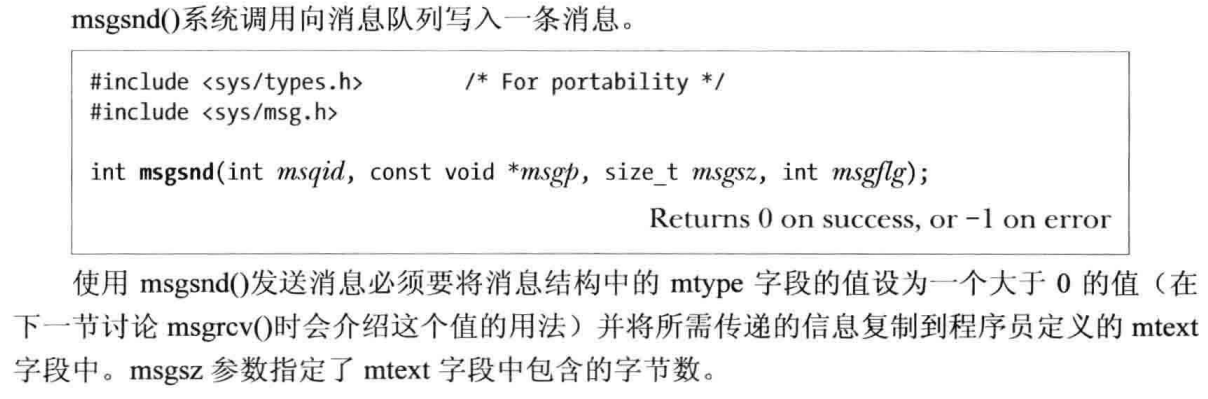


1. 交换消息：

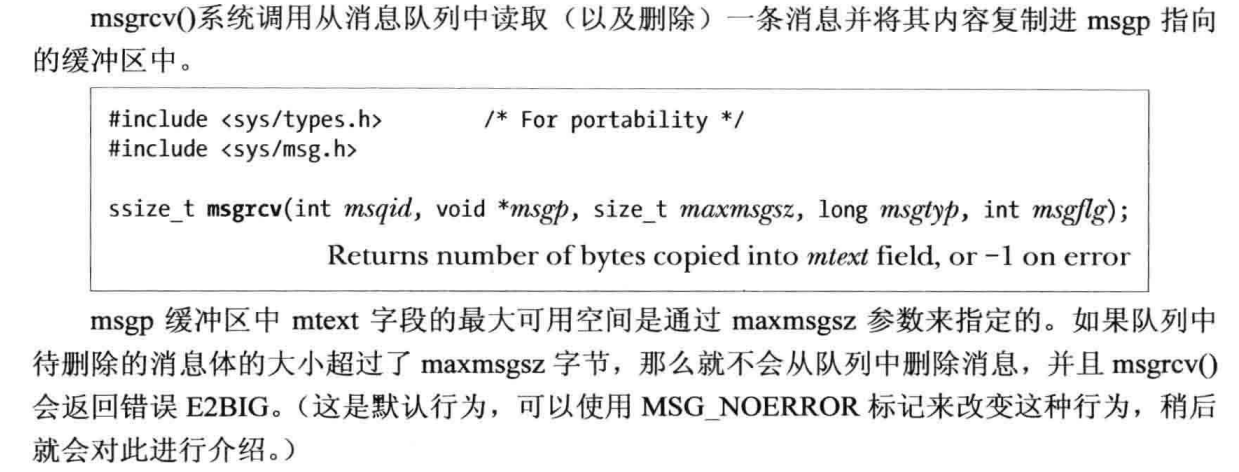
交换消息的第二个参数是由程序员定义的结构的指针，常规形式如下：



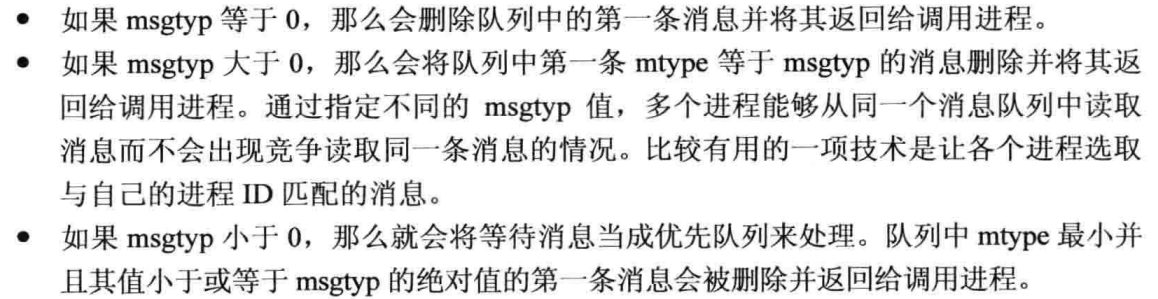
发送消息：



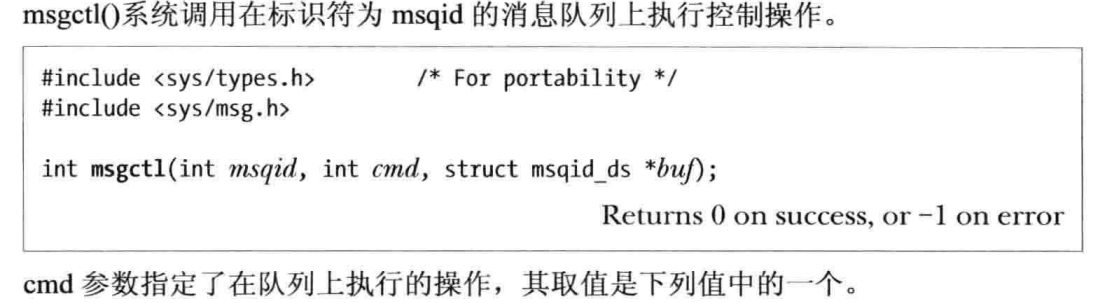
接收消息：



根据msgtyp的不同，接收有不同的语义：



1. 消息队列控制操作：



注意，具体cmd的取值查看文档或书本。常用的取值有：IPC\_RMID，IPC\_STAT，IPC\_SET，分别用来删除，获取以及修改msqid\_ds结构体。

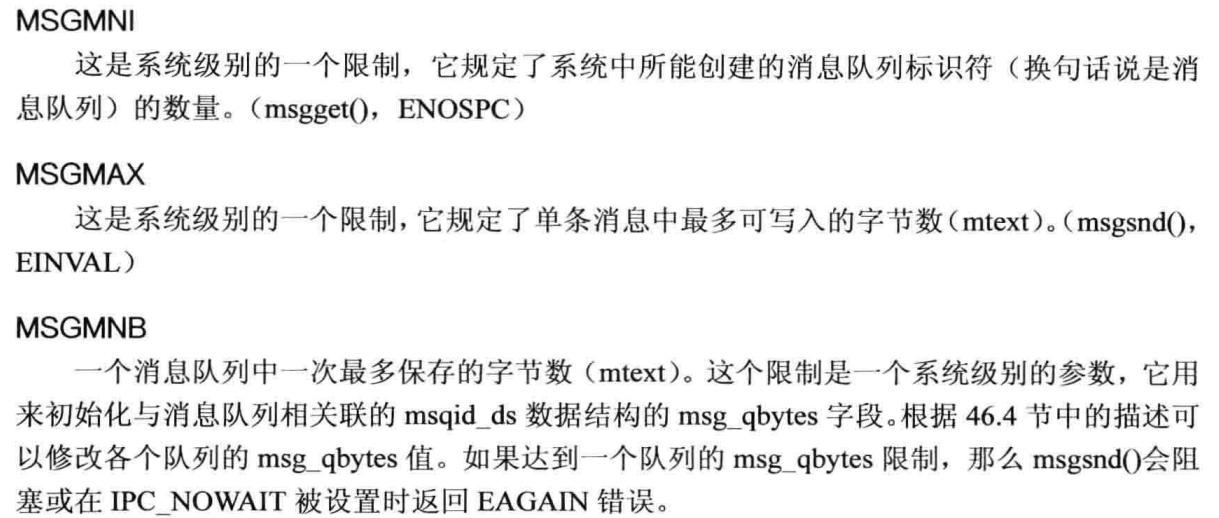
1. 消息队列关联数据结构：



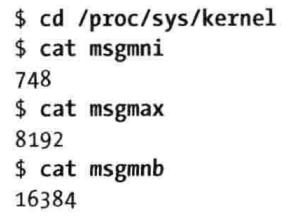
具体内容和注意事项查看书本或者手册。

1. 消息队列中的限制：

Linux上有的限制：



在Linux上可以通过/proc文件系统中的文件查看这些限制：



1. 显示系统中的消息队列：

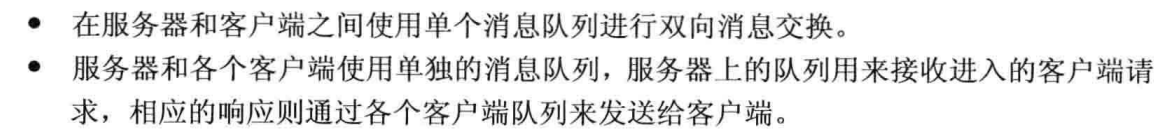
除了之前介绍的/proc文件外，可以使用ctl系统调用。

主要的原理是：

MSG\_INFO返回的是表示消息队列对象的数据结构的entries数组中最大项的下标。而MSG\_STAT可以根据下标获取消息队列。注意判断不存在和没有权限读的错误。

1. 使用消息队列实现客户端和服务器应用程序：

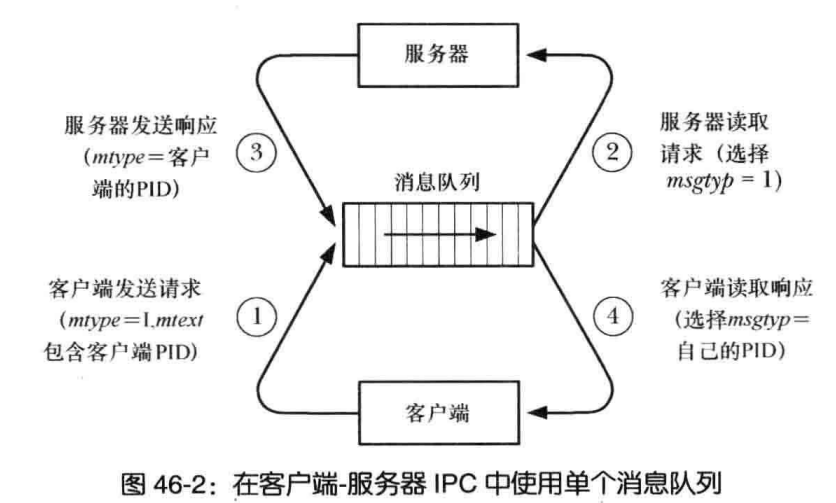
使用消息队列进行通信的方式有两种：



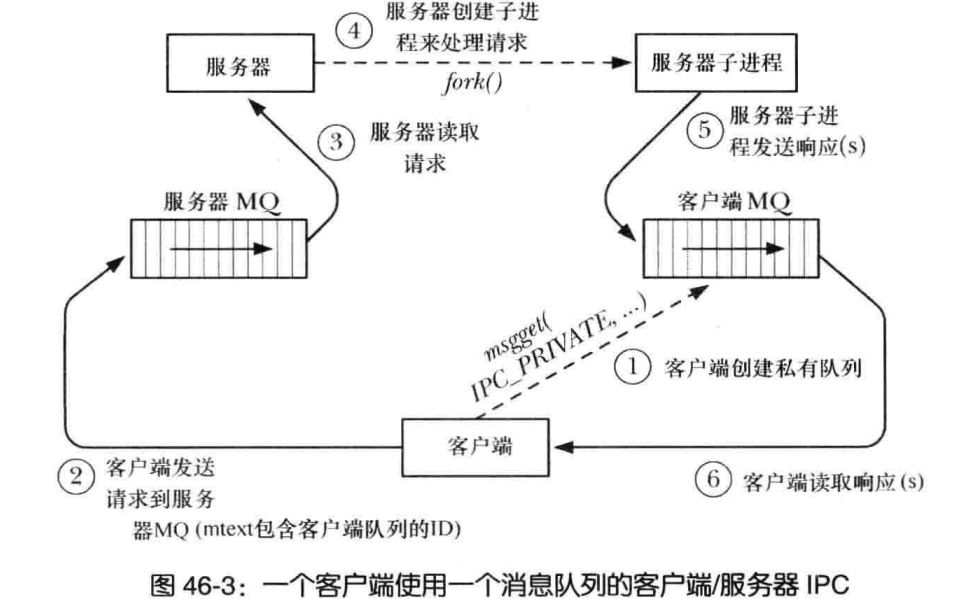
对于第一种情况，需要注意的点：

1. 需要使用类型来区别各个进程需要读取的消息。
2. 因为消息队列数量有限所以会导致一系列问题：死锁和故意不接受消息的恶意客户端。

模型如下：



第二种情况模型如下：



一个非常有意思的点，消息队列的消息除了包含类型，消息，甚至可以包含别的字段。只要正确表示出它的大小。如结构体常用到的offsetof这个宏。

1. System V消息队列的缺点：
2. 消息队列使用标识符而不是文件标识符，增加程序难度。
3. 使用键而不是文件。
4. 消息队列是无连接的。无法得知何时删除消息队列。
5. 存在各种限制。

