# 实验四 Linux进程的管道通信

**1、概念**

①是一个用于连接一个读进程·和一个写进程的pipe文件；

②按照FIFO（**先进先出**）方式单向传递消息，只允许在建立者及其子进程之间适用

③互斥和同步：

互斥：一个进程正在对pipe进行读写时，另一进程必须等待

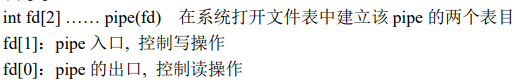
同步：

发送进程：写pipe时，若pipe文件长度已经到4096字节，仍有部分信息没有写入，则进程进入睡眠状态；当读进程收走了全部信息，被唤醒，将余下信息送入pipe中

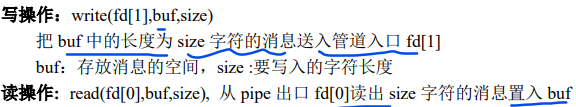
接收进程：读pipe时，若pipe为空，则进入等待状态。一旦有发送进程执行完写操作，被唤醒。

**2、涉及到的系统调用**

① pipe()



② write()和read()



③ sleep(n)

进程调用后自动阻塞n秒，n秒后自动唤醒，进入就绪状态。

④ lockf(int files, int function , long size )

其中：files文件描述符，

function是锁定和解锁：1表示锁定；0表示解锁

size：锁定和解锁的字节数；0代表从文件的当前位置到文件尾

**3、实验涉及到的头文件**

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<string.h>

#include<sys/msg.h>

#include<sys/wait.h>

#include<sys/msg.h>

# 实验五 Linux软中断信号处理

**1、概念**

① 进程之间可以通过调用kill函数发送软中断信号

② 必须先执行void(\*signal(int sig,void (\*func)(int)))(int)，在执行int kill(pid\_t pid , int sig)

**2、相关系统调用**

