# 信号

1 注册的信号处理函数在fork后，父进程和子进程共享相同的信号处理函数。

2 当造成信号的事件发生时，（内核）为进程产生一个信号（或向一个进程发送一个信号）。事件可以是：

1. **硬件异常**：除0，访问无效内存
2. **软件条件**：alarm超时，abort函数调用，子进程退出
3. **终端产生的信号**：CTRL+C产生SIGINT中断信号
4. **Kill或raise函数发送**

当一个信号产生时，内核通常在进程表（即**接收该信号的进程**）中以某种形式设置一个标志。当**在进程表中设置好该标志时，我们说向进程递送了一个信号**。

3 在信号产生和递送之间的时间间隔内，称信号是**未决的**。

4 **每个进程都有一个信号屏蔽字**，它规定了当前要阻塞递送到该进程的信号集。进程可以设置进程屏蔽字来阻塞某个信号的递交。如果为进程产生了一个阻塞的信号，而且对该信号是**系统默认动作**或**捕捉该信号**，则为该进程将此**信号保持为未决状态**，直到该进程对该信号解除了阻塞或者将对此信号的动作更改为忽略。

5 如果在进程解除对某个信号的阻塞之前，这种信号发生了多次，那么在解除阻塞后，内核一般只递送一次，这是因为**大多数unix系统不对信号排队**。

6 对于Linux操作系统，sigaction函数默认是不重启动被中断的系统调用。

# 管道

1 管道pipe是半双工的，且只能工作在**具有亲缘关系的进程**（即父子进程或兄弟进程）之间。

2 FIFO（也叫命名管道）是半双工的，但能够工作在任意进程之间。

3 linux shell中的管道符号是**|**，管道**|**前面的命令的标准输出将通过管道传递给管道后面的命令作为后面命令的标准输入。

4 当open一个FIFO时，非阻塞标志会产生以下影响：

* 当FIFO文件为阻塞模式（默认）：

1. 只读open要阻塞到某个其他进程为写而打开这个FIFO为止。
2. 只写open要阻塞到某个其他进程为读而打开它为止。

* 当FIFO文件设置为非阻塞模式时：

1. 只读open立即返回
2. 如果没有进程为读而打开一个FIFO，只写open将返回-1，并设置errno为ENXIO。

5 **FIFO管道的用途**：

1. Shell命令使用FIFO将数据从一条管道传递到另一条，而无须创建中间临时文件。
2. 客户-服务器模式中，FIFO用作汇聚点，在客户进程和服务器进程之间传递数据。

# XSI IPC：消息队列、信号量、共享内存

1 XSI IPC存在以下问题：

1. **IPC对象没有引用计数**。如果进程创建了一个消息队列，并且在该消息队列中放入几条信息，然后终止，那么该消息队列及其内容不会被删除。它们会一直留在系统中直到发生下列动作之一：

* **由某个进程调用msgrcv读消息**：当消息队列中的消息都被读走时，消息队列被删除
* **由某个进程调用msgctl删除消息队列**
* 在shell中执行ipcrm命令删除消息队列
* 系统重启

2）**IPC结构在文件系统中没有名字**：这意味着不能对其调用open、read、write等操作。**内核中新增了系统调用（msgget、semop、shmat等）**来提供对ipc对象的操作。另外，也不能使用ls命令查看ipc对象，于是又增加了两个shell命令：**ipcs**、**ipcrm**

2 **消息队列**：消息队列是消息的链接表，**存储在内核**中，由消息队列标识符标识。

* **发送的消息总是放在消息队列尾部**。
* 接收者和发送者之间可以协商消息类型，这样接收者在接收时可以接收指定消息类型的消息，从而避开了先进先出的次序。
* 因为每个消息队列没有维护引用计数，所以在**队列被删除以后（即删除队列会立即生效）**，仍在使用这一队列的进程在下次对队列进行操作时会出错返回。

3 **信号量**：就是一个计数器，用于为多个进程提供对共享数据的访问。信号量是在内核中实现的。

* 对于XSI信号量，如果在进程终止时，它占用了经由信号量分配的资源，并且没有为信号量操作指定SEM\_UNDO标志，那么当该进程终止时，内核不会回收该信号量，导致资源泄露。
* POSIX信号量解决了XSI信号量存在的几个问题，使用更简单，如果要使用信号量，那么应该选择使用POSIX信号量。POSIX信号量有两种形式：

1. **匿名信号量**：位于内存中，只能用于一个进程的不同线程间或具有亲缘关系的进程间访问共享资源。
2. **命名信号量**：可以通过名字访问，因此可以被任何已知它们名字的进程（线程）使用。
3. **如果进程退出前没有调用sem\_close释放信号量，那么内核将自动关闭任何打开的信号量。**
4. **POSIX信号量有引用计数**：当调用sem\_unlink删除信号量的名字时，如果没有打开的信号量引用，那么该信号量会被销毁。否则，销毁将延迟到最后一个打开的引用关闭。

4 **共享内存**：共享内存没有数据的复制操作，是最快的一种IPC。但是由于共享内存被多个进程共享，因此必须提供同步机制。目前有以下两种类型的共享内存：

* **文件映射**：利用mmap，可以使用**/dev/zero作为映射文件**
* **XSI共享内存**：没有相关的文件，用的是内存的匿名段。