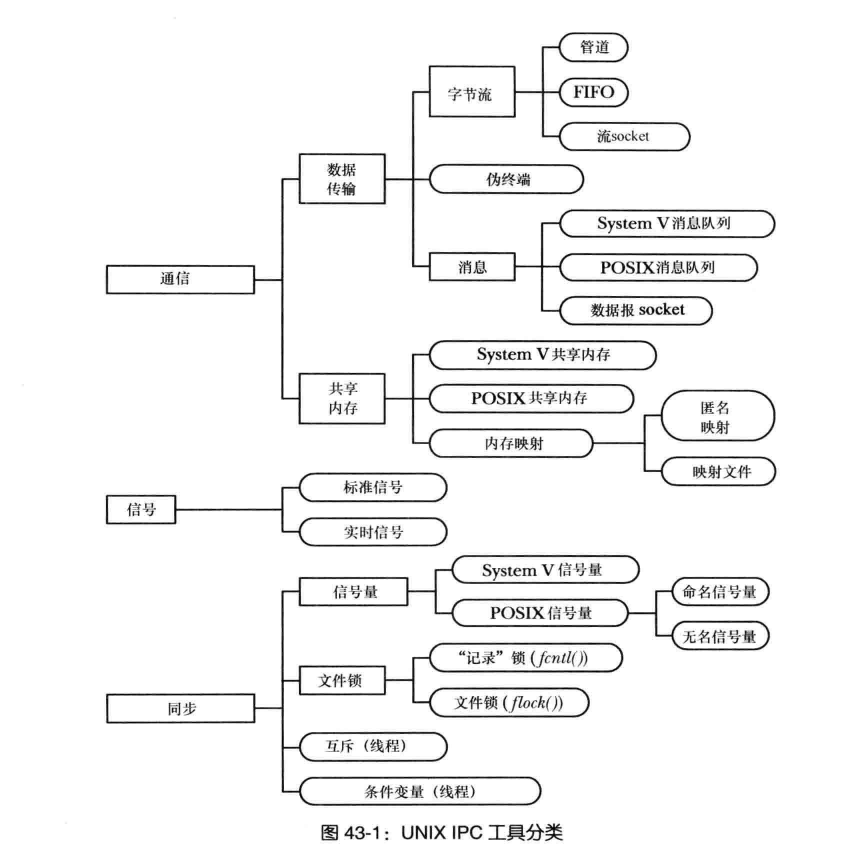
IPC大汇总

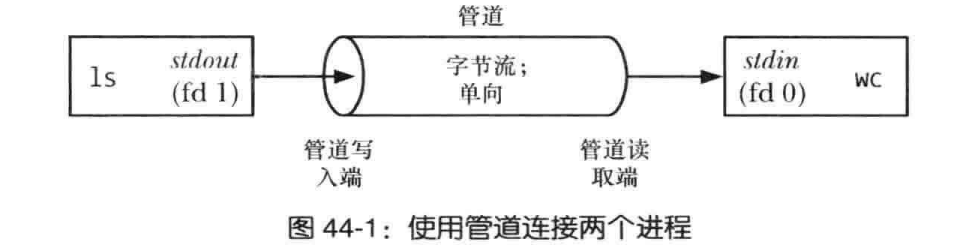
大纲：本文不对各种IPC的具体用法进行详解，而是列举各个IPC的特征，并从IPC工具的分类，特征，识别对象，如何引用，权限，及持久性来进行综述。IPC的分类如下图：



1. 管道：

管道的图示：

在shell上执行命令：ls | wc -l 就用到了管道，图示：



管道的特征：

管道是内核内存中维护的缓冲器。

1. 一个管道是一个字节流。
2. 可以从管道中读取数据。
3. 管道是单向的。
4. 可以确保写入不超过PIPE\_BUF字节的操作是原子的。
5. 管道的容量是有限的。

属于IPC工具里的通信工具，字节流传输，没有名称用于识别对象，在程序中使用文件描述符引用，权限方面仅允许“相关的”进程进行访问，即通过fork进行相关联的进程。具有进程持久性。

1. FIFO：

又称命名管道，语义上和管道类似，但是在文件系统中拥有名称。

注意，FIFO的数据持久性和名称持久性是不一样的。在权限上，FIFO的权限由mode参数和umask决定的。

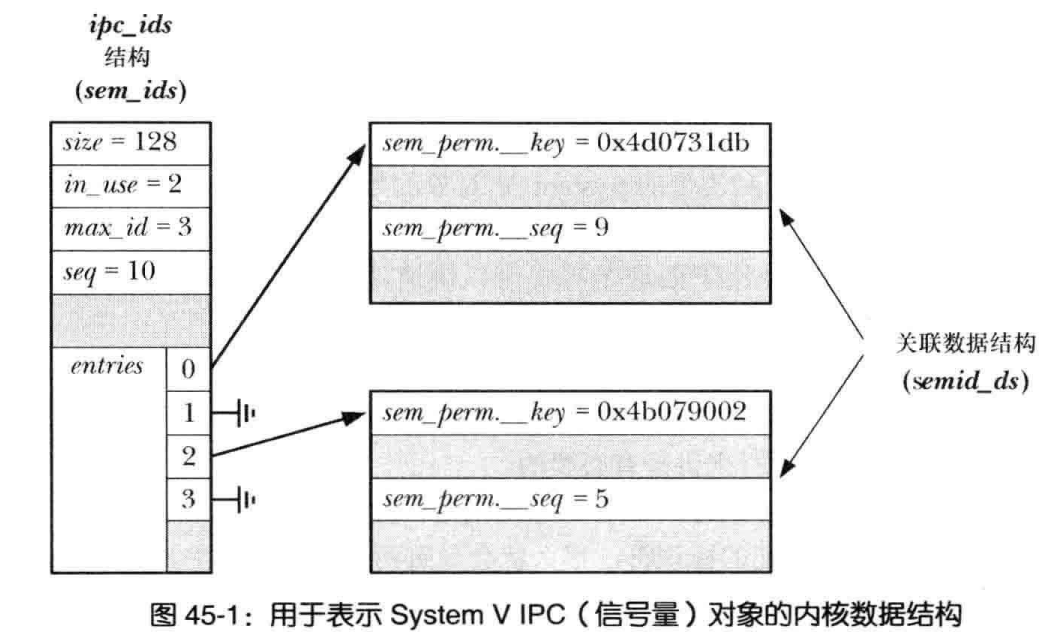
与别的文件打开不同，FIFO的非阻塞打开和非阻塞读写具有自己的语义，具体查看书本。

1. 接下来讲述System V IPC：

包括三种不同的类别，分别是消息队列，信号量和共享内存。

共同点：

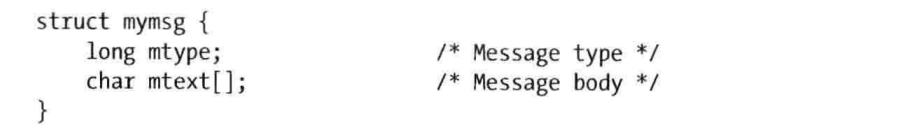
1. 用于识别对象是通过key键。返回的引用对象的句柄是标识符。
2. 权限方面是通过权限掩码判别。
3. 创建key的方法类似：IPC\_PRIVATE和ftok。
4. 内部维护的结构如图：



1. System V 消息队列：

System V 消息队列是以消息的形式交换数据，此外每条消息还包含一个整数表示类型。

消息的常规形式如下：



注意，也不一定要按照这种格式，可以多个字段，但要使用offsetof宏对长度加以判断。

读取消息的时候，可以根据消息类型获取消息。具有系统持久性。

1. System V 信号量：

System V 信号量不是用来传输数据的，而是用来同步进程的动作的。

System V 信号量的分配是以被称为信号量集的组为单位进行的。

一个信号量是内核维护的一个整数，其值被限制为大于或等于0。

System V 信号量具有信号量撤销的特点。

1. System V 共享内存：

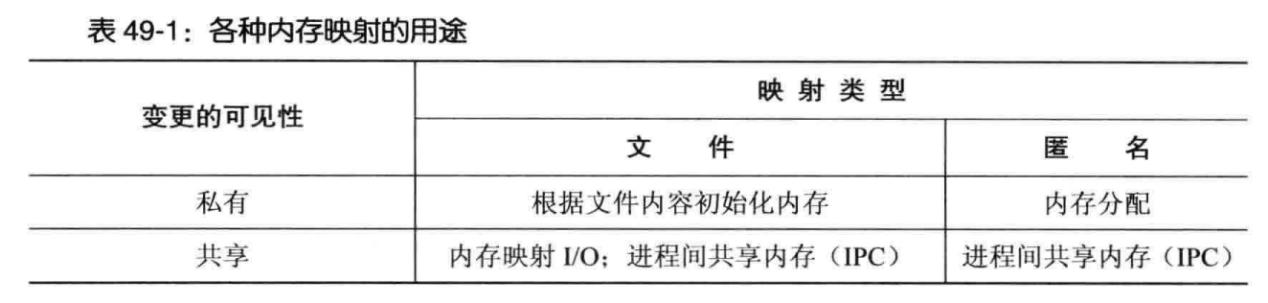
共享内存段会成为一个进程用户空间的一部分，因此这种IPC机制无需内核介入，正因为如此，需要通过某种方法同步访问。

一般使用共享内存段需要使用信号量进行同步。

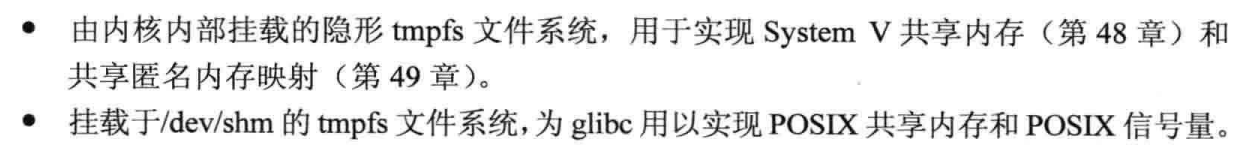
1. 内存映射：

根据CSAPP的定义：Linux通过将一个虚拟内存区域与一个磁盘上的对象关联起来，以初始化这个虚拟内存区域的内容，这个过程称为内存映射。映射分为文件映射和匿名映射。

根据不同的映射类型，用途不一样，能够进行IPC的是共享的映射。



共享文件映射内容上的变更会反应到底层文件上，所以它的持久性是文件系统持久性。



1. POSIX IPC的总体概况：

POSIX IPC机制开发出来的目的是弥补System V IPC的不足。

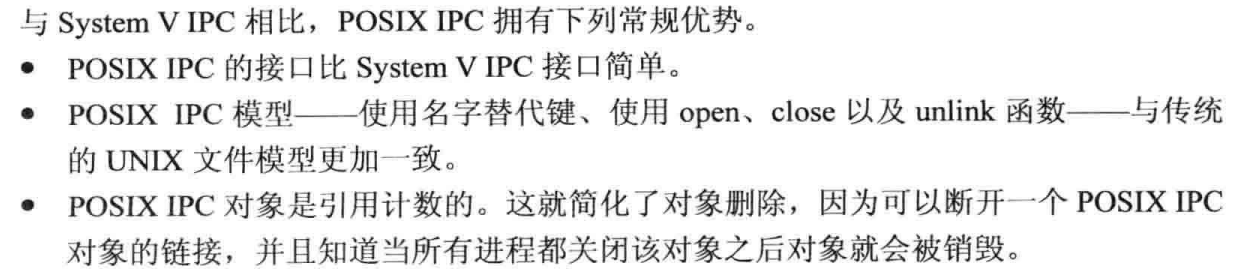
权限机制与文件权限一样。

与System V 相比，内核会维护打开对象的引用计数。

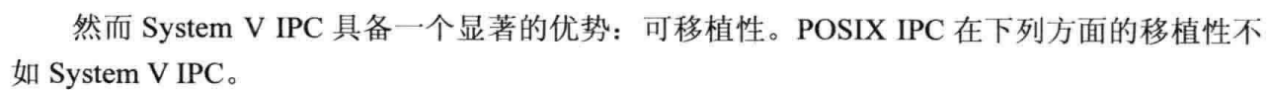
具有内核持久性。

通过命令行ls，rm列出和删除对象。

POSIX优势：



System V优势：



1. POSIX 消息队列：

与管道不同，当没有写入者的时候，读者依旧会等待，不会看到文件结束。

以消息为单位交换数据。

在Linux上实现为虚拟文件系统。

POSIX的优势：

1. 接口与传统文件模型更加一致。
2. 引用计数。
3. 异步通知。
4. 可以使用select等。

System V消息队列：

1. 兼容性好。
2. 接收消息更加灵活。
3. POSIX 信号量：

持久性：

对于命名信号量来说，由于是依据POSIX共享内存实现的，所以是内核持久性。而对于无名信号量来说，持久性需要分情况。线程共享的信号量是进程持久性，而进程共享信号量与与之相关的共享内存持久性有关。一般是内核持久，而匿名共享内存是进程持久性。

可访问性：

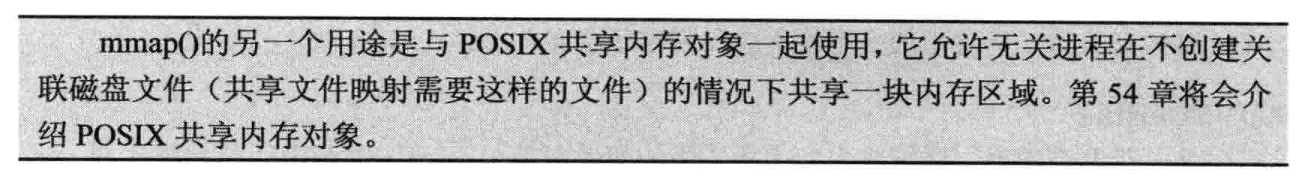
对于命名信号量，取决于权限掩码。而线程共享的未命名信号量是相关线程，进程共享的则依据共享内存的可访问性。

1. POSIX 共享内存：

POSIX共享内存的持久性是内核持久性，可访问性取决于权限掩码。

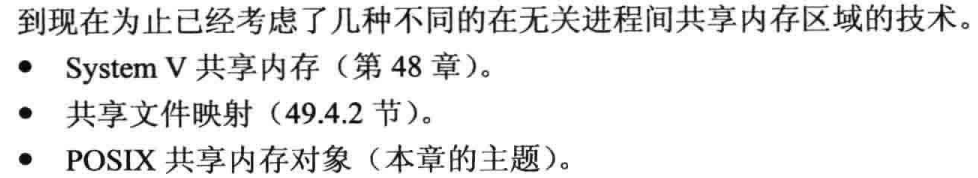
实现方式：简单来说就是采用文件系统标识共享内存对象，将共享对象名创建为文件系统上的文件。Linux采用的是挂载在/dev/shm上的专用tmpfs文件系统。

实际上，这里所做的事情是使用shm\_open代替open调用，其中的差别是前者无需在一个基于磁盘的文件系统上创建文件。

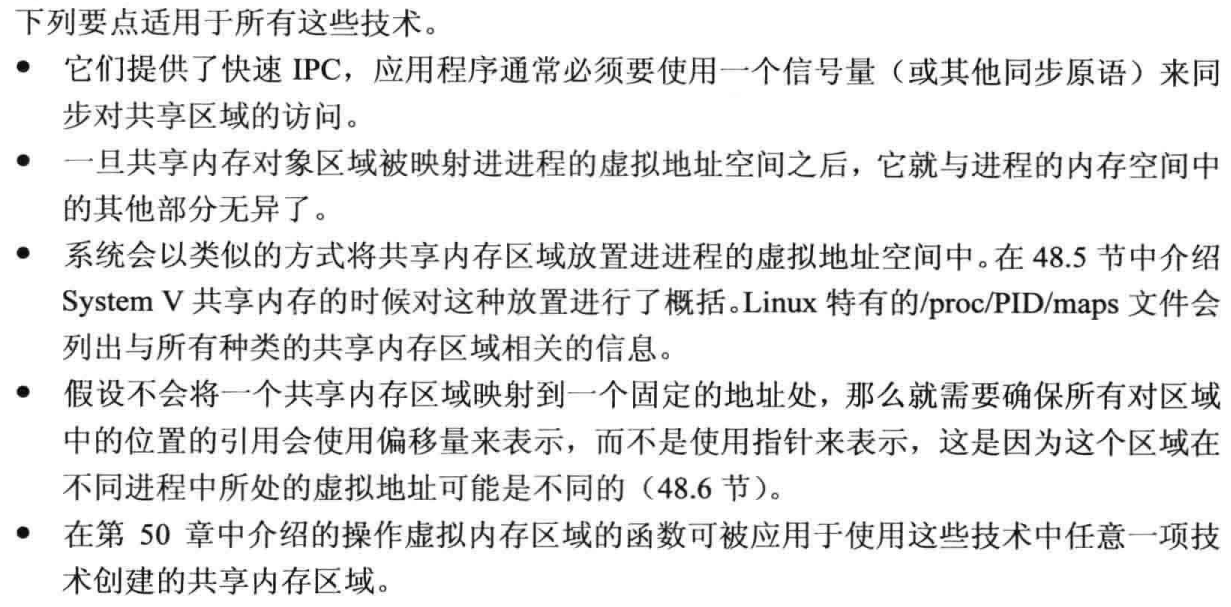


1. 暂时性总结：

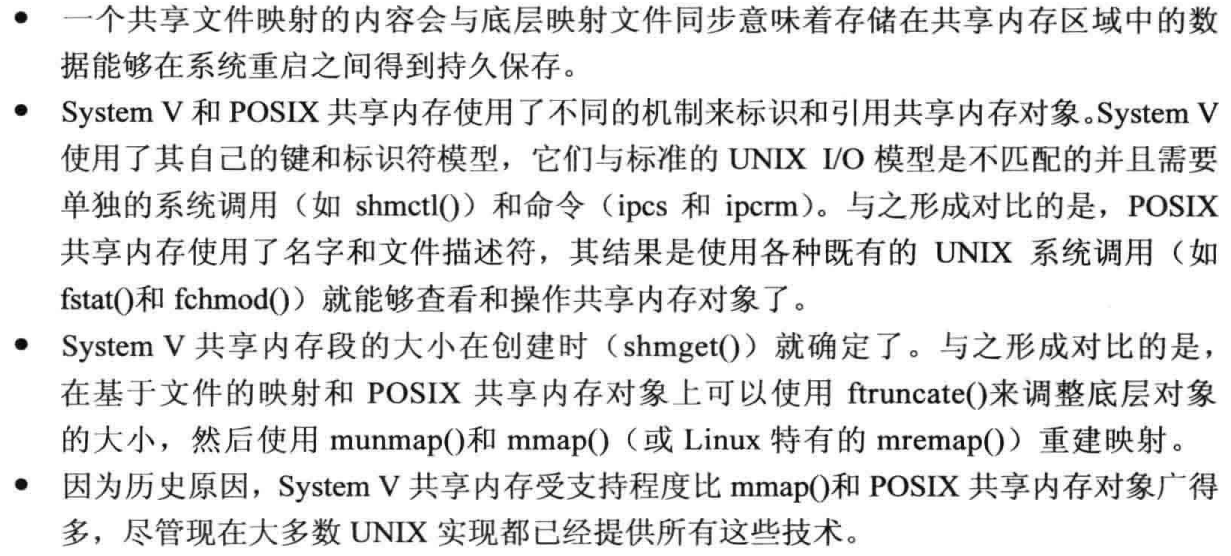
至此，IPC有三种无关进程共享内存区域的技术：



共同点：



差异：



如何选择：

