



Nginx

— 进程机制和通信剖析

主讲martin老师: 2390560487





课程提纲

高并发概述

单机服务框架

Nginx进程间通信概述

Nginx进程间通信实战



课程提纲

[“微信指数”上线 你在8亿用户中的影响力一搜即知 TechWeb](#)

2017年3月24日 - 中韩长沙一战的当晚,微信悄然上线了“微信指数”能+与百度指数、新浪微博指数类似,这个基于**微信8亿多用户**的微信指数,能够实时查询微信热点,甚至...
www.techweb.com.cn/int... - 百度快照

[微信8亿用户价值被严重高估:我看公众号的末路和头条的未来](#)

2016年10月2日 - 依我看,微信平台、尤其是公众号平台的价值被严重高估了:微信和公众号值不了腾讯的半壁江山。有人拿出**微信**是国内第一大社交软件平台,用户**8亿**;公...
www.360doc.com/content... - 百度快照

[9亿用户的QQ、8亿用户的微信都免费,马化腾凭啥3个月赚100亿?](#)

2016年5月25日 - (原标题:9亿用户的QQ、8亿用户的微信都免费,腾讯凭啥3个月赚100亿?)日前,腾讯公司发布了2016年第一季度业绩收入320亿元,非通用会计准则下,...
money.163.com/16/0525/... - 百度快照

[腾讯微信活跃的用户就有8亿多了,这个公司的用户量居然是腾讯的...](#)

[微信红包的并发机制? - 互联网 - 知乎](#)

可是却碰到红包服务器繁忙的问题,请问腾讯的红包机制最多支持多少**并发**数据的实时...**微信**包会给市场带来多大冲击? 22 个回答 如何评价微信的「**新年红包**」功能? 体验...
<https://www.zhihu.com/question...> - 百度快照

[解密春晚微信红包十亿级并发压力 - CSDN博客](#)

2016年9月2日 - 对于春晚**微信红包**这个达到十亿级别的请求来说,如果使用传统的分布式事处理方式,**微信红包**这个复杂的过程会放大**并发**压力,并且足以导致系统异常。例如...
blog.csdn.net/qj_35446... - 百度快照

[揭秘微信红包:架构、抢红包算法、高并发和降级方案](#)



2016年4月10日 - 揭秘**微信红包**:架构、抢**红包**算法、高**并发**和降级案 <http://www.infoq.com/cn/articles/2016-hongbao-weixin-archtec>
微信红包 高**并发** 降级 阅读...
geek.csdn.net/news/det... - 百度快照

[微信高并发资金交易系统设计方案——百亿红包背后的技... CSDN博客](#)

[双十一破亿的最新相关信息](#)

[美的洗碗机双十一销售额破亿元大关,化身厨房神器走... 百家号](#)

在今天的**双十一**,这些热门机型都化身成厨房利器,走进千家万户,解放人们的双手,从而也带来突破**亿元**大关的销售佳绩,而未来,洗碗机普及道路还有待...

[山东双十一期间收发件总数破亿!你贡献了多少? 网易](#)

11)

[宠物生活也革命 双十一撬动中国“宠市”单日破亿 网易](#)

11)

[淘沙时代双十一为客户创造业绩破2亿! 凤凰网](#)

11)

[仅2分58秒破亿 努比亚双十一销售额惊人 凤凰科技](#)

11,

[双十一阿里交易额3分01秒破百亿 40分钟破500亿元|阿里... 新浪新闻](#)

新华社上海11月11日电(记者张璇、周琳)11日零时一过,“**双十一**”全球狂欢节正式进入第九
阿里巴巴集团披露的数据称,仅用了3分钟1秒,阿里巴巴平台...

news.sina.com.cn/c/nd/... - 百度快照

[盘点历年“双十一”淘宝销售数据 2017年有望破1500亿-闽南网](#)

[技术揭秘12306改造\(一\):尖峰日PV值297亿下可每秒出票1032张-CSDN...](#)

2015年2月15日 - 高流量高**并发**的问题,如果单靠硬件升级解决的话,可能需要扩充数十倍的硬件...3. 2012和2013年,12306 没有屏蔽**抢票**软件。在2014年以后,实现了...
<https://www.csdn.net/article/2...> - 百度快照

[解密12306在高并发下如何保证车票不出现多卖的情况\(一\)](#)

电商的秒杀活动以及12306的**抢票**这种业务,如果没有锁来保证数据的一致性,出现的情况...客户购买的票肯定超过了100张,但是在这种多线程**并发**的情况下很有可能造成票数多...
baijiahao.baidu.com/s?... - 百度快照

[十分钟在 12306 抢票千余张,黄牛的抢票软件是何原理? - 知乎](#)

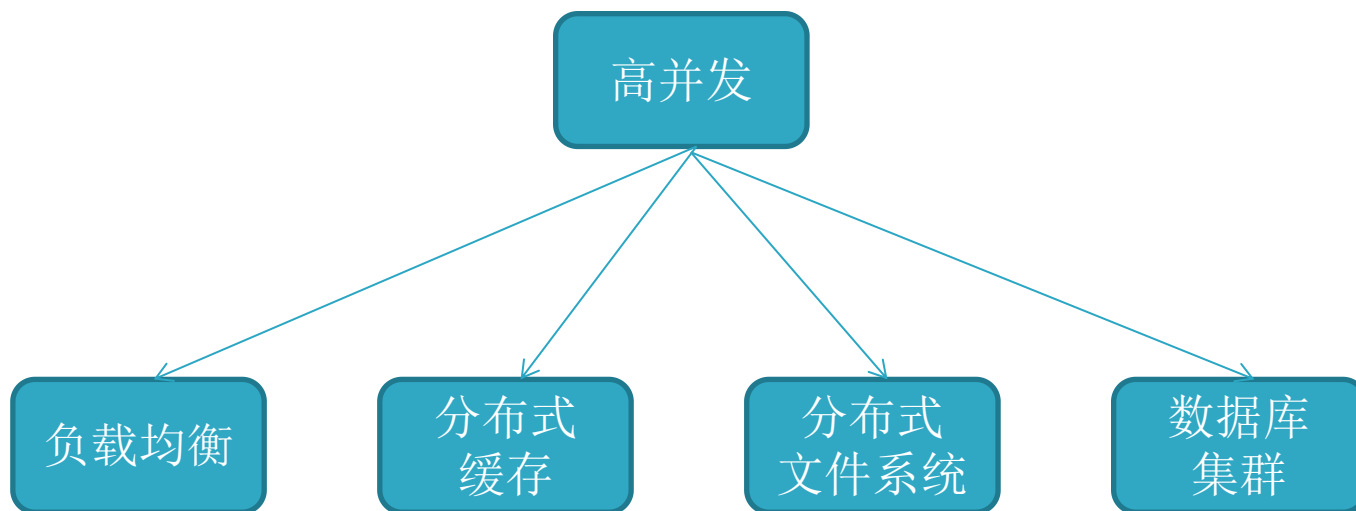
2017年6月29日 - 而且12306服务器端根本没有做IP**并发**请求的限制,所以一台电脑同时登陆几千个账号...黄牛可以依靠的就是资金量的大小而不是**抢票**软件的速度,而且也无法釜...
<https://www.zhihu.com/question...> - 百度快照

[12306 网站的负载到底有多大? - 知乎](#)

2015年6月16日 - 服务器计算能力够,并支持的**并发**连接数顶得住10万TCP链接的建立的话,那没有...十分钟在 12306 抢票千余张,黄牛的**抢票**软件是何原理? 54 个回答 为什么...
<https://www.zhihu.com/question...> - 百度快照

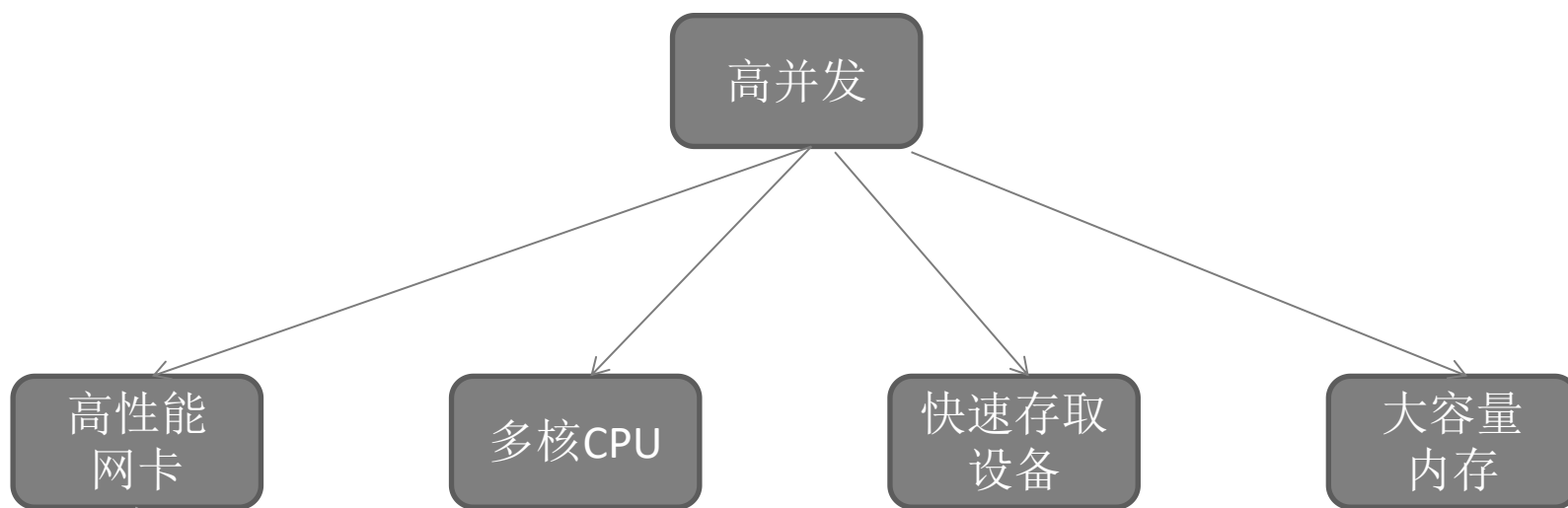


高并发—软件





高并发—硬件





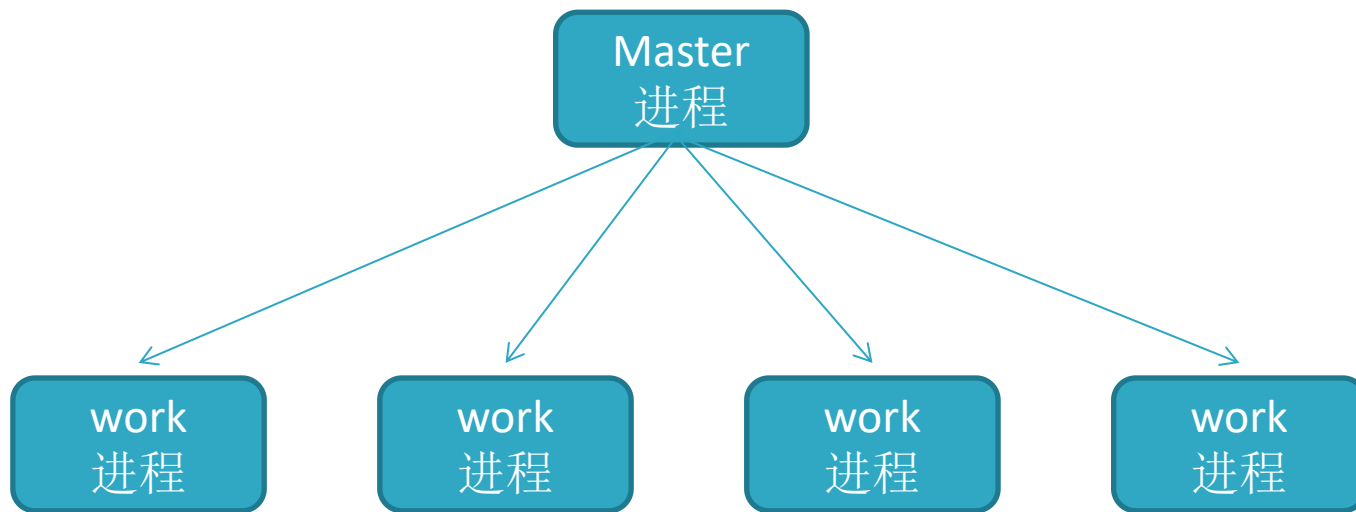
经典单机服务框架

1. 单进程异步: **Redis, HAProxy, Squid**
2. 多线程异步: **memcached**
3. 多进程异步: **nginx**
4. 每请求每进程（线程）: **Apache/CGI**
5. 微进程框架: **erlang/go**
6. 协程框架: **libco**



Nginx 进程设计

组成： 一个master 管理进程 + 多个完全相同worker 工作进程



优点：

- ✓ 充分利用多核系统的并发处理能力
- ✓ 负载均衡
- ✓ 职责明确，管理进程仅负责管理，工作进程仅负责处理业务逻辑



进程通信应用场景

- 紧急情况需要结束服务
- 需要重启应用服务，但不能影响已在线用户的访问，优雅的退出
- 热更新配置，不重启服务，不影响用户的使用
- 执行文件的热更新，不重启服务，不影响用户的使用
- 程序运行过程中出现致命错误，导致整个进程挂掉，父进程启动新的进程继续服务
- 多进程通信，需要再进程间传送数据，共享信息
- 等等。。。。。



Nginx进程间通信概述

Linux 进程间通信机制： 共享内存、套接字、管道、消息队列、信号。

Nginx主要使用了其中的三种方式：

- 套接字（全双工匿名套接字对 `socketpair`）
- 信号
- 共享内存



进程间通信利器 - 信号

信号 — 给程序提供一种可以处理异步事件的方法, 它利用软件中断实现。

- 特点:**
1. 和你一样, 有名字! SIG 开头, 如: 异常挂掉了 SIGABRT 6
 2. 在一下情况被触发:
 - 终端键 Ctrl+C
 - 硬件异常 (除0, 内存越界);
 - Kill 命令或kill 函数;
 - 软件条件发生;
 3. 随机出现

- 处理:**
- 忽略此信号 (但SIGKILL 和 SIGSTOP 例外)
 - 捕捉信号, 指定信号处理函数进行处理
 - 执行系统默认动作, 大多数都是终止进程



进程间通信利器 - 匿名套接字对 `socketpair`

建立双向通信的一对套接字，即每一端都可以进行读写

```
#include <sys/types.h>
```

```
#include <sys/socket.h>
```

```
int socketpair(int domain, int type, int protocol, int sv[2]);
```

参数说明：

`domain` — 表示协议族，仅可用：`AF_UNIX`(`AF_LOCAL`)。

`type` — 表示协议，可以是`SOCK_STREAM`或者`SOCK_DGRAM`。

`protocol` — 表示类型，只能为0。

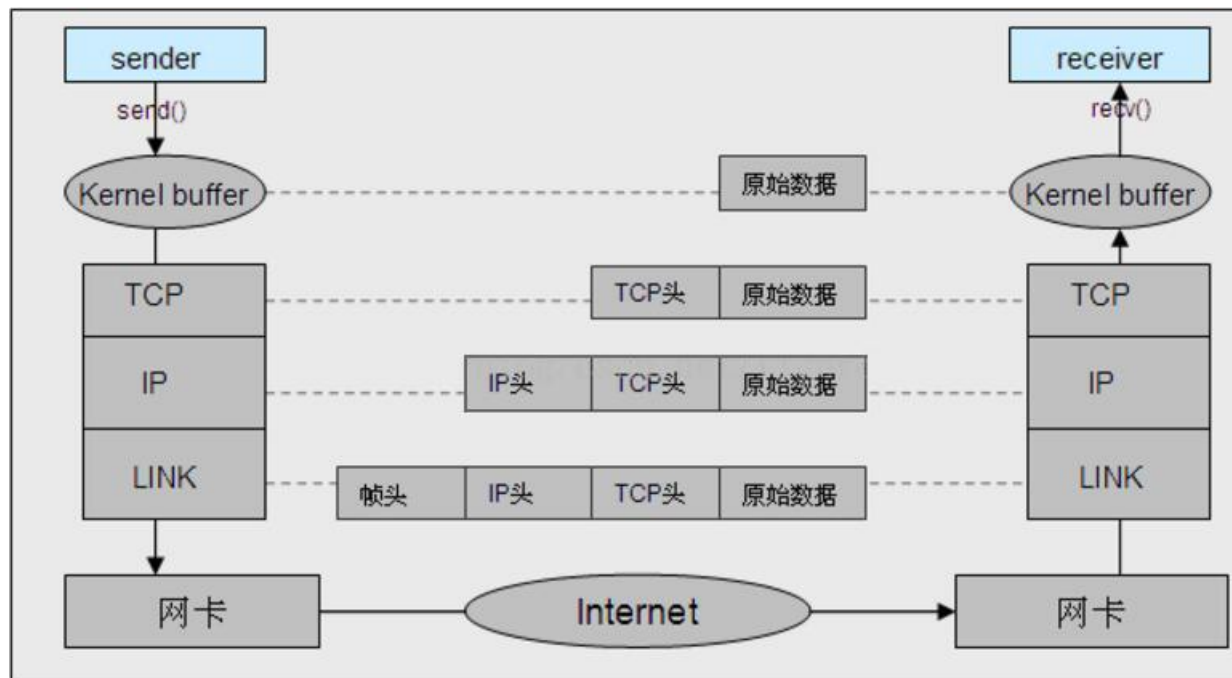
`sv[2]` — 是接收代表两个套接口的整数数组。每一个文件描述符代表一个套接口，并且与另一个并没有区别。

函数返回值：

如果函数成功，将会返回0值。否则将会返回-1表明创建失败，并且`errno`来表明特定的错误号。



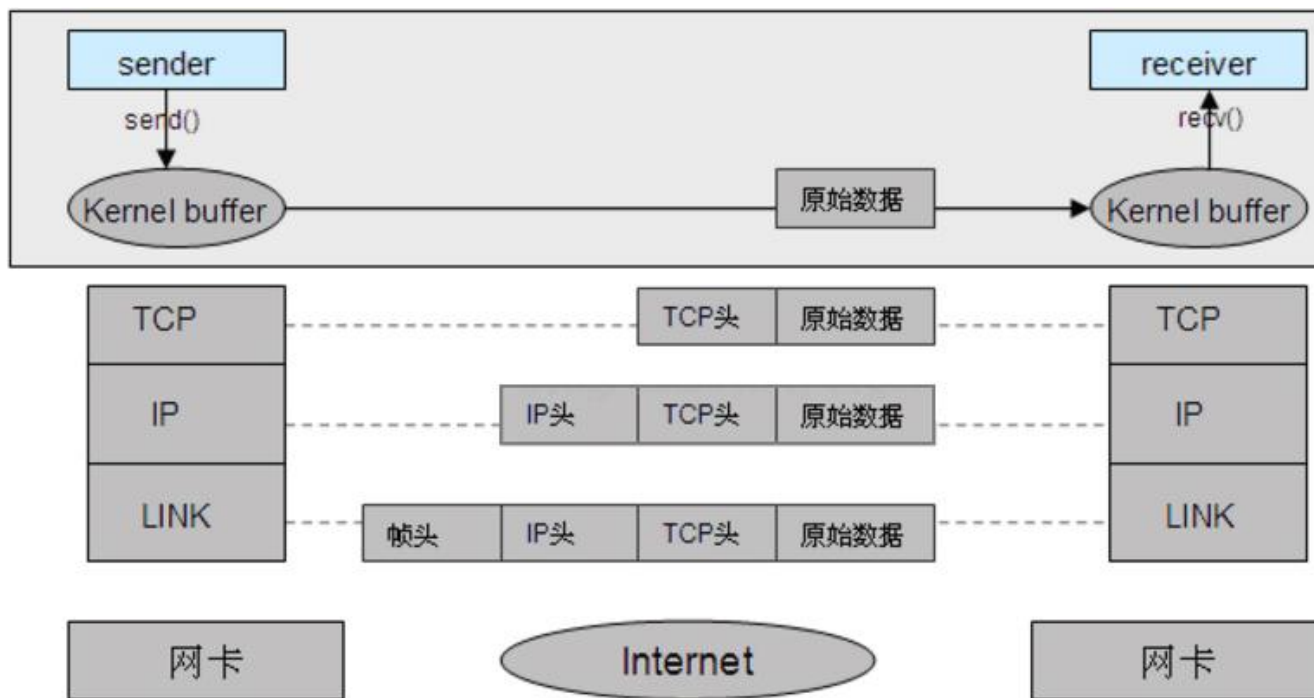
AF_UNIX 和 AF_INET 区别



AF_INET 通信图解



AF_UNIX 和 AF_INET 区别



AF_UNIX 通信图解

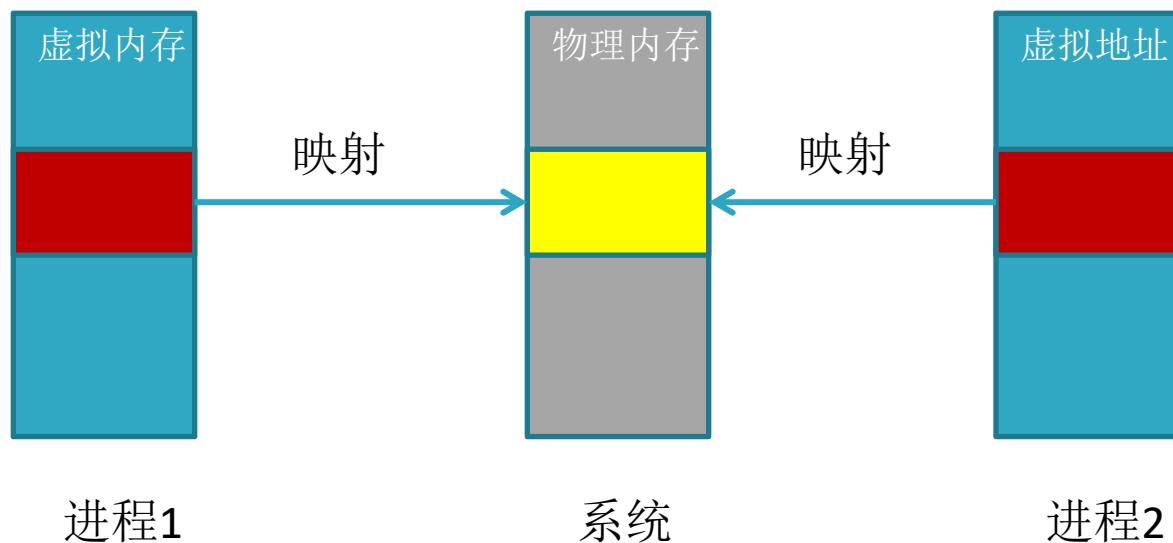


进程间通信 - 共享内存

两种常用共享内存方式： **shm & mmap**

➤ **IPC通信System V版本的共享内存 shm**

原理： 利用共享内存完成进程间通信，两个进程都通过虚拟地址空间到用户页表，然后通过用户级页表映射到物理内存的相同一块内存区域





进程间通信 - shm实现

1. ftok函数生成key标识符

key_t ftok(const char *pathname,int proj_id)

2. 创建一个共享内存块，返回这个共享内存块的标识符shmid

int shmget(key_t key,size_t size,int shmflg)

参数说明: size — 申请的共享内存的大小，为4k的整数倍；

shmflg — IPC_CREAT 创建新的共享内存,已存在 使用IPC_EXCL

3. 挂接共享内存（将进程地址空间挂接到物理空间，可以有多个挂接）

void *shmat(int shmid,const void *shmaddr, int shmflg)

参数说明: shmid — 挂接的共享内存ID.

shmaddr — 一般为0，表示连接到由内核选择的第一个可用地址上

shmflg — 一般为0

4. 取消共享内存映射

int shmdt(const void *shmaddr);

5. 用于控制共享内存

int shmctl(int shmid, int cmd, struct shmid_ds *buf);

参数 shmid — 由shmget返回的共享内存标识码

cmd — 将要采取的动作（可取值: IPC_STAT、IPC_SET、IPC_RMID）

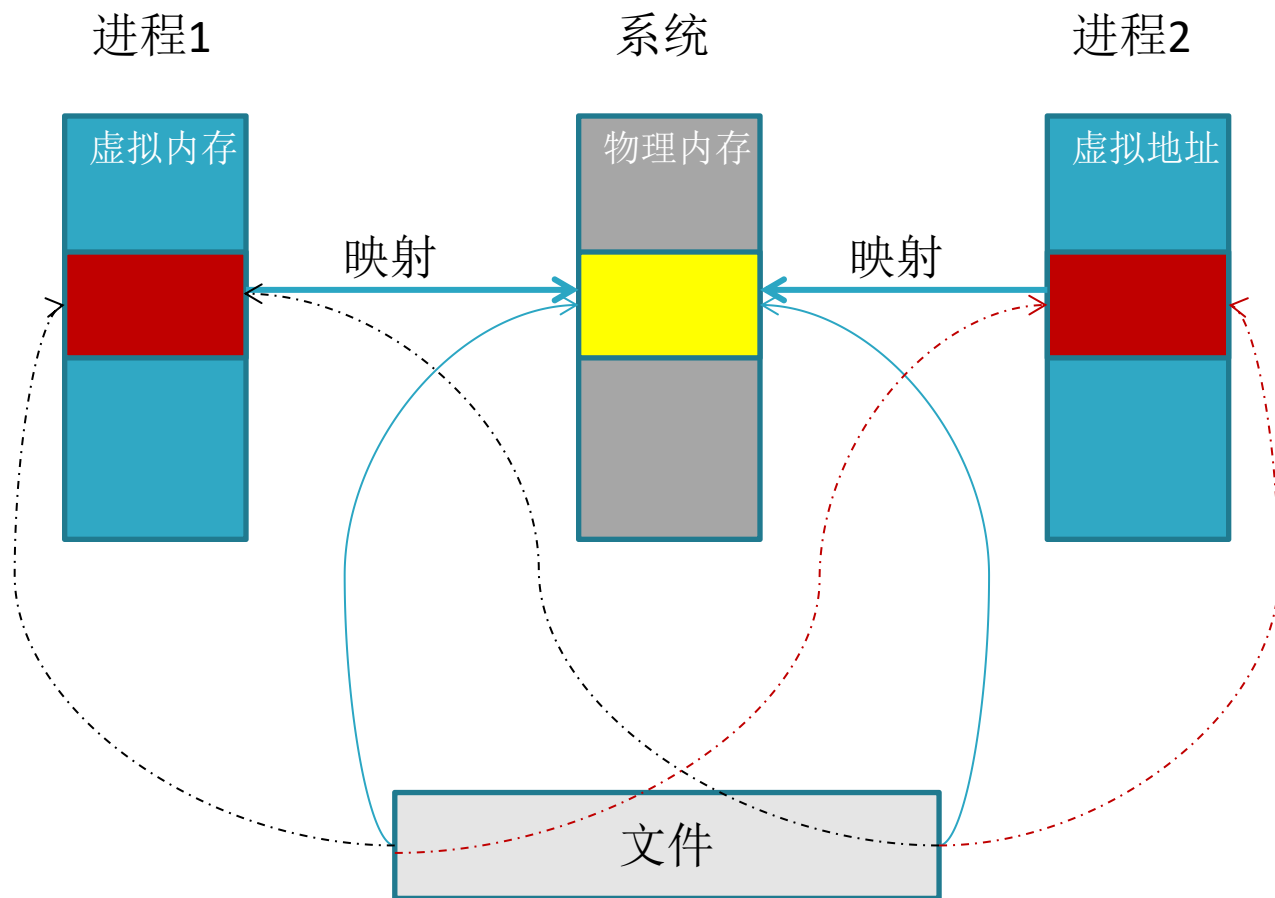
buf — 指向一个保存着共享内存的模式状态和访问权限的数据结构



进程间通信 - 存储映射共享I/O（mmap函数）

原理：将一个文件或者其它对象映射进内存。

1. 使用普通文件提供的内存映射：
2. 使用特殊文件提供匿名内存映射：





进程间通信 -mmap实现

```
#include <sys/mman.h>
```

```
void *mmap(void *addr, size_t length, int prot, int flags,  
           int fd, off_t offset);  
int munmap(void *addr, size_t length);
```

- 1、将一个普通文件映射到内存中，通常在需要对文件进行频繁读写时使用，这样用内存读写取代I/O读写，以获得较高的性能；
- 2、将特殊文件进行匿名内存映射，可以为关联进程提供共享内存空间；
- 3、为无关联的进程提供共享内存空间，一般也是将一个普通文件映射到内存中。

函数: `void *mmap(void *start, size_t length, int prot, int flags, int fd, off_t offsize);`

参数start: 指向欲映射的内存起始地址，通常设为 NULL，代表让系统自动选定地址，映射成功后返回该地址。

参数length: 代表将文件中多大的部分映射到内存。

参数prot: 映射区域的保护方式。可以为以下几种方式的组合:

PROT_EXEC 映射区域可被执行

PROT_READ 映射区域可被读取

PROT_WRITE 映射区域可被写入

PROT_NONE 映射区域不能存取



进程间通信 - mmap实现

```
#include <sys/mman.h>
```

```
void *mmap(void *addr, size_t length, int prot, int flags,  
           int fd, off_t offset);  
int munmap(void *addr, size_t length);
```

参数start: 指向欲映射的内存起始地址，通常设为 NULL，代表让系统自动选定地址，映射成功后返回该地址。

参数length: 代表将文件中多大的部分映射到内存。

参数prot: 映射区域的保护方式。可以为以下几种方式的组合：

PROT_EXEC	执行
PROT_READ	读取
PROT_WRITE	写入
PROT_NONE	不能存取



进程间通信 - mmap实现

参数flags: 影响映射区域的各种特性。必须要指定MAP_SHARED 或 MAP_PRIVATE。

MAP_SHARED	— 映射区域数据与文件对应，允许其他进程共享
MAP_PRIVATE	— 映射区域生成文件的copy，修改不同步文件
MAP_ANONYMOUS	— 建立匿名映射。此时会忽略参数fd，不涉及文件，而且映射区域无法和其他进程共享。
MAP_DENYWRITE	— 允许对映射区域的写入操作，其他对文件直接写入的操作将会被拒绝。
MAP_LOCKED	— 将映射区域锁定住，这表示该区域不会被置swap

参数fd: 要映射到内存中的文件描述符。如果使用匿名内存映射时，即flags中设置了MAP_ANONYMOUS，fd设为-1。有些系统不支持匿名内存映射，则可以使用fopen打开/dev/zero文件，然后对该文件进行映射，可以同样达到匿名内存映射的效果。

参数offset: 文件映射的偏移量，通常设置为0，代表从文件最前方开始对应，offset必须是分页大小的整数倍。



非常感谢您的观看!

