

## Ep01 昨日重现

### • 管道

- 流管道
  - 输入输出流重定向
- 无名管道
  - 只能在有亲缘关系的进程间使用（父子/兄弟）
  - 半双工，有固定的读端(fds[0])和写端(fds[1])
  - 是特殊的文件，可以用read, write读写管道，程序结束后消失
  - 先关闭管道读端再写管道时，写管道的进程会收到SIGPIPE信号，导致进程终止
- 命名管道
  - 创建管道文件，文件会一直存在，不会随进程结束而消失

### • 共享内存

- 创建共享内存：`int shmget(key, size, flags)`
- 将共享内存映射到进程的地址空间 `void *shmat(shmid, NULL, 0)`
- 解除映射 `int shmdt(const void *addr)`
- 控制共享内存：`shmctl()`;
  - IPC\_RMID：删除共享内存
  - IPC\_STAT：获取共享内存信息
  - IPC\_SET：修改共享内存信息
- 命令：
  - `ipcs`：查看创建的共享内存
  - `ipcrm -m shmid`：删除共享内存
  - shmid为共享内存id
  - 共享内存的删除是标记删除，只有当最后连接共享内存的进程解除映射之后，才会真正的删除这段共享进程
- 进程虚拟地址到物理地址的转换：多数为写时复制
  - 页目录>>页表项>>页内偏移（对于32位系统仅有3级）
  - TLB：快表，加快映射速度
    - 本质是因为cpu速度和内存速度差别过大
    - 将频繁访问的数据存入TLB（速度和cpu差不多）中，不造成cpu时间片的浪费
  - 当虚拟页和物理页的映射关系没有建立，并且要访问这段内存时，会发生缺页异常，需要调页。
  - 建立映射关系，可以通过使用大页，减少缺页异常。
- mmap方式实现共享内存
  - MAP\_SHARED
  - 速度快，可以多对多
  - 用法类似通告版

### • 信号量

- 分类（一个与多个）

- 单个信号量：可以看作是二进制开关：相当于锁
  - 对同一个数各加一千万
  - 若不加锁则会两个进程竞争导致所得数小于两千万
- 多个信号量：计数信号量→信号量的值代表资源数量，限制资源可以被多少进程访问
- 创建/获取信号量集合：int semget
- semop：改变信号量状态
  - P操作：对信号量-1，释放资源 本质可以看作加锁
  - V操作：对信号量+1 本质可以看作解锁
- semctl：信号量的控制
  - SETVAL：设置单个信号量的值
  - SETALL：设置全部信号量的值（此时需要一个指针类型存储设置值）
  - GETVAL：得到单个信号的值
  - GETALL：得到所有信号的值（此时需要一个指针类型存储获得值）
  - IPC\_RMID：删除信号量的值

## • 消息队列

- 消息队列的创建：int msgget(key, flag);
- 添加消息：int msgsnd(msgid, msgp, size, flag);
- 获取消息：int msgrcv(msgid, msgp, size, msgtype, flag); (可选NOWAIT参数，即不等待直接读取)
  - flg参数：常用为0，或者IPC\_NOWAIT，若此时队列满再调用msgsnd或者队列空的是hi调用msgrcv将直接返回错误
  - msgp为一个可变长度结构体，原型为

```

1 struct msgbuf{
2     long mtype;
3     char mtext[64];
4     //此处仅可以自定义大小
5 };

```

- 消息类型：即msgtype的三种选项
- msgtype = 0：接受信号队列中的第一个信息
- msgtype > 0：接受指定序列号的第一个消息
- msgtype < 0：接受类型小于等于msgtype绝对值的第一个最低类型消息
- 消息队列编号中，无0号队列
- 消息队列控制函数：int msgctl(msgid, cmd, buf);
  - 其中cmd常用为：IPC\_RMID
  - 表示删除消息队列
  - buf常用NULL
- 特点：使用方便，灵活，将任务异步运行