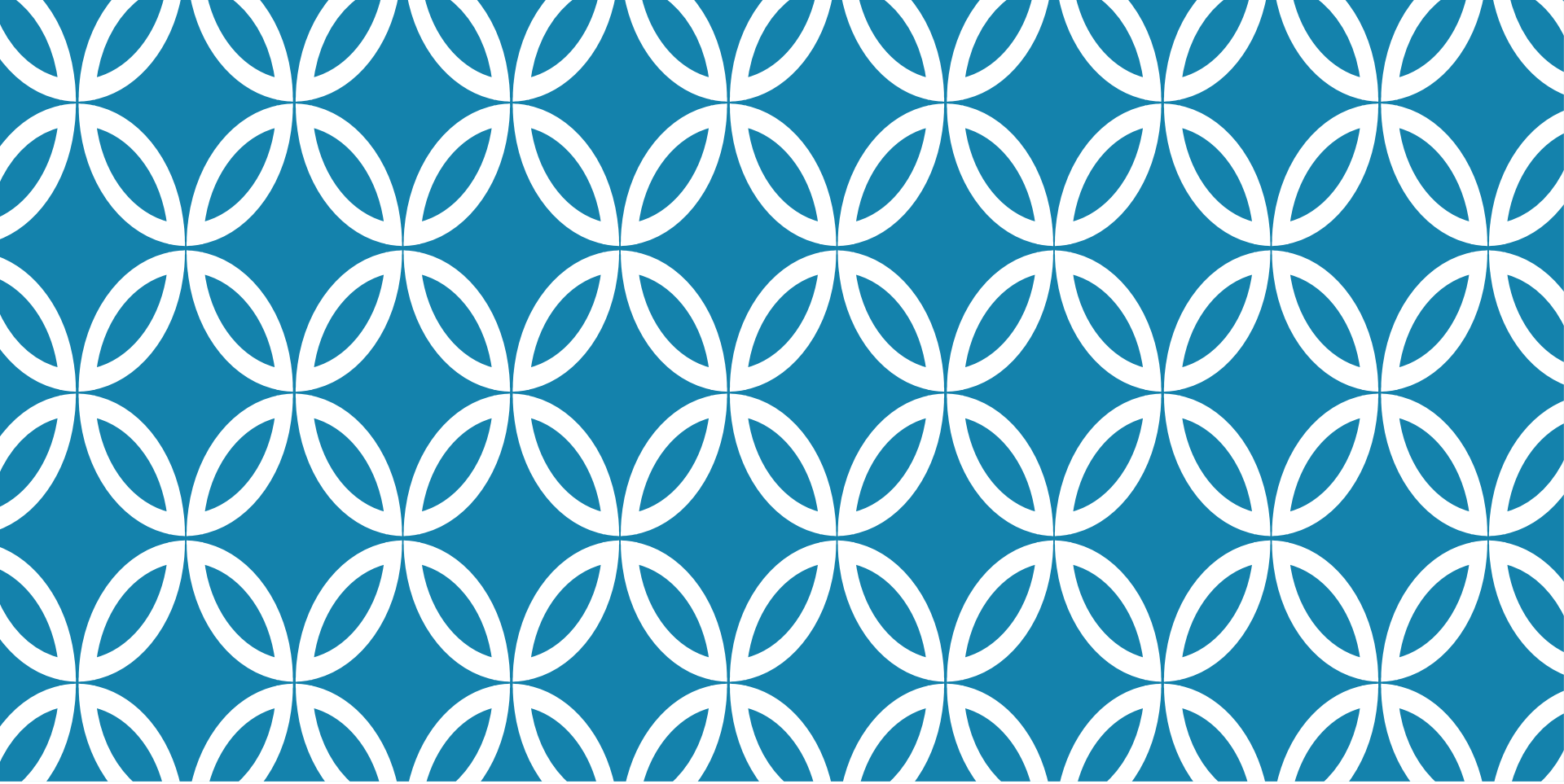

LINUX ： MESSAGE 示范

左航

MESSAGE

IPC:inter-process communication ，进程间通信对象；包括如下组件：

管道通信 ： FIFO,PIPE, 流式数据

消息队列： message queue

信号量： semaphore

共享内存： share memory

消息数据格式

```
struct Msg{
long type; // 消息类型。值必须 > 0 ，这个值被系统使用
******     // 消息正文，多少字节随你而定
};
```

例如：

```
struct Msg {
 long type;
 char isbncode[20];
} msg;
```

```
struct Msg {
   long type;
   int start;
   int end; } msg;
```

消息队列

max queues system wide = 16

【系统最多的消息队列数量（最多支持同时 16 个消息队列）】
max size of message (bytes) = 8192

【单个消息的最大字节数】
default max size of queue (bytes) = 16384

【默认的单个队列的大小 16384 】

消息队列相关的函数

```
// 创建和获取 ipc 内核对象
int msgget(key_t key, int flags);
// 将消息发送到消息队列
int msgsnd(int msqid, const void *msgp, size_t msgsz, int msgflg);
// 从消息队列获取消息
ssize_t msgrcv(int msqid, void *msgp, size_t msgsz, long msgtyp, int msgflg);
// 查看、设置、删除 ipc 内核对象 ( 用法和 shmctl 一样 )
int msgctl(int msqid, int cmd, struct msqid_ds *buf);
```

创建 函数

int msgget(key_t key, int flags);

创建一个新的或打开一个已经存在的消息队列，此消息队列与 key 相对应。

msgflag ：

IPC_CREAT: 创建新的消息队列。

IPC_EXCL: 与 IPC_CREAT 一同使用，表示如果要创建的消息队列已经存在，则返回错误。

IPC_NOWAIT: 读写消息队列要求无法满足时，不阻塞。返回值： 调用成功返回队列标识符，否则返回 -1.

第一位：0 表示这是个八进制数
第二位：当前用户的经权限 :6=110(二进制), 每一位分别对就 可读，可写，可执行 ,,6 说明当前用户可读可写不可执行
第三位 :group 组用户 ,6 的意义同上
第四位：其它用户，每一位的意义同上 ,0 表示不可读不可写也不可执行

创建 函数

int msgget(key_t key, int flags);

创建一个新的或打开一个已经存在的消息队列，此消息队列与 key 相对应。

msgflag ：

IPC_CREAT: 创建新的消息队列。

IPC_EXCL: 与 IPC_CREAT 一同使用，表示如果要创建的消息队列已经存在，则返回错误。

IPC_NOWAIT: 读写消息队列要求无法满足时，不阻塞。返回值： 调用成功返回队列标识符，否则返回 -1.

第一位：0 表示这是个八进制数
第二位：当前用户的经权限 :6=110(二进制), 每一位分别对就 可读，可写，可执行 ,,6 说明当前用户可读可写不可执行
第三位 :group 组用户 ,6 的意义同上
第四位：其它用户，每一位的意义同上 ,0 表示不可读不可写也不可执行

查看\设置\删除函数

所需头文件	#include<sys/types.h> #include<sys/ipc.h> #include<sys/msg.h>	
函数原型	int msgctl(int msgqid,int cmd,struct msgid_ds *buf)	
函数参数	msqid: 消息队列的队列 ID	
	cmd:命令参数	IPC_STAT：读取消息队列的数据结构 msqid_ds，并将其存储在 buf 指定的地址中
		IPC_SET：设置消息队列的数据结构 msqid_ds 中的 ipc_perm 域(IPC 操作权限描述结构)值，这个值取自 buf 参数
		IPC_RMID：从系统内核中删除消息队列
	buf：描述消息队列的 msqid_ds 结构类型变量	
函数返回值	成功：0 出错：-1	

表 4 msgctl()函数

查看\设置\删除函数

msgctl(msgid,IPC_RMID,NULL);// 移除消息队列

发送 函数

```
int msgid,ret;
key_t key=8;
msgid = msgget(key,IPC_CREAT |0660); // 创建标号为 8 的消息队列，如果已经存在打开使用
// 给用户和用户所在组赋予可读可写可执行权限
if(-1 == msgid)    {
    perror("msgid");
    exit(1);
}
```

接收函数

ssize_t msgrcv(int msqid, void *msgp, size_t msgsz, long msgtyp, int msgflg);

参数 msqid ： ipc 内核对象 id
参数 msgp ：用来接收消息 / 数据的变量地址
参数 msgsz ：消息正文部分的大小 (不包含消息类型)
参数 msgtyp ：指定获取哪种类型的消息

msgtyp = 0 ：获取消息队列中的第一条消息
msgtyp > 0 ：获取类型为 msgtyp 的第一条消息，除非指定了 msgflg 为 MSG_EXCEPT ，这表示获取除了 msgtyp 类型以外的第一条消息。
msgtyp < 0 ：返回队列中消息类型值小于或等于 type 绝对值，而且在这种消息中，其类型值又最小的消息。

参数 msgflg ：可选项，如果为 0 表示没有消息就阻塞。
IPC_NOWAIT ：如果指定类型的消息不存在就立即返回，同时设置 errno 为 ENOMSG
MSG_EXCEPT ：仅用于 msgtyp > 0 的情况。表示获取类型不为 msgtyp 的消息
MSG_NOERROR ：如果消息数据正文内容大于 msgsz ，就将消息数据截断为 msgsz

消息队列示意图

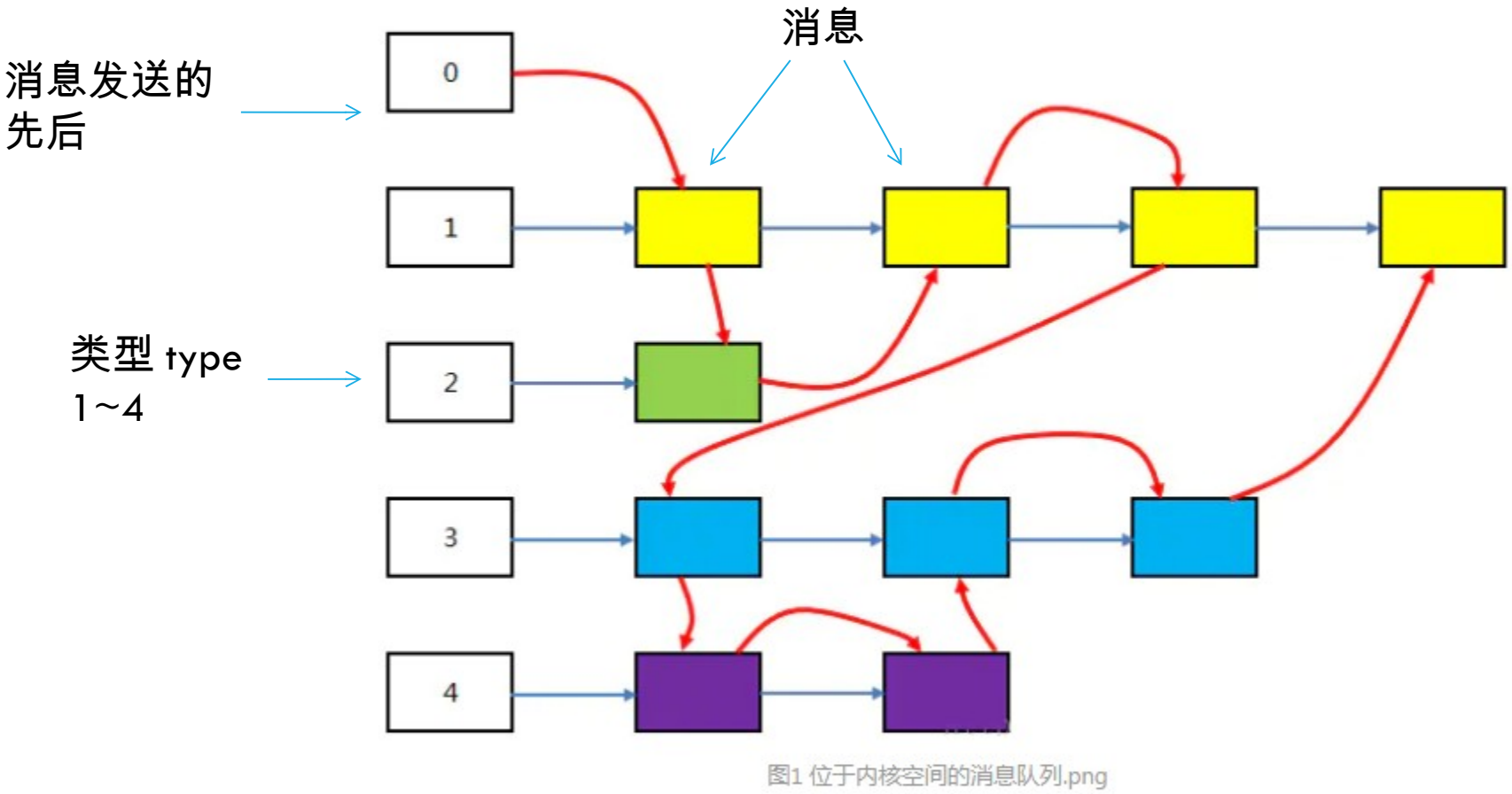

图1 位于内核空间的消息队列.png

示例

1. 线程收发
2. 进程收发
3. 生产者消费者
4. 临界区