



西安邮电大学
XI'AN UNIVERSITY OF POSTS & TELECOMMUNICATIONS

Linux 编程技术



第6章 进程间通信

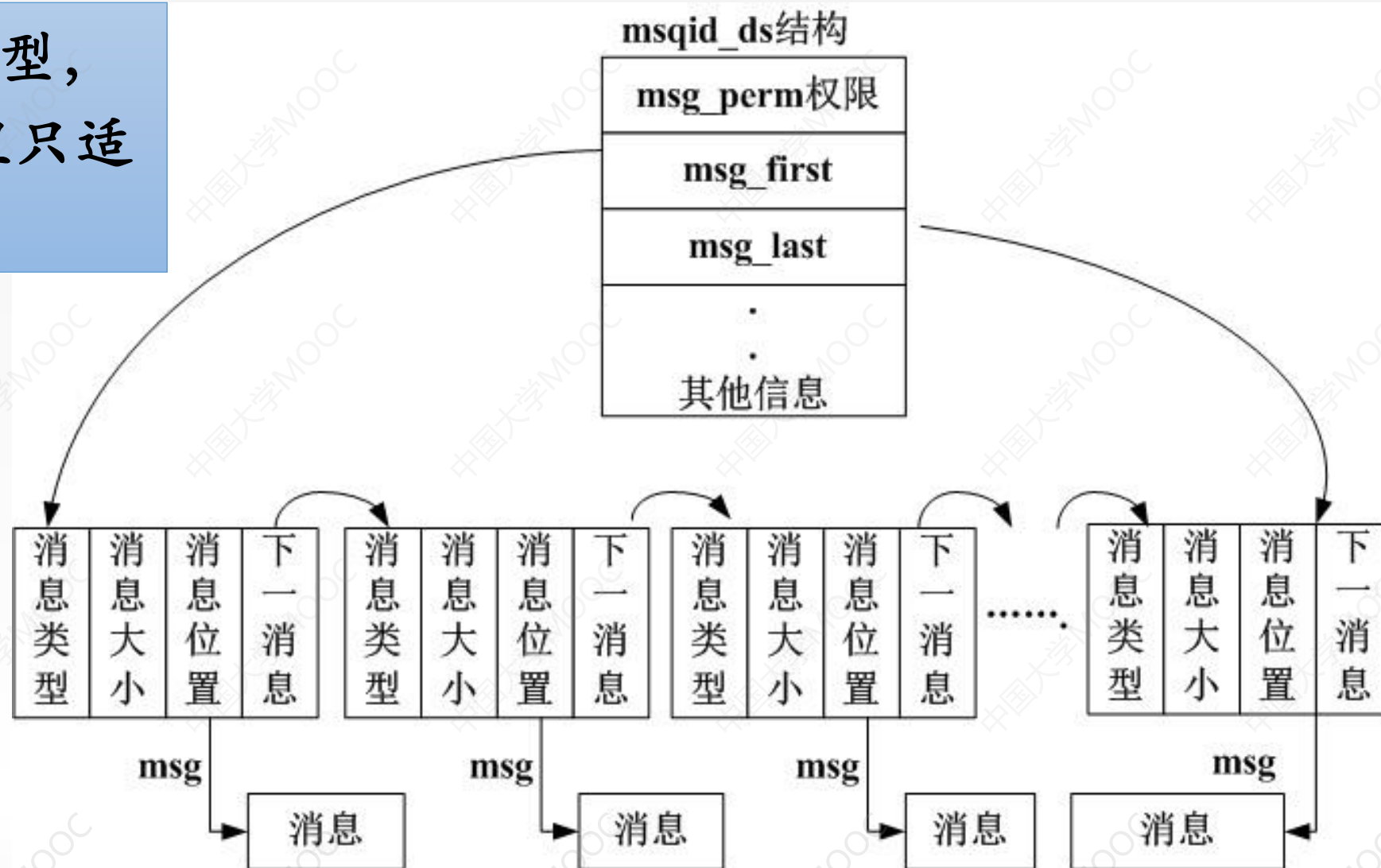
——消息队列



主讲：黄茹

6.5 消息队列

消息队列可以基于类型，
因此FIFO原则在这里只适用
于同类型的消息。



msgget	
功能	创建一个消息队列
头文件	/usr/include/sys/msg.h
函数原型	int msgget(key_t key, int flag);
参数	key 创建IPC的键值
	flag 消息队列的权限
返回值	非-1 消息队列的ID
	-1 失败

```
struct mymsg  
{  
    long msgtype;  
    char msgtext[256];  
}
```

- flag除了可以直接设置为0644这样的权限外，还可以附加以下参数值：

```
#define IPC_CREAT 00001000
```

```
#define IPC_EXCL 00002000
```

```
#define IPC_NOWAIT 00004000
```

msgsnd		
功能	发送一个消息到消息队列	
头文件	/usr/include/sys/msg.h	
函数原型	int msgsnd(int msqid, void * ptr, size_t nbytes, int flag);	
参数	msqid	消息队列的ID
	ptr	指向消息的指针
	nbytes	消息的长度
	flag	当消息队列已满时，如何处理
返回值	0	成功
	-1	失败

- ptr指向一个用户定义的结构体数据，该结构体必须包含消息的类型(正数)和消息的数据。例如可定义如下：

```
struct mymsg{  
    long msgtype;  
    char msgtext[256];  
}
```

- 若队列已满，则根据flag的设置，msgsnd做出不同的反应。
 设置为IPC_NOWAIT时：若消息队列已满，则使用msgsnd立即出错；
 设置为0时：则调用msgsnd的进程被挂起，直到消息队列可接受该消息或该队列被删除或捕捉到信号，从信号处理程序中返回。

msgrcv	
功能	从消息队列接收一个消息
头文件	/usr/include/sys/msg.h
函数原型	int msgrcv(int msqid, void * ptr, size_t nbytes, long type, int flag);
参数	msqid 消息队列的ID
	ptr 保存消息的内存的指针
	nbytes 消息的长度
	type 指定请求的消息的类型
	flag 若指定的消息不存在，如何处理
返回值	>-1 消息的数据部分的长度
	-1 失败

- 消息长度大于nbytes时，若flag中设置了MSG_NOERROR，则截短该消息；如未设置，则出错返回，消息仍在队列中。
- type用来指定想要哪种类型的消息
 - type==0 返回队列中的第一个消息；
 - type>0 返回队列种类型为type的第一个消息；
 - type<0 返回队列中消息类型值小于或等于-type的消息，如果这种消息有多个，则取类型值最小的消息。
- flag设置为IPC_NOWAIT时，若没有符合条件的消息，则msgrcv返回-1，如未设置，则进程阻塞直到：有了满足条件的消息或从系统中删除了该队列或捕捉到一个信号并从信号处理程序返回。

msgctl		
功能	控制消息队列的基本属性	
头文件	/usr/include/sys/msg.h	
函数原型	int msgctl(int msqid, int cmd, struct msqid_ds *buf);	
参数	msqid	消息队列ID
	cmd	要执行的操作
	buf	存放消息队列属性的内存地址
返回值	0	成功
	-1	失败

- cmd:
 - IPC_STAT 获取消息队列的msqid_ds结构，将其存放在由buf指向的结构中。
 - IPC_SET 按buf指向结构中的值设定此队列相关结构中的msg_perm.uid、msg_perm.gid、msg_perm.mode、msg_perm.qbytes
 - IPC_RMID 从系统中删除消息队列及其中的数据



西安邮电大学
XI'AN UNIVERSITY OF POSTS & TELECOMMUNICATIONS

Linux 编程技术

谢谢大家!