



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

——创建、操作信号量集合



主 讲：黄 茹

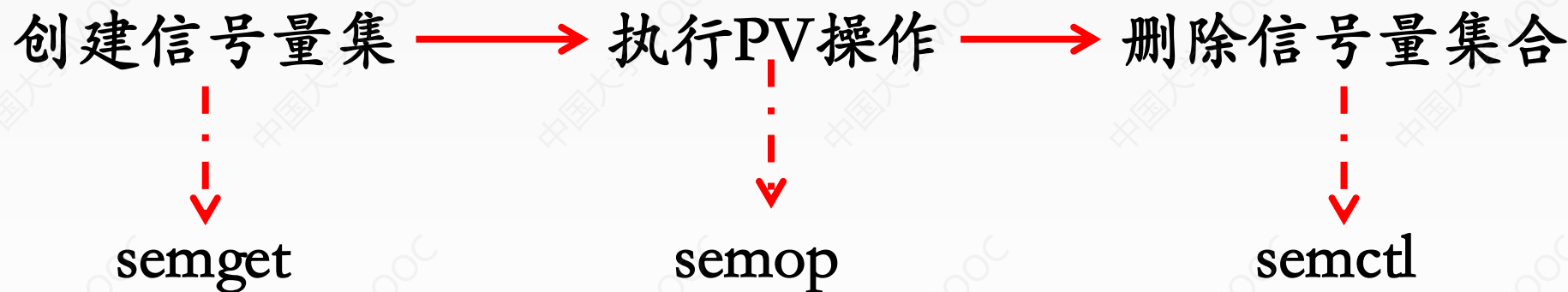
- 信号量是一个内存变量，可以被系统中的任何进程所访问。
- 多个进程使用信号量来协调对临界资源的访问。
- Linux中信号量是以集合的形式存在的，一个集合中存在着多个信号量。

- 1) 测试与该临界资源对应的信号量;
- 2) 如果信号量值大于0, 则进程将信号量值减1, 随后进入临界区, 执行访问临界资源的代码;
- 3) 如果信号量值不大于0, 则阻塞当前进程。

当进程不再使用临界资源，退出临界区时，需要完成以下操作：

- 1) 将信号量值加1；
- 2) 如果有阻塞的进程正等待使用临界资源，唤醒排在阻塞队列队首的进程。

使用信号量的流程：



semget	
功能	创建一个信号量集合
头文件	/usr/include/sys/sem.h
函数原型	int semget(key_t key, int nsems, int flag);
参数	key 创建信号量集的键值
	nsems 集合中信号量的个数
	flag 信号量集的权限
返回值	>0 信号量集的ID
	-1 失败

说明：

创建时必须指明nsems；引用时，将nsems指定为0。

semctl	
功能	操作信号量集合
头文件	/usr/include/sys/sem.h
函数原型	int semctl(int semid, int semnum, int cmd, union semun arg);
参数	semid 信号量集ID
	semnum 要操作的信号量序号
	cmd 要执行的命令
	arg 修改信号量时所需的数据或 获取信号量数据时所需的变量(可选)
返回值	与cmd相关

- arg为可选参数，其类型union semun类型必须自行实现，且类型定义是固定的。

```
semun
```

```
{
```

```
    int val;
```

```
    struct semid_ds *buf;
```

```
    unsigned short *array;
```

```
}
```


cmd	含义
IPC_STAT	取信号量集的属性，存放在arg.buf所指单元
IPC_SET	按arg.buf所指单元的数据设置信号量集合中的sem_perm.uid、sem_perm.gid、sem_perm.mode三个属性。
IPC_RMID	删除该信号量集合
GETVAL	返回第semnum个信号量的semval值
SETVAL	用arg.val设置第semnum个信号量的semval值
GETALL	取信号量集所有信号量的值，存放在arg.array指向的数组中
SETALL	按照arg.array所指数组的值设置信号量集中所有信号量的值

例4：初始化信号量中的某个信号

```
init_sem(int semid,int semnum,int val){  
    union semun{  
        int val;  
        struct semid_ds *buf;  
        unsigned short *array;  
    }initval;  
    initval.val=val;  
    if(semctl(semid,semnum,SETVAL,initval)==-1) {  
        perror("semctl");  
        exit(1);  
    }  
}
```



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

Linux 编程技术

谢谢大家!