(一) Unix System V的三种ipc资源查看方法

查看IPC通信资源可以使用ipcs命令：

[root@D2 lqm]# ipcs

------ Shared Memory Segments --------

key shmid owner perms bytes nattch status

0x6c6c6536 0 root 600 4096 0

0x00000000 32769 gdm 600 393216 2 dest

0x00000000 65538 gdm 600 196608 2 dest

0x00000000 98307 gdm 600 393216 2 dest

------ Semaphore Arrays --------

key semid owner perms nsems

0x00000000 0 root 600 1

0x00000000 65537 root 600 1

------ Message Queues --------

key msqid owner perms used-bytes messages

该命令显示了3种ipc资源：共享内存、信号量集和消息队列。

也可使分别显示其中一种:

ipcs -m 为共享内存；ipcs -s 为信号量；ipcs -q为消息队列。

这些资源的使用权限与文件是使用权限类似，上述系统输出中的perm列就是访问权限的标志，共三个8进制数字，分别对应资源属主、同组用户和其他用户，每个8进制数字的三个儿进制位分别表示可读可写和可执行。

（二）共享内存

创建system V IPC 共享内存：

int shm\_id; /\*.......\*/

...

shm\_id=shmget(IPC\_PRIVATE, BUFSZ, 0666 ) ;

此时可以用ipcs -m查看到有一个新的共享内存区出现，其编号等于上述shm\_id。其他进程可以使用该id来指定这个共享内存区。

创建子后需要将共享内存区映射到本进程的进程空间中：

shm\_buf = shmat( shm\_id, 0, 0)

shmat对应于（shared memory attach）其中shm\_id用于指定选用ipc 共享内存区中的哪一个。如果映射成功则返回共享内存区在本进程上的地址指针，否则则返回空指针。

使用结束后若不再需要该共享内存区则可以解除映射：

shmdt(shm\_buf)

shmdt()名字对应（shared memory detach），如果失败则返回值小于0。

（三）posix 信号量

posix信号量分为匿名有名两种，这里只讨论有名信号量。

有名信号量使用前，先需要创建：

sem\_t \*sem;

...

sem=sem\_open(sem\_name,O\_CREAT,0644,1);

其中信号量名sem\_name是一个字符串

执行post/V操作:

sem=sem\_open(argv[1],0);

sem\_post(sem);

sem\_getvalue(sem,&val);

执行wait/P操作：

sem=sem\_open(argv[1],0);

sem\_wait(sem);

sem\_getvalue(sem,&val);

不需要时可以删除：

sem\_unlink(sem\_name)

（四）其他 同步方式请学生自行学习：

posix的匿名信号量使用sem\_init()创建;

posix的互斥量为pthread\_mutex\_t

system V的信号量集的实现

（五）消息队列

消息队列使用邮箱来存储消息，ipcs –q 可以查看到不同的有效以及上面的消息数量。

创建邮箱的方法为：

msgqueue\_id = msgget(key, IPC\_CREAT|0660)

其中key可以从ftok()函数来生成，ftok()将一个文件路径名转换成一个key值。

在邮箱创建之后，想要用这个邮箱的进程需要执行类似下面的代码：

msgqueue\_id = msgget(key, IPC\_CREAT|0660)

在发送消息之前，需要构建消息。消息队列中的消息是有格式的：

struct mymsgbuf {

long mtype;

char mtext[MAX\_SEND\_SIZE];

};

其中mtype成员指出消息类型，使用整数数字来表示类型编码。

发送消息到指定邮箱：

msgsnd(qid, (struct msgbuf \*)qbuf, strlen(qbuf->mtext)+1, 0)

从指定邮箱接收消息：

msgrcv(qid, (struct msgbuf \*)qbuf, MAX\_SEND\_SIZE, type, 0);

使用完成后，如果需要撤销邮箱：

msgctl(qid, IPC\_RMID, 0);

（六）管道

管道分为匿名管道和FIFO两种。

1. FIFO

创建“mkfifo –m XXX *fifoname”* 其中xxx是访问权限（8进制） fifoname是管道名。创建后可以用ls –l查看这个管道文件。

[root@D2 lqm]# mkfifo -m 600 myfifo1

[root@D2 lqm]# ls -l

total 0

prw------- 1 root root 0 Oct 13 22:28 myfifo1

可见管道文件在ls –l命令的输出中，第一列为“p”表示管道文件，以区别普通文件、目录和设备文件等。

1. 匿名管道

匿名管道仅存在于父子进程间，无法在任意两个进程间使用。通常先创建管道，然后再创建父子进程。

创建管道前线需要声明两个文件描述符，然后再用pipe()创建。

int fds[2]; 声明两个文件描述符（整数）

pipe(fds); 由pipe()创建管道

创建后，第一个文件描述符（本例中的fds[0]对应读端，fds[1]对应写端）。

管道的读写和普通文件的读写相同，使用read()和write()完成。

另外在shell中，使用在两个命令之间用“|”表示匿名管道，例如ls|more，就是将ls的输出通过管道作为more的输入。