

# 实验三、共享内存与进程同步

## 一、实验目的

- 1、掌握Linux下共享内存的概念与使用方法；
- 2、掌握环形缓冲的结构与使用方法；
- 2、掌握Linux下进程同步与通信的主要机制。

## 二、实验内容

利用多个共享内存（有限空间）构成的环形缓冲，将源文件复制到目标文件，实现两个进程的誊抄。

## 三、预备知识

### 1、共享内存

使用共享内存是运行在同一计算机上的进程进行进程间通信的最快的方法。

shmget与shmat 系统调用：

```
int shmget(key_t key, int size, int shmflg)
```

```
IPC_CREAT|0666
```

```
int shmat ( int shmid, char *shmaddr, int shmflg)
```

```
S = (char *)shmat(shmid1, NULL, SHM_R|SHM_W)
```

- 共享存储区的控制shmctl:对其状态信息进行读取和修改。

系统调用格式: `int shmctl(shmid, cmd, buf);`

`int shmid, cmd;`

`struct shmid_ds *buf;`

其中, buf是用户缓冲区地址, cmd是操作命令:

- (1) 用于查询有关共享存储区的情况。
- (2) 用于设置或改变共享存储区的属性。
- (3) 对共享存储区的加锁和解锁命令。
- (4) 删除共享存储区标识符等。

如`shmctl(shmid, IPC_RMID, 0)`

## 2、进程控制

fork与execv系统调用

```
pid_t  p1;

if ((p1=fork()) == 0) {           //sub
    puts("get created\n");
    execv("./get", NULL);
} else { //main
    .....
}
```

### 3、Linux下的信号灯及其P、V操作

见实验二相关内容

## 4、环形缓冲

缓冲的目的是为了匹配CPU与设备的速度差异和负荷的不均衡，从而提高处理机与外设的并行程度。

- 缓冲技术可以用硬件缓冲器来实现，通常容量较小
- 软件缓冲是应用较广泛的一种缓冲技术，由缓冲区和对缓冲区的管理两部分组成

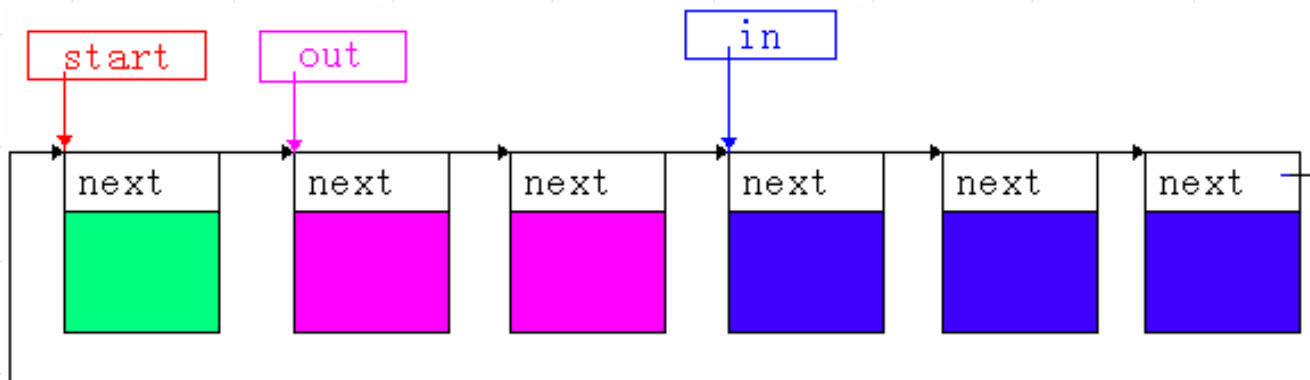
常用的缓冲技术：

单缓冲、双缓冲、环形缓冲、缓冲池

## 4、环形缓冲

在主存中分配一组大小相等的存储区作为缓冲区，并将这些缓冲区链接起来。

可循环使用这些缓冲区：输入指针in、输出指针out



除了链表方式，还可以使用数组，如

```
buffers=Data[10];
```

```
i = i mod 10;
```



## 5、编译、编辑、调试

```
cc -o sub1 sub1.c
```

```
vi
```

```
gdb
```

## 四、实验指导

主函数：main()

```
{ 创建共享内存组;  
  创建信号灯;  
  信号灯赋初值;  
  创建两个进程readbuf、writebuf;  
  等待两个进程运行结束;  
  删除信号灯;  
  删除共享内存组;  
}
```

Readbuf负责读、writebuf负责写，如何定义？