

4.4 何谓原语？它与系统调用有何区别？如何实现原语执行的不可分割性？

原语完成某种功能且不被中断执行的操作序列。

原语更加强调操作的不可分割性，即一个不可中断的子程序调用，但是系统调用是由用户态进入核心态，系统调用若遇到更高更紧迫的系统调用时，可以打断原来的系统调用。

原语执行的不可分割性可通过硬件方法实现，例如“Test_and_Set”指令(多处理机)和“Swap”指令，在单处理机上用关闭中断的方法实现。

4.13 假设有三个并发进程（P.Q.R），其中 P 负责从输入设备上读入信息并传递给 Q,Q 将信息加工后传送给 R,R 则负责将信息打印输出。

写出下列条件的并发程序：

（1）进程 P,Q 共享一个缓冲区，进程 Q, R 共享另一个缓冲区；
设信号量如下：

e1=1, e2=1, f1=0, f2=0

P:

Do

```
{
    p(e1);
    add item to buffer1 list;
    v(f1);
}while(1);
```

Q:

Do

```
{
    P(f1);
    Handle item from buffer1 list;
    V(e1);
    P(e2)
    Transfer item to buffer2 list;
    V(f2);
}while(1);
```

R

do

```
{
    P(f2);
    Print item from buffer2 list;
    V(e2);
```

```
}while(1);
```

(2) 进程 P, Q 共享一个由 m 个缓冲区组成的缓冲池, 进程 Q, R 共享另一个由 n 个缓冲区组成的缓冲池 (假设缓冲池足够大, 进程间每次传输信息的单位均小于等于缓冲区长度);

```
e1=m,e2=n,f1=0,f2=0, mutex1=1;mutex2=1;
```

```
P:
Do
{
P(e1);
P(mutex1);
add items to  buffer1 list;
V(f1);
V(mutex1);
}while(1);
```

```
Q:
Do
{
P(f1);
P(mutex1);
handle items from buffer1 list;
V(e1);
V(mutex1);
P(e2);
P(mutex2);
Transfer items to buffer2 list;
V(f2);
V(mutex2);
}while(1);
```

```
R:
Do
{
P(f2);
P(mutex2);
Printf items from buffer2 list;
V(e2);
V(mutex2);
}while(1);
```

4.26 设系统有三种类型的资源，数量为 (4, 2, 2)，系统中有进程 A, B, C。按如下顺序请求资源：

进程 A 申请 (2, 2, 1)

进程 B 申请 (1, 0, 1)

进程 A 申请 (0, 0, 1)

进程 C 申请 (2, 0, 0)

该进程按照死锁防止第 2 种资源剥夺法分配资源。试对上述请求序列，列出资源分配过程。指出哪些进程需要等待资源，哪些资源被剥夺。进程可能进入无限等待状态吗？

解：1.进程 A 申请资源

发现有资源，申请成功，剩余资源数 (2, 0, 1)

2.进程 B 申请资源

发现有资源，申请成功，剩余资源数 (1, 0, 0)

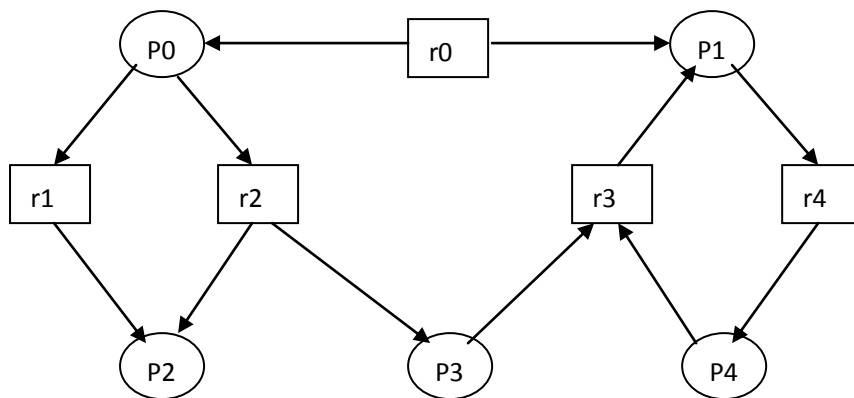
3.进程 A 申请资源

发现资源被占用，检查 B 是否处于等待资源状态，若是则剥夺其资源，否则将 A 设置为资源等待。此时将 A 设置为资源等待。

4.进程 C 申请资源

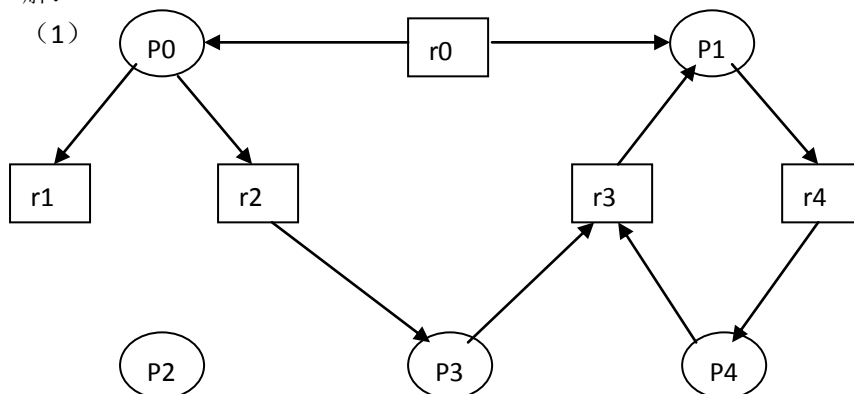
发现资源被占用，检查 A 处于资源等待状态，则剥夺 A 占有的资源，分配给 C。
进程不会进入无限等待状态。

4.31 化简如图所示的资源分配图，并说明有无进程处于死锁状态。

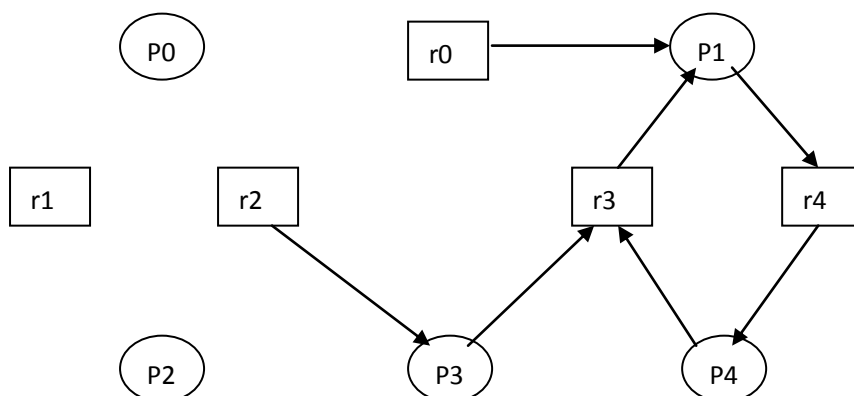


解：

(1)



(2)



P1、P3、P4 的进程处于死锁状态。

5.17 在页式虚存管理系统中，系统为用户提供了 2^{24} 字节的虚存空间。系统有 2^{20} 字节的主存空间。每页的大小是 512 字节。设用户给出了 11123456（八进制）的虚存地址。试说明系统怎样得到相应的物理地址。列出各种可能，并指出那些工作由硬件完成，那些工作由软件完成。

答：

(1) 由于每页大小是 512 字节，所以用户给出的 11123456 虚存地址的页号是 11123，页内地址是 456。

(2) 从虚存地址得到物理地址的过程如下：

cpu 从联想存储器中查找逻辑页号为 11123 对应的物理页帧号：

如果有：

则将物理页帧号和页内偏移合成物理地址进行访问；

否则从页表中查找：

在页表中找到相应项时，要查看该页的合法位是否置上：

若为 1，则将该项指出的页帧号和页内偏移合成物理地址进行访问；

若合法位为 0，则产生页故障，并执行页故障处理：

系统将页表项所指的辅存块号调入主存；

改写页表内的合法位为 1；

将页帧号填写入页表中对应的表项；

将页号和页帧号信息填写如联想存储器中；

将物理页帧号和页内偏移合成物理地址进行访问。

(3) 硬件完成的工作：获得页帧号，合成物理地址，以及合法位是否置上等。

软件完成的工作：接收缺页异常，并进行处理，返回现场。

5.18 某程序大小为 **460** 个字。考虑如下访问序列：**10, 11, 104, 170, 73, 309, 189, 245, 246, 434, 458, 364**，页帧大小为 **100** 个字，驻留集大小为两个页面

1、 给出访问串 **2、** 分别求出采用 **FIFO, LRU** 和 **OPT** 替换算法控制上述访问串的故障数和页故障率

1、

访问序列是 10, 11, 104, 170, 73, 309, 189, 245, 246, 434, 458, 364

由于页帧大小为 100 个字，所以访问序列可以简化为：

0, 0, 1, 1, 0, 3, 1, 2, 2, 4, 4, 3

2、

FIFO 算法

页面	0	0	1	1	0	3	1	2	2	4	4	3
页 0	0	0	0	0	0	1	1	3	3	2	2	4
页 1			1	1	1	3	3	2	2	4	4	3
缺页	0	1	0	1	1	0	1	0	1	0	1	0

缺了 6 次，故障率 50%

LRU 算法

页面	0	0	1	1	0	3	1	2	2	4	4	3
页 0	0	0	0	0	0	0	1	1	1	4	4	4
页 1			1	1	1	3	3	2	2	2	2	3
缺页	0	1	0	1	1	0	0	0	1	0	1	0

缺了 7 次，故障率 58.3%

OPT 算法

页面	0	0	1	1	0	3	1	2	2	4	4	3
页 0	0	0	0	0	0	3	3	3	3	3	3	3
页 1			1	1	1	1	1	2	2	4	4	4
缺页	0	1	0	1	1	0	1	0	1	0	1	1

缺了 5 次，故障率 41.67%

6.6 以下的工作各在 3 个 I/O 软件层的哪一层完成？

- ①为一个磁盘读操作计算磁道、扇区、磁头。
- ②维护一个最近使用的磁盘块得缓冲。
- ③向设备寄存器写入命令。
- ④检查用户是否有权使用设备。
- ⑤将二进制整数转换成 ASCII 码以便打印。

答：（1）设备驱动及中断处理

（2）与设备无关的 I/O

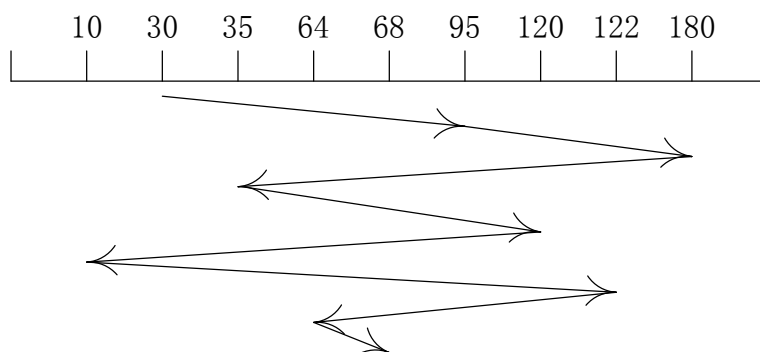
（3）设备驱动及中断处理

（4）与设备无关的 I/O

(5) 用户层 I/O

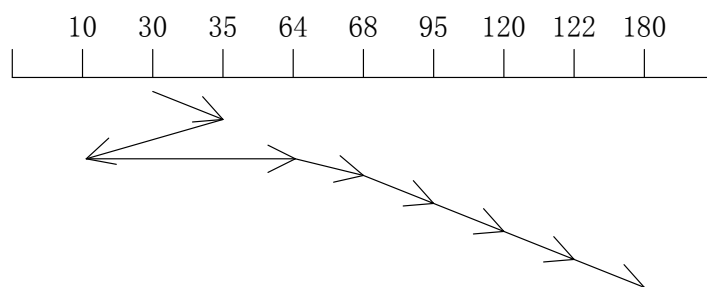
6.11 假设对磁盘的请求串为 95, 180, 35, 120, 10, 122, 64, 68, 磁头初始位置为 30, 试分别画出 FCFS, SSTF, SCAN, C-SCAN 调度算法的磁头移动轨迹及磁头移动的磁道数真 (磁道号: 0~199)。

答: FCFS:



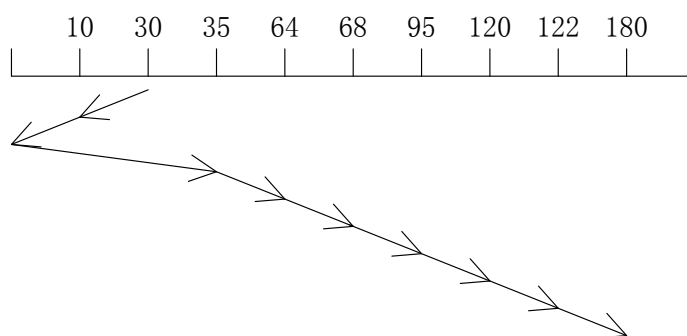
磁头移动的磁道数为: $65+85+145+85+110+112+58+4=664$ 。

SSTF:



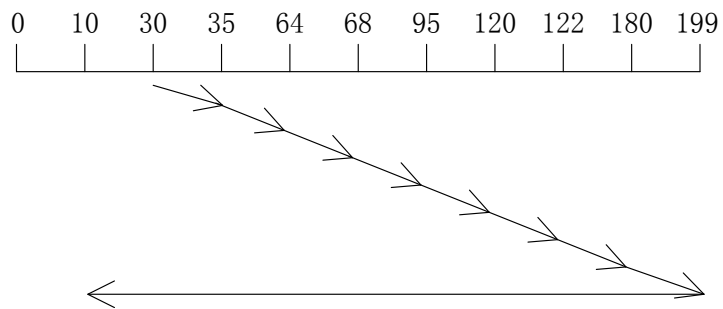
磁头移动的磁道数为: $5+25+54+4+27+35+2+58=200$ 。

SCAN: 假设磁头正向 0 方向走, 则:



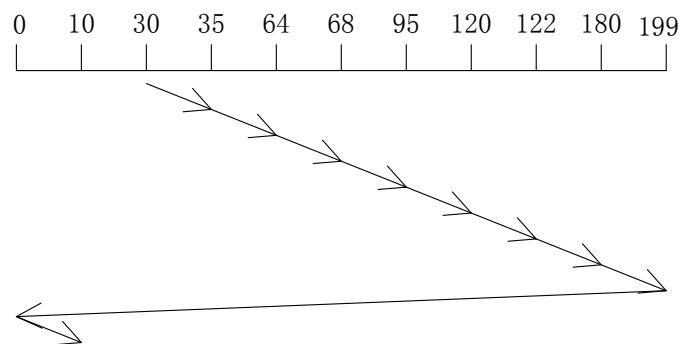
磁头移动的磁道数为： $20+10+35+29+4+27+25+2+58=210$ 。

如果磁头正向 199 方向走，则：



磁头移动的磁道数为： $5+29+4+27+25+2+58+19+189=358$ 。

C-SCAN:



磁头移动的磁道数为： $5+29+4+27+25+2+58+19+199+10=378$ 。