



计算机操作系统

6 并发程序设计 – 6.3 PV操作

6.3.3 生产者消费者问题探究

PV操作解决生产者消费者问题

PV操作解决苹果橘子问题

应用PV操作解决进程同步

生产者消费者问题

1. 一个生产者、一个消费者共享一个缓冲区
2. 一个生产者、一个消费者共享多个缓冲区
3. 多个生产者、多个消费者共享多个缓冲区
4. 多个生产者、多个消费者共享一个缓冲区
5. 多个生产者、一个消费者共享一个缓冲区
6. 一个生产者、多个消费者共享一个缓冲区
7. 多个生产者、一个消费者共享多个缓冲区
8. 一个生产者、多个消费者共享多个缓冲区

PV解决1生产者1消费者1缓冲区问题



```
Int B;
Semaphore sput;          /* 可以使用的空缓冲区数 */
Semaphore sget;          /* 缓冲区内可以使用的产品数 */
sput = 1;                /* 缓冲区内允许放入一件产品 */
sget = 0;                /* 缓冲区内没有产品 */
```

```
process producer {
    L1:
    produce a product;
    P(sput);
    B = product;
    V(sget);
    goto L1;
}
```

```
process consumer {
    L2:
    P(sget);
    product= B;
    V(sput);
    consume a product;
    goto L2;
}
```

PV解决1生产者1消费者N缓冲区问题

```
Int B[k]; // 共享缓冲区队列
Semaphore sput; // 可以使用的空缓冲区数
Semaphore sget; // 缓冲区内可以使用的产品数
sput = k; // 缓冲区内允许放入 k 件产品
sget = 0; // 缓冲区内没有产品
Int putptr, getptr; // 循环队列指针
putptr = 0; getptr = 0;
```

```
process producer_i {
    L1: produce a product;
    P(sput);
    B[putptr] = product;
    putptr = (putptr+1) mod k;
    V(sget);
    goto L1;
}
```

```
process consumer_j {
    L2: P(sget);
    product = B[getptr];
    getptr = (getptr+1) mod k;
    V(sput);
    consume a product;
    goto L2;
}
```

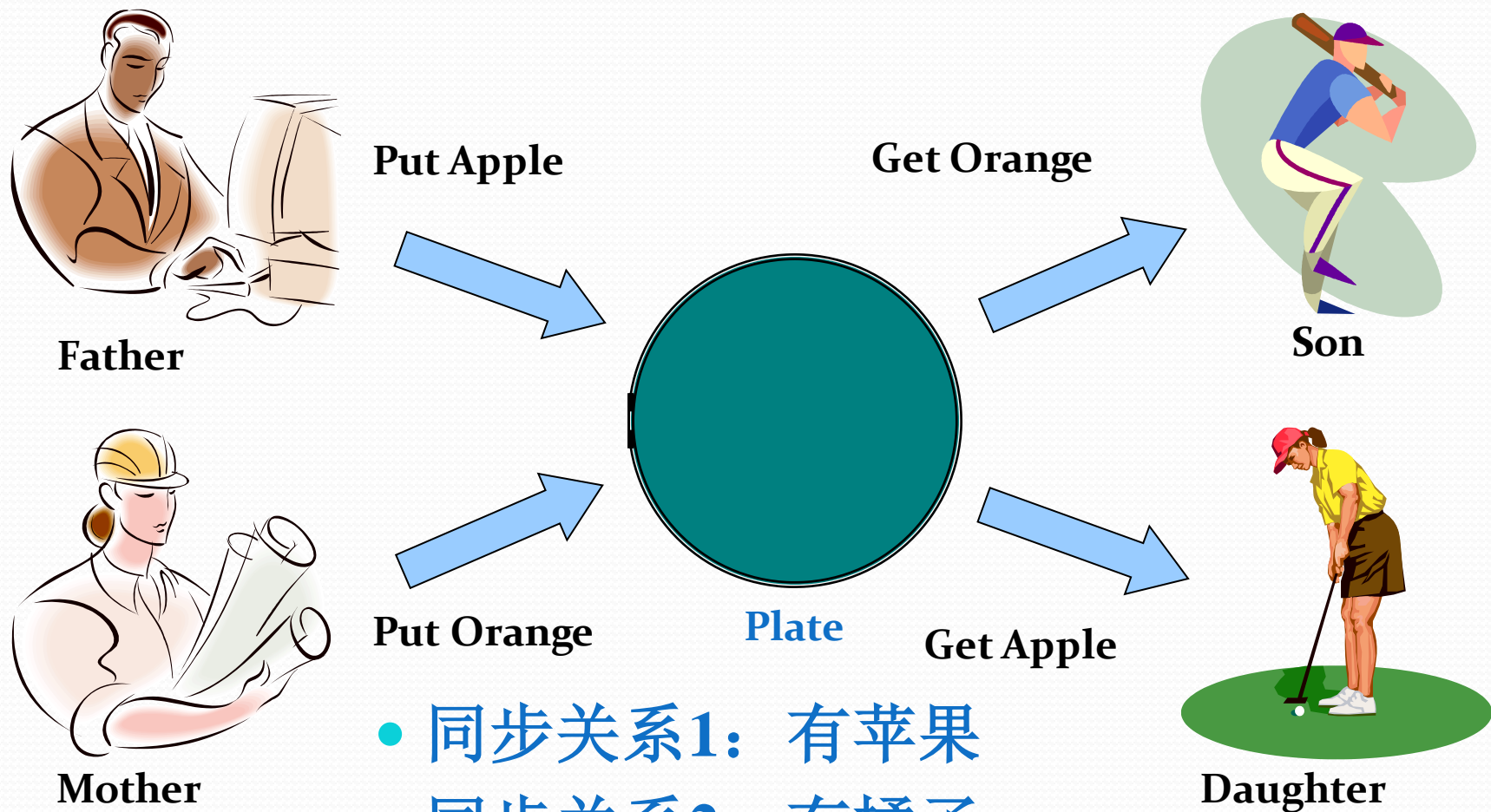
PV解决N生产者N消费者N缓冲区问题

```
Int B [k];  
Semaphore sput;          /* 可以使用的空缓冲区数 */  
Semaphore sget;          /* 缓冲区内可以使用产品数 */  
sput = k;                /* 缓冲区内允许放入 k 件产品 */  
sget = 0;                /* 缓冲区内没有产品 */  
Int putptr, getptr; putptr = 0; getptr = 0;  
s1, s2: semaphore;      /* 互斥使用 putptr, getptr */  
s1 = 1; s2 = 1;
```

```
process producer_i {  
    L1: produce a product;  
    P(sput);  
    P(s1);  
    B[putptr] = product;  
    putptr = (putptr+1) mod k;  
    V(s1);  
    V(sget);  
    goto L1:  
}
```

```
process consumer_j {  
    L2: P(sget);  
    P(s2);  
    Product = B[getptr];  
    getptr = (getptr+1) mod k;  
    V(s2);  
    V(sput);  
    consume a product;  
    goto L2:  
}
```

苹果橘子问题



- 同步关系1: 有苹果
- 同步关系2: 有橘子
- 同步关系3: 有空位

PV 操作 解决 苹果 橘子 问题

```
Int plate;
Semaphore sp;          /* 盘子里可以放几个水果 */
Semaphore sg1;          /* 盘子里有桔子 */
Semaphore sg2;          /* 盘子里有苹果 */
sp = 1;                 /* 盘子里允许放入一个水果 */
sg1 = 0;                /* 盘子里没有桔子 */
sg2 = 0;                /* 盘子里没有苹果 */
```

```
process father {
    L1: 削一个苹果;
    P(sp);
    把苹果放入plate;
    V(sg2);
    goto L1;
}
```

```
process mother {
    L2: 剥一个桔子;
    P(sp);
    把桔子放入plate;
    V(sg1);
    goto L2;
}
```

```
process son {
    L3: P(sg1);
    从plate中取桔子;
    V(sp);
    吃桔子;
    goto L3;
}
```

```
process daughter {
    L4: P(sg2);
    从plate中取苹果;
    V(sp);
    吃苹果;
    goto L4;
}
```