南京大学
NANJING UNIVERSITY

计算机操作系统

6 并发程序设计 – 6.3 PV操作

6.3.2 PV操作与进程同步

理解PV操作解决进程同步的方法

研习PV操作解决进程同步的简单例

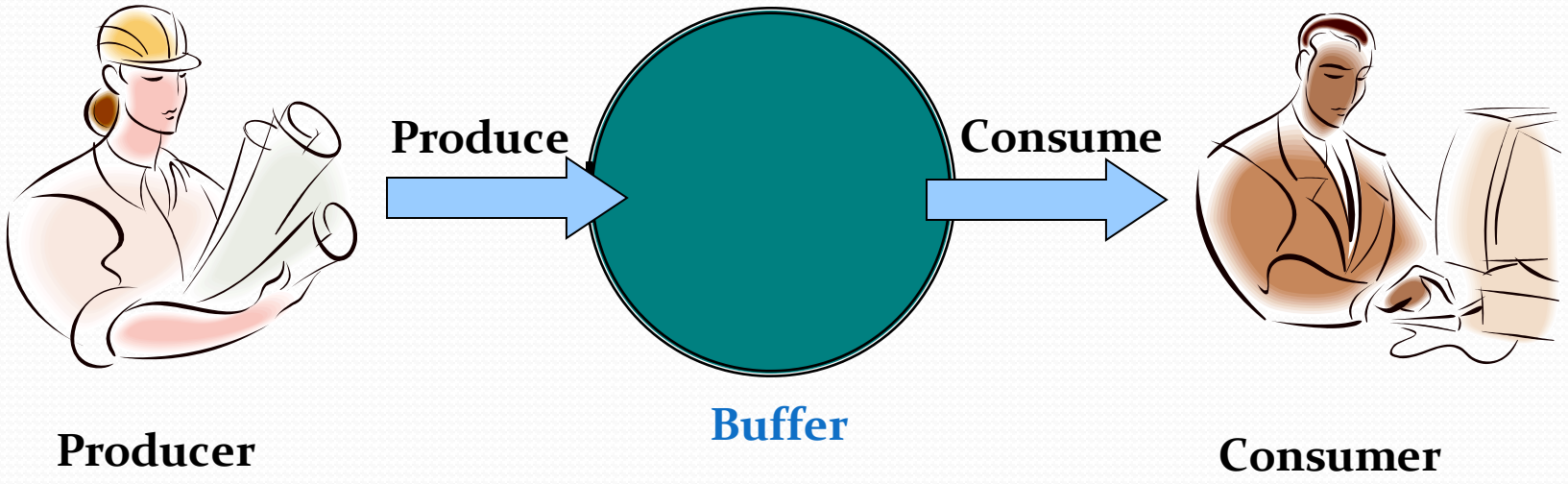
PV操作解决进程同步问题

- 进程同步：并发进程为完成共同任务基于某个条件来协调执行先后关系而产生的协作制约关系
- 一个进程的执行等待来自于其他进程的消息

- 解决的基本思路
  - 定义一个信号量：其数值代表可用消息数
  - 等待消息进程：执行P，无消息则等待
  - 发出消息进程：执行V，有等待进程则释放

1生产者1消费者1缓冲区问题

- 生产者和消费者共享缓冲区
- 缓冲区有空位时，生产者可放入产品，否则等待
- 缓冲区有产品时，消费者可取出产品，否则等待



程序框架

Int B;	

process producer
begin
L1:
produce a product;
B = product;
goto L1;
end;
	process consumer
	begin
	L2:
	product = B;
	consume a product;
	goto L2;
	end;

正确执行次序： P—C— P—C—P—C—…

信号量仅仅解决信号传递

数据传送需要共享缓冲区

同步关系2：等待缓冲

同步关系1：等待产品

解决思路

- 同步关系1：消费者一开始在等待产品到来，考虑设置一个信号量（等待产品）；一开始无产品，初值为0
- 同步关系2：消费者则在等待缓冲区中有空位，也可设置一个信号量（等待缓冲区）；一开始缓冲区有空位，初值为1

PV解决1生产者1消费者1缓冲区问题

```
Int B;                          // 共享缓冲区
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;
}
```