1. 假定某小学有600人，只有一个阅览室可供100个人同时阅读。读者进入和离开阅览室时都必须在阅览室入口处的一个登记表上登记，阅览室有100个座位，规定每次只允许一个人登记或注销登记。

要求：(1)用P、V操作描述读者进程间关系的算法.

(2)指出算法中所用信号量的名称、作用、初值及变化范围。

参考：（1）用PV操作描述读者进程的同步算法如下：

semaphore mutex=1, seat=100;

parbigin

process readeri (i=1, 2, 3, …)

begin

读者来到阅览室;

P(seat); //看看阅览室是否有空位置

P(mutex); //等级需互斥

在登记表上登记;

V(mutex);

在阅览室阅览资料;

P(mutex);

在登记表上注销;

V(mutex);

V(seat); //阅览室空座位数增1，必要时唤醒等待进入阅览室的读者

该读者离开;

end

parend

（2）互斥信号量mutex，其初值为1，用于互斥使用登记表；资源信号量seat，其初值为100，用于表示阅览室空座位的数量。Mutex变化范围为[-99，1]，seat变化范围为[-500，100]

1. 三个饮料厂P1、P2、P3都要生产橙汁，他们各自已购得三种必需原料（水、糖、浓缩汁）中的两种，待购得第三种原料后即可配制出售。有一供应商能够不断供应这些原料，但每次只能拿出一种原料放入容器出售，当容器中有原料时需要该原料的饮料厂可取走，容器空时，供应商又可放入一种原料，假定：P1已有糖和水，P2已有水和浓缩汁，P3已有糖和浓缩汁，试用PV操作写出供应商和三个饮料厂之间的同步进程

参考解答：

|  |  |
| --- | --- |
| semaphore S=1, SO=0, SS=0, SW=0; //容器是否可用, 容器中是浓缩汁/糖/水  enum { sugar, water, orange } container; | |
| cobegin  process Provider {  while(true){  P(S);  将原料装入容器内;  if (cantainer==orange) V(SO);  else if (cantainer==sugar) V(SS);  else V(SW);  }  } | process P2 {  while(true){  P(SS);  从容器中取糖；  V(S);  生产橙汁;  }  } |
| process P1 {  while(true){  P(SO);  从容器中取浓缩汁；  V(S);  生产橙汁;  }  }  coend | process P3 {  while(true){  P(SW);  从容器中取水；  V(S);  生产橙汁;  }  } |

1. 有一座东西方向的独木桥；用P,V操作实现：

（1）每次只允许一个人过桥；

（2）当独木桥上有行人时，同方向的行人可以同时过桥，相反方向的人必须等待。

（3） 独木桥上有自东向西的行人时，同方向的行人可以同时过桥，从西向东的方向，只允许一个人单独过桥。（此问题和读者与写者问题相同,东向西的为读者，西向东的为写者）。  
（1）解:

设信号量 MUTEX=1  
Cobegin

Traveler\_i(){

P (MUTEX)

过桥

V (MUTEX)

}

Coend

(2)解:

设信号量： MUTEX=1 (东西方互斥)

MD=1 (东向西使用计数变量互斥)

MX=1 (西向东使用计数变量互斥)

设整型变量： CD=0 (东向西的已上桥人数)

CX=0 (西向东的已上桥人数)

Cobegin

Traveler\_j(){//从东向西

P (MD)  
IF (CD=0) P (MUTEX)

.CD=CD+1  
V (MD)  
过桥  
P (MD)  
CD=CD-1  
IF (CD=0) V (MUTEX)   
V (MD)

}

Traveler\_j(){//从西向东  
P (MX)  
IF (CX=0). .P (MUTEX)..

. CX=CX+1  
V (MX)  
过桥  
P (MX)  
CX=CX-1  
IF (CX=0) .V (MUTEX)   
V (MX)

}

Coend

(3) 解：从东向西的，和（2）相同；从西向东的和（1）相同。