1.**什么是临界资源、临界区，临界区的使用原则有哪些？**

必须互斥使用的资源为临界资源，

访问临界资源的那段代码称为临界区。

临界区的使用原则：

空闲让进：如临界区空闲，则有进程申请就立即进入。

忙则等待：每次只允许一个进程处于临界区。

有限等待：保证进程在有限时间内能进入临界区(不会死锁或饥饿)。

让权等待：进程在临界区不能长时间阻塞等待某事件。（不能进入临界区时释放处理器）

**2.简述信号量的含义及作用。**

（1）含义：两个或多个进程可以通过传递信号进行合作，迫使进程在某个位置暂时停止执行（阻塞），直到它收到一个可以“向前推进”的信号（被唤醒），将实现信号灯作用的变量称为信号量。

信号量常定义为记录型变量，包含两个域：一个整型域表示信号量的值，另一个域为队列其元素为等待该信号量的阻塞进程。信号量的物理意义

* + s.count ＞ 0,表示目前临界资源的可用数量，即还可执行wait(s)而不会阻塞的进程数。
  + 每执行一次wait(s)操作，意味着请求分配一个单位的资源。
  + s.count ≤ 0表示已无可用的临界资源，请求该资源的进程被阻塞。此时，s.count的绝对值等于该信号量阻塞队列中的等待进程数。

执行一次signal操作，就意味着释放一个单位的资源。若s.count ≤ 0，表示s.queue队列中还有被阻塞的进程，需要唤醒该队列中的第一个进程，将它转移到就绪队列中

（2）信号量作用：利用信号量可以实现进程的互斥和同步。

**3.请用P、V操作描述下列过程**



解：  
semaphore s1=1, s2=0; //s1代表是否停车，s2代表是否关门  
cobegin   
void driver(void)  
{  
 while(TRUE)

{  
 P(s2);  
 启动车辆;  
 正常行车;  
 到站停车;  
 V(s1);

}  
 }  
void conductor(void)  
 {   
 while(TRUE)

{  
 P(s1);  
 开车门;  
 上、下乘客;

售票；  
 关车门;  
 V(s2);

}  
}  
coend

**4.图书馆有N个座位，一张登记表，要求(1)阅读者进入时取得座位号，并加以登记；(2)出来时注销。请用P、V操作描述一个读者的使用过程。**

semaphore N=空座位数量, //空座位的数量

semaphore mutex=1//登记表

void reader()

{

While(true)

{

P(N); //获得座位，没有则阻塞

P(mutex)

登记

V（mutex）

阅读

P(mutex)

注销

V（mutex）

V(N)；//释放座位

｝

｝

5.有3个进程PA，PB和PC合作解决文件打印问题：(1)PA将文件记录从磁盘读入主存的缓冲区1，每执行一次读一个记录；(2)PB将缓冲区1的内容复制到缓冲区2，每执行一次复制一个记录；(3)PC将缓冲区2的内容打印出来，每执行一次打印一个记录。缓冲区的大小等于一个记录大小。请用P，V操作来保证文件的正确打印。

semaphore S1empty＝1；//初始时缓冲区A的空单元数

semaphore S1full＝0; //初始时缓冲区A的有数据的单元数

semaphore S2empty＝1；//初始时缓冲区B的空单元数

semaphore S2full＝0；//初始时缓冲区B的有数据单元数

Process PA

{

While(true)

{

读一个数据；

P(S1empty)；

写入缓冲1；

V(S1full)；

}

}

Process PB

{

While(true)

{

P(S1full)；

从缓冲1中取数据；

V(S1empty)；

P(S2empty)；

写入缓冲2；

V(S2full)；

}

}

Process PC

{

While(true)

{

P(S2full)；

从缓冲2中读数据；

V(S2empty)；

打印；

}

}

Cobegin

   PA;PB;PC;

Coend.

6.过独木桥问题，独木桥最大称重为N人。

int countA=0, countB=0；

semaphore mutex=1, muteA=1, mutexB=1,count=N;

void east\_west() {

while (1) {

P(mutexA);

countA++;

if (countA==1) P(mutex);

V(mutexA);

P(count);

walk across the bridge from east to west;

V(count);

P(mutexA);

countA--;

if (countA==0) V(mutex);

V(mutexA);

}

}

void west\_east() {

while (1) {

P(mutexB);

countB++;

if (countB==1) P(mutex);

V(mutexB);

P(count);

walk across the bridge from west to east;

V(count);

P(mutexB);

countB--;

if (countB==0) V(mutex);

V(mutexB);

}

}

7. 某寺庙，有小、老和尚若干，有一水缸，由小和尚提入水缸供老和尚饮用。水缸可容10桶水，水取自同一井中。水井径窄，每次只能容一个水桶取水，水桶总数为3个。每次入、取缸水仅为1桶，且不可同时进行。试给出有关老和尚从缸取水和小和尚打水、入水的算法描述。

semaphore empty\_vat ＝10; //水缸可容纳水的桶数

semaphore full\_vat ＝0；//水缸内水数量

semaphore empty\_bucket ＝3；//水桶数量

semaphore mutex\_cell ＝ 1；//水井互斥信号量

semaphore mutex\_vat ＝1； //水缸互斥信号量

process 小和尚: //注意：应该先判断缸有空，再打水。如果先把水打了，再来看缸是否有空，当缸不空时，可能会导致死锁，因为此时桶不释放，老和尚无法喝水，则水缸不会空

{

repeat

wait(empty\_vat); //判断缸有空

wait(empty\_bucket);//判断桶有空

wait(mutex\_cell);

从井中取水；

signal(mutex\_cell);

wait(mutex\_vat);

送水入水缸；

signal(mutex\_vat);

signal(empty\_bucket); //桶空了

signal(full\_vat); //缸增加水量

until false;

}

process 老和尚:

{

repeat

wait(full\_vat); //判断缸有水，此处不能先取桶，否则如果缸里没有水，桶不释放，则别人无法打水

wait(empty\_bucket); //判断桶有空

wait(mutex\_vat);

从缸中取水；

signal(mutex\_vat);

signal(empty\_vat); //缸有空

signal(empty\_bucket); //桶有空

until false;

｝

8. 有一水果盘，最多盛放10个水果，爸爸负责向盘中放苹果，妈妈负责向盘中放桔子，均每次放入1个，**要求任意时刻已放入盘中苹果、桔子数量之差不能超过2（包含已被女儿取走的水果）**，女儿每次随机取出1个水果并消费。放入者和取出者不允许同时使用盘子。请用信号量机制实现女儿、爸爸和妈妈之间的同步与互斥活动，并说明所定义信号量的含义。

semaphore mutex = 1;//盘子操作互斥信号量

semaphore ao = 2, oa = 2;//苹果、桔子数量之差

semaphore empty = 10;//盘子中可放入的水果数目

Dad()

{

while (true) {

prepare\_apple();//准备苹果

P(ao);//根据水果之差，判断可否放入

P(empty);//盘子中可放入的水果数目减1

P(mutex);//互斥访问盘子

put an apple on the plate;//将一个苹果放入盘子

V(mutex);//恢复访问盘子

V(oa);//可再多放入一个桔子

V(full);//盘子中水果数量加1

}

}

Mom()

{

while (true){

prepare\_orange();//准备桔子

P(oa);//根据水果之差，判断可否放入

P(empty);//盘子中可放入的水果数目减1

P(mutex);//互斥访问盘子

put an orange on the plate;//将一个苹果放入盘子

V(mutex);//恢复访问盘子

V(ao);//可再多放入一个苹果

V(full)；//盘子中水果数量加1

}

}

daughter(){

while(true){

P(full);//判断是否有水果可取

P(mutex);//互斥访问盘子

get a fruit from plate;//随机取一个水果

V(mutex);//恢复访问盘子

V(empty);//盘子中可放入的水果数目加1

}

}