**操作系统原理**

**资源共享课程——网上作业**

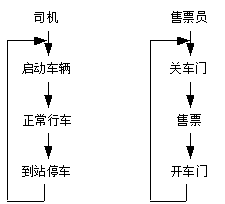
**专业班级 软件工程1505**

**学 号 17**

**姓 名 陈榕**

**Part 2 Process Synchronization Problems**

**在公共汽车上，司机和售票员的工作流程如图所示。为保证乘客的安全，司机和售票员应密切配合协调工作。请用信号量来实现司机与售票员之间的同步。**



**司机和售票员工作流程图**

解答：

思路：设置两个资源信号量car和door，car表示司机是否启动车辆，door表示售票员是否打开车门，初始值均为0，分别表示司机不启动车辆，售票员不打开车门。在工作运行的过程中，只有售票员关闭车门，司机才能启动车辆，另外，只有司机停车不启动，售票员才能打开车门，其间的过程司机和售票员互不干扰。

用两个函数分别表示司机进程和售票员进程。

代码如下：

semaphore car = door = 1; //定义两个信号量并初始化为1

void Driver{

while(1){

wait(car); //司机使用启动车辆资源，car信号量-1

//do:启动车辆过程

//do:正常行车过程

//do:到站停车过程

signal(door); //售票员释放开门资源，door信号量+1

}

}

void Salesman{

while(1){

//do:关闭车门过程

signal(car); //司机可以使用启动车辆资源，car信号量+1

//do:售票过程

wait(door); //售票员使用开门资源，door信号量-1

//do:开车门过程

}

}

void main{

re-begin{

Driver();

Salesman();

}

}

**桌子上有一只盘子，盘子中只能放一只水果。爸爸专向盘子中放苹果，妈妈专向盘子中放橘子，一个儿子专等吃盘子中的橘子，一个女儿专等吃盘子中的苹果。用wait()和signal()信号量操作实现他们之间的同步机制。**

解答：

1. 分析：爸爸和妈妈放水果的时候需要互斥，只能保证同一时间只有一个人能进行放水果操作；爸爸和女儿对盘子苹果的操作需要互斥，只能在爸爸放了苹果之后女儿才能吃，只能在女儿吃了苹果之后爸爸才能继续放，妈妈和儿子对盘子中句子的操作也需要互斥，互斥的方式和爸爸女儿的一样，因此需要三个信号量，分别用于表示盘子、苹果和橘子的互斥。用四个函数分别表示四个人的操作。
2. 代码：

semaphore plate = apple = orange = 1; //定义三个信号量，初始化为1

void Father{

while（1）{

//do:准备苹果

wait(plate); //使用盘子资源，plate信号量-1

signal(apple); //苹果数量增加1，信号量+1

}

}

void Mother{

while（1）{

//do:准备橘子

wait(plate); //使用盘子资源，plate信号量-1

signal(orange); //橘子数量增加1，信号量+1

}

}

void Daughter{

while{

wait(apple); //使用苹果资源，信号量-1

//do：吃苹果

signal(plate); //归还盘子资源，信号量+1

}

}

void Son{

while{

wait(orange); //使用橘子资源，信号量-1

//do：吃橘子

signal(plate); //归还盘子资源，信号量+1

}

}

void main{

re-begin{

Father();

Mother();

Daughter();

Son();

}

}

**5、有一只铁笼子，每次只能放入一只动物，猎手向笼子里放入老虎，农民向笼子里放入猪；动物园等待取笼子里的老虎，饭店等待取笼子里的猪。现请用wait()和signal()操作写出能同步执行的程序。**

解答：

1. 分析：需要定义三个互斥信号量，分别表示对笼子、老虎、猪的操作。猎手和农民放动物的时候需要互斥，动物园和饭点取动物的时候需要互斥，每次只能放一只动物，笼子是共享资源。分别用四个函数表示猎手、农民、动物园、饭点的操作。
2. 代码：

semaphore cage = tiger = pig = 1; //定义三个信号量，初始化为1

void hunter{

while(1){

wait(cage); //猎人申请使用笼子资源，信号量-1

//do：将老虎放入笼中

signal(tiger); //老虎数量增加，信号量+1

}

}

void farmer{

while(1){

wait(cage); //农民申请使用笼子资源，信号量-1

//do：将猪放入笼中

signal(pig); //猪数量增加，信号量+1

}

}

void zoo{

while(1){

wait(tiger); //使用老虎资源，信号量-1

//do：从笼中取出老虎

signal(cage); //归还笼子资源，信号量+1

}

}

void restaurant{

while(1){

wait(pig); //使用猪资源，信号量-1

//do：从笼中取出猪

signal(cage); //归还笼子资源，信号量+1

}

}

void main{

re-begin{

hunter();

farmer();

zoo();

restaurant();

}

}

**10、生产围棋的工人不小心把相等数量的黑子和白子混装在一个盒子里，现在要用自动分拣系统把黑子和白子分开，该系统由两个并发执行的进程PA和PB组成，系统功能如下：**

**(1) PA专拣黑子，PB专拣白子；**

**(2) 每个进程每次只拣一个子，当一个进程拣子时，不允许另一个进程去拣子；**

**(3) 当一个进程拣一个子（黑或白）后，必须让另一个进程去拣一个子（白或黑）；**

**请回答：(1) 这两个并发进程之间的关系是同步还是互斥；**

**(2) 写出wait()和signal()操作管理时应定义的信号量及其初值；**

**(3) 根据定义的信号量，写出用wait()和signal()操作管理两个并发进程的程序。**

解答：

（1）互斥的。当PA进程发生时，不允许PB同时发生，反过来也一样。

（2）定义一个信号量mutex，用于表示两个进程的互斥操作，初始值为1

（3）分析：一个信号量控制两个进程的互斥，相同时间段只允许一个进程执行。

代码：

semaphore mutex = 1; //定义信号量，并初始化为1

void PA{

while（1）{

wait（mutex）； //申请使用mutex资源，信号量-1

//do:拣黑子过程

signal(mutex); //完成进程，归还资源，信号量+1

}

}

void PB{

while（1）{

wait（mutex）； //申请使用mutex资源，信号量-1

//do:拣白子过程

signal(mutex); //完成进程，归还资源，信号量+1

}

}

void main{

re-begin{

PA();

PB();

}

}