教育部与高通公司产学合作协同育人项目

燕山大学讨论课
研究报告

课程名称：计算机操作系统

主题：操作系统演化过程中的
理论创新、技术创新和实现创新

本组主题:感受进程的创新及操作系统未来构想

小组名称：01 小组
小组组长：乔翱
小组成员：

姓名	学号	班级	分工	自评成绩
乔翱	201811040809	18 软工-6	进程同步互斥、信号量、操作系统未来畅想部分的报告书及 PPT 撰写	9
王紫晔	201811040810	18 软工-6	进程的描述、状态及其转换、进程控制部分报告书及 PPT 撰写	7

指导教师：申利民

2020 年 12 月 10 日

目录

1 进程描述

1.1 进程的引入及定义

在早期无操作系统和单道批处理系统计算机中，程序在系统中以顺序的方式执行，后进入内存的程序需要等待先进入内存的程序执行完毕才会分配 CPU 资源。在这种情况下，当某个程序因其他资源等待，一直占用 CPU 资源，其他程序将会一直处于等待状态，会导致系统的运行效率、资源的利用率以及 CPU 的使用效率比较低。

为提高资源的利用率提出了“并发执行”的概念，但是并发执行会使程序失去顺序性、封闭性及可再现性，所以通常情况下程序不能并发执行。 之后引入了进程的概念，使用 PCB 来控制、管理进程，使各个程序并发独立的运行。在后来的多道批处理系统和分时处理系统中，内存中可以同时驻足多个程序，程序在系统中并发执行。并发执行表示多个程序可以在同一个时间间隔中执行，多个程序在系统中同时驻足，提高了 CPU 的使用率、资源利用率及系统的运行效率。

进程是程序在系统中执行的一个实例，是程序在一个数据集合上运行的动态过程。进程可以提高 CPU 的利用率、使程序在系统中并发执行并可以对并发执行的程序进行描述和控制，由程序、数据、程序控制块三部分组成。

1.2 进程的特征

动态性：进程的实质是进程实体的执行过程，是动态的且有一定生命周期。由创建产生，由调度执行，由撤销消亡。当点击某一 APP 时，系统会动态创建进程并对相应的操作进行调度，当退出时会将此进程撤销。

并发性：多个进程存在内存中且能在一段时间内同时执行。就比如手机或平板上面的分屏操作，可以在边看电影的同时边在微信上聊天。

独立性：进程是能独立运行、独立获得资源并独立接受调度的基本单位。每次进程的运行、调度、获得资源都只依赖于自身。

异步性：进程以不可预知的速度向前推进，按异步方式运行，进程的运行不是一气呵成的，是走走停停的。比如当有打印命令时创建打印进程，如果当前打印机正在打印，则此进程要进行等待直到打印机空闲时调度这一进程。当点击打印机上的暂停按钮时，进程会暂停，当继续时进程继续执行。

1.3 进程控制块（Process Control Block,PCB）

PCB 是为了描述和管理进程的运行而引入的，可以使一个在多道程序环境下不能独立运行的程序成为一个能独立运行的、能与其他程序并发执行的进程。PCB 随进程的创建而产生，随进程的终止而消失。当新建一个进程时，系统分配资源及 PCB 给它。而当其完成了特定的任务后，系统收回这个进程所占的资源和取消该进程的 PCB 就撤消了该进程。PCB 可以实现间断性运行方式、提供进程管理及调度所需要的信息以及实现与其它进程的同步与通信。PCB 中包含的信息有：进程标识符、处理机状态、进程调度信息、进程控制信息等。

每个进程的进程标识符是唯一的，由操作系统和用户访问该进程时使用。PCB 的间断性运行方式主要依靠处理机状态的支持，处理机状态中保存了处理机运行时的信息，当处理机被中断时，所有保存在 PCB 中的这些信息方便了该进程在重新执行时，能从断点继续执行。进程要根据进程状态、优先级、进程调度所需的其它信息以及事件等来确定进程的调度。进程根据地址、进程同步和通信机制以及链接方式来进行进程中信息的控制。

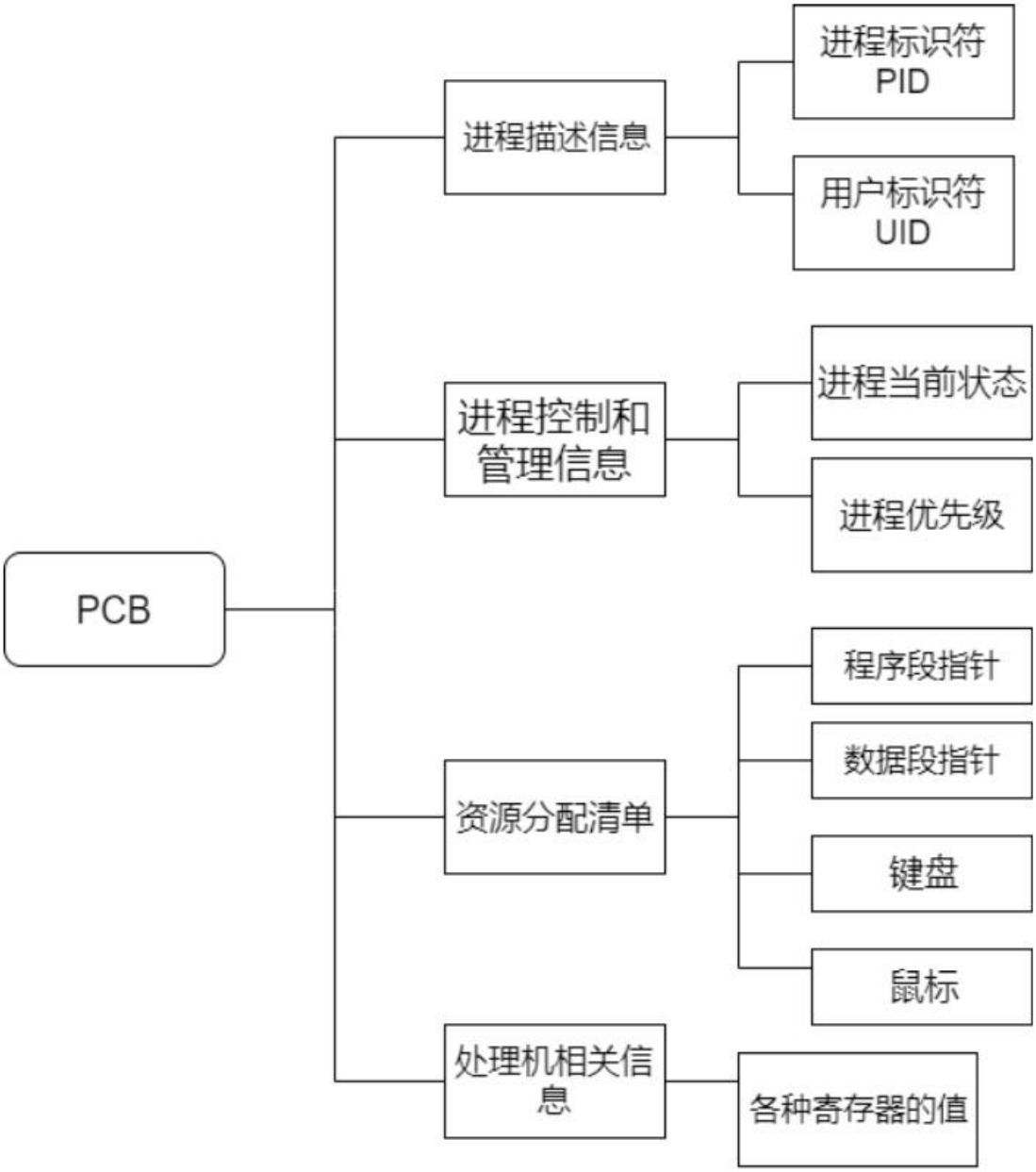

图 1.3.1 PCB 中的信息

2 进程状态及转换

2.1 进程的状态

进程的状态主要有就绪状态（活动就绪状态、静止就绪状态）、执行状态、阻塞状态（静止阻塞状态、活动阻塞状态）、创建状态、终止状态。

处于就绪状态的进程可以被调度，其中静止就绪状态是处于挂起状态的，需要等到被激活之后被调度，动态就绪状态可直接被调度。

执行状态的进程是就绪状态的进程被调度之后进入的状态，此时程序获得 CPU 正在执行。

阻塞状态的进程是进程在执行的过程中发生事件不能继续执行，此程序进入阻塞状态并将资源分配给其他进程，当被唤醒后进入就绪状态，之后才可以继续执行。

创建状态可以保证程序的调度是在创建工作完成后进行的，增强了管理的灵活性和控制完整性。

终止状态即进程的结束，不能再执行。

2.2 状态之间的转换

通过调度可以使进程由活动就绪状态转换到执行状态；通过进程的创建过程可以使进程由创建状态转换到活动就绪状态或静止就绪状态；通过挂起可以使进程由活动就绪状态转换成静止就绪状态、由活动阻塞状态转换成静止阻塞状态、由执行状态转换成静止就绪状态；

通过激活可以使静止阻塞状态转换成活动阻塞状态、由静止就绪状态转换成活动就绪状态；通过释放可以使进程由活动阻塞状态转换成活动就绪状态、由执行状态转换成终止状态；若时间片完会使进程由活动就绪状态转换为执行状态；若请求 I/O 未请求到，会使进程由执行状态转换成活动阻塞状态。

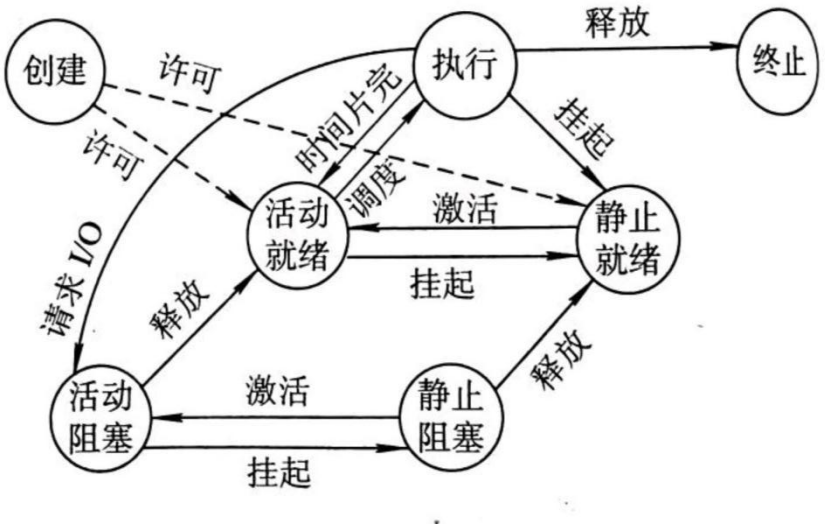

图 2.2.1 进程状态转换图

3 进程控制

3.1 进程创建与撤销

进程的创建状态：①进程申请一个空白 PCB，并向 PCB 中填写用于控制和管理进程的信息②为进程分配运行时所必需的资源③将进程转入就绪状态并转入就绪队列。

进程的终止状态：①等待操作系统进行善后处理②将 PCB 清零并将空白 PCB 返回系统

只有当进程创建工作完成后才能进行进程的调度，也就相当于在未创建完成时锁一直是锁着的，之后完成进程的创建->就绪状态，在进程创建完成后将锁打开。锁打开时才可以进行进程的调度，创建状态确保对进程控制的完整性以及增强管理的灵活性。如果没有进程创建及撤销控制的话，很可能进程在创建的工程中还没有完全创建完毕就进行了调度，这会导致程序运行的失败，浪费大量时间。

3.2 进程阻塞与唤醒

进程的阻塞与唤醒状态也类似锁操作法，阻塞之后就像是将锁锁上，不能对该进程进行调度，唤醒之后才打开锁进入就绪状态。进程的阻塞是自发的行为，阻塞原因有：1.向系统请求共享资源失败 2.等待某种操作的完成 3.新数据尚未到达 4.等待新任务的到达。 如果没有进程阻塞与唤醒控制，很可能当数据还没有到达或资源不到位的情况未被发觉而导致程序的继续执行，之后会导致程序运行失败，浪费时间及系

统资源。

3.3 进程挂起与激活

进程的挂起与激活状态也类似锁操作法。挂起之后就相当于将锁锁上，激活之后才打开锁进入就绪状态。进程的挂起状态：使进程处于静止状态，不再参与资源的竞争，直到挂起解除为止挂起原因有：硬件故障；调试程序；解决进程过多、资源消耗大的问题。如果没有进程挂起与激活控制，很可能当硬件出现故障的情况未被发觉而导致程序的继续执行，之后会导致程序运行失败，浪费时间及系统资源最终也很难发现问题所在。

3.4 进程控制的实例

在智能家居系统中，很可能因为向系统请求共享资源失败、等待某种操作完成而导致阻塞，比如当你回到家后，每间卧室的开灯进程都会创建、就绪，只有当你踩过地板，地板上的触感器感知到之后进入唤醒状态，才对于开灯进程进行调度，这在一定程度上既方便又节约资源。阻塞就像给进程上了一把锁，唤醒是它的钥匙。

就比如在智能管家系统中，当你工作一天拖着疲惫的身体回到家打开门，打开门后，系统自动创建客厅开灯的进程、热水器放合适温度热水的进程以及语音识别创建进程等，进程创建好后进程进入就绪状态，之后进程进入执行状态，虽然在这里写的此过程较于繁琐，但系统的速度还是比较快的，所以对于来人类说这个速度是适宜的。

开灯进程：当你即将要进入哪一个房间时，地板上的传感器感知到以后为该房间创建开灯进程->就绪->执行，当你离开时，地板上的传感器也同样感知到之后将进程终止，灯关闭。

放热水进程：当你刚刚回到家就会创建放热水进程并进入就绪、执行，里面的温度传感器会根据温度的变化来进行放热水的进程的调度，当你要离开时放热水进程终止。

语音识别创建进程：当你对语音助手说话时，比如你对他说定一个明天早上七点的闹钟，它会创建一个定闹钟的进程->就绪，智能管家系统会在明天早上七点执行这个进程，当你起床之后，该进程终止。当你对语音助手说打开电视机播放自己喜欢的纪录片，它会创建一个打开电视机并搜寻你想看的纪录片的进程，之后就绪、执行，当你不想看的时候或者睡着的时候进程终止。

在智能家居系统中，最可能出现的就是硬件故障而挂起，比如当灯的灯管出现故障需要更换灯管时，开灯这一进程创建之后进入就绪状态，之后不可以进入执行状态、直接进入挂起状态，不可以对此进程进行调度，直至更换灯管硬件设备正常之后调用激活状态将挂起解除，之后才可对开灯进程进行调度。当进程处于挂起状态的过程中是不能够对该进程进行调度的，只有将其激活之后才可以，就像是上了一道锁。

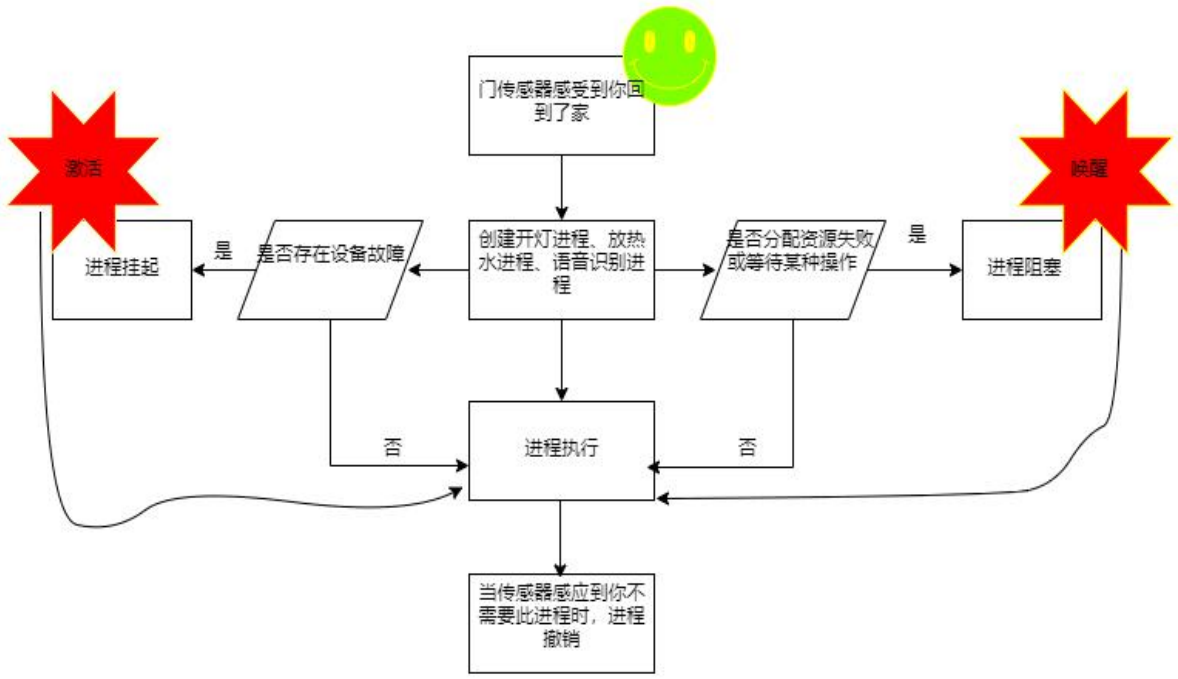

图 3.4.1 智能管家进程控制简单示意图

4 进程同步与互斥

4.1 进程间的制约关系

在多道程序环境下，系统中各进程以不可预测的速度向前推进，进程的异步性会给系统造成混乱，造成了结果的不可再现性。为防止这种现象，异步的进程间推进受到两种限制：

（1）资源共享关系

多进程共享资源，例如各进程争用一台计算机，这时各进程使用这台打印机时有一定的限制。如各进程随意使用打印机，会造成打印机结果交织在一起难以区分。所以必须由系统统一分配，每次只允许一个进程使用一段时间打印机，等该进程使用完毕后再将打印机分配给其它进程。这种使用原则称为互斥使用。

（2）相互合作关系

在某些进程之间还存在合作关系，例如并发执行一个程序的输入、计算、打印三个程序段作为三个进程并发执行，由于这三个进程间存在着相互合作的关系，即先输入再计算、最后再打印的关系，所以这三个进程在并发执行时推进序列受到限制，要保证其合作关系正确，进程间这种关系称为同步关系。

4.2 进程同步与互斥的定义

同步：异步环境下的一组并发进程因直接制约而互相发送消息、进行互相合作、互相等待，使得各进程按一定的速度执行的过程称为进程间的同步。具有同步关系的一组并发进程称为合作进程，合作进程间互相发送的信号称为消息或事件。

互斥：两个或两个以上的进程，不能同时进入关于同一组共享变量的临界区域，否则可能发生与时间有关的错误，这种现象被称作进程互斥。也就是说，一个进程正在访问临界资源，另一个要访问该资源的进程必须等待。

4.3 同步机制应遵循的规则

1.空闲让进：当临界资源处于空闲状态，允许一个请求进入临界区的进程立即进入临界区，从而有效的利用资源。

2.忙则等待：已经有进程进入临界区时，意味着相应的临界资源正在被访问，所以其他准备进入临界区的进程必须等待，来保证多进程互斥。

3.有限等待：对要求访问临界资源的进程，应该保证该进程能在有效的时间内进入临界区，防止死等状态。

4.让权等待：当进程不能进入临界区，应该立即释放处理机，防止进程忙等待。

5 信号量机制

5.1 信号量机制的概念

信号量是一个确定的二元组(S,Q),其中 S 是一个具有非负初值的整形变量,且 S 的值只能由定义在信号量上的 P 操作原语和 V 操作原语来改变,而 Q 是个初始状态为空的队列,即信号量表示某类资源实体与进程队列有关的整形变量。

通常,S 用来表示系统中资源的使用情况。当 S> 0 时,其值表示系统中对应可用资源的数目;当 S=0 时,表示系统中对应资源已经都被占用,并且没有因该类资源而被阻塞的进程;当 S<0 时,其绝对值表示因该类资源而被阻塞的进程数目。

5.2 P、V 操作

P(S):执行一次 P 操作意味着分配一个单位资源，因此 S 的值减一，若 S>=0，其值代表了可利用资源的数目，则表示执行 P 操作的进程可以申请到相应资源，继续执行；若 S<0,表示已没有资源可用，请求者必须等待其他进程释放该类资源之后才能继续运行，则置该进程为阻塞状态，并将其插入阻塞队列。

P 操作原语执行的操作：

```
void wait(semaphore s)
{
 s.value=s.value-1;
 if(s.value<0)
   block(s.queue);
}
```

P 操作的流程图如下图所示：

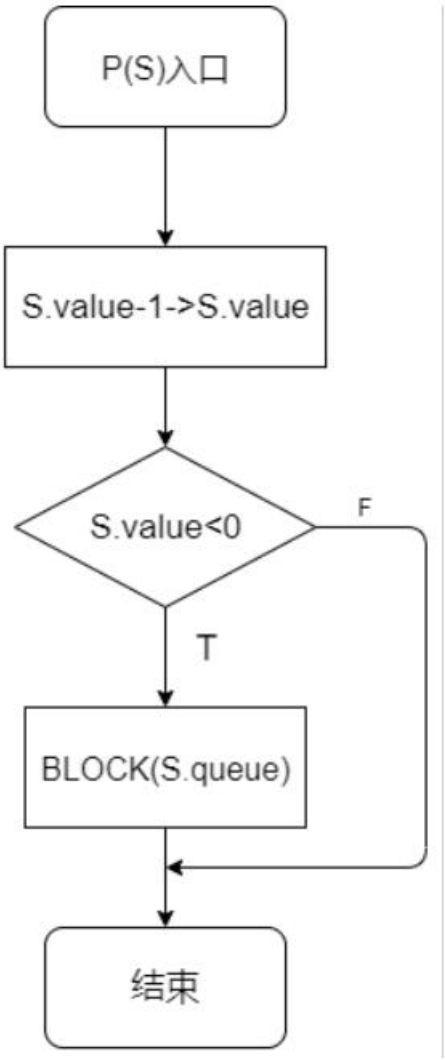

图 5.2.1 P 操作流程图

V（S）：执行一个 V 操作意味着释放一个单位资源，因此 S 的值加 1，若 S>0，则执行 V 操作的进程继续执行；若 S<=0，则从阻塞队列唤醒一个进程，并将其插入就绪队列，然后执行 V 操作的进程继续执行。

V 操作执行的原语的操作：

```
void signal(semaphore s)
{
 s.value=s.value+1;
 if(s.value<=0)
   wakeup(s.queue);
}
```

V 操作的流程图：

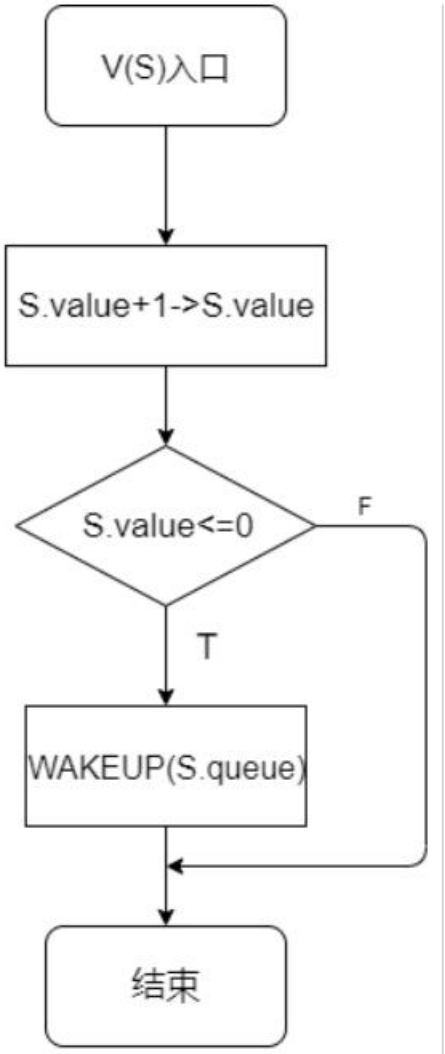

图 5.2.2 V 操作流程图

通常 P 操作意味着请求一个资源，V 操作意味着释放一个资源。在一定条件下，P 操作代表挂起进程操作，而 V 操作代表唤醒挂起进程的操作。

5.3 用 P、V 操作原语实现进程的同步和互斥

5.3.1 同步问题的分析方法

进程同步指两个或多个进程为了合作完成同一任务,在执行速度或某个确定的时序点上必须相互协调,即一个进程依赖于另一个进程发送的消息 ,进程之间体现了一种相互合作的关系。当用 P、V 操作实现进程的同步问题时,合作进程之间通过信号量进行互发消息 ,执行 P 操作测试消息是否到达 ,即是否得到合作进程的通知可以执行该进程;而执行 V 操作是向其合作进程发送消息,通知它进行下一步操作。 通常 ,需要收发几条消息就设置几个信号量,这些同步信号量的初值一般为 0,有时也可设为一个整数 ,这往往与合作进程所共享资源的可用数目有关。在同步关系的进程中,对同一个信号量的 P、V 操作不出现在同一个进程中,而是出现在各个合作进程中确定的时序点,即当一个生产者进程在完成了前面的生产任务后,应立即给消费者进程发送一条消息(执行 V 操作),而消费者进程在消费前要执行对同一信号量的 P 操作。

一般来讲,解决进程同步问题的主要步骤为:

(1)分析清楚题目涉及的进程间的制约关系;

(2)设置信号量，设置信号量的个数和初值及其意义;

(3)给出进程相应程序算法描述或流程控制,并把 P、V 操作加到程序的适当处。

5.3.2 互斥问题的分析方法

进程互斥指对于某个系统资源,如果一个进程正在使用,则另外一个想使用该资源的进程就必须等待 ,而不能使用。互斥进程之间体现了对资源的竞争关系,竞争的资源被称为临界资源,而进程中访问临界资源的那段代码叫做临界区。

为了实现对临界资源的互斥访问,应保证进程互斥地进入各自的临界区。用 P、V 操作实现进程的互斥,每个进程中用户实现互斥的 P、V 操作必须成对出现,先做 P 操作,进入临界区,再做 V 操作,退出临界区。P、V 操作应分别紧靠临界区的头尾部,临界区代码应尽可能短,且不能有死循环。互斥信号量的个数有临界资源的类型决定,有几类临界资源设几个信号量,初值一般设为 1,表示该临界资源未占用,且其可用数目为 1。

6 带标记的信号量——一种新型同步与互斥机制

6.1 问题引出

信号量机制是并发程序中一种基本的、重要的同步与互斥原语。运用信号量可以很方便的实现进程的同步与互斥，信号量机制描述能力强、机理简单、易于掌握，能方便地描述并发程序中的同步和互斥问题。然而，对于某些问题用信号量机制解决起来比较困难或着比较复杂。例如下面这个典型的入厕问题：

某个地方有一个公共厕所，此公共厕所里面有 N 个蹲位，但是这个公共厕所并没有将男女蹲位分隔开。因此，当厕所里有人在入厕时，会用一块指示牌标记入厕的人的性别，那么即使有空位异性也不能进去。并且，只要 N 个蹲位未被全部占满，与已在入厕者同性别的人就可以进去；而当厕所中无人入厕时，任何人都可以进去。

对于这个问题，情况比较特殊，但是此类问题在生活中也有很多，此类问题也有一定的普适性。例如：一个球场在出租打球时，规定所有入场人员必须是同一个团体的，并且该球场对于场内人数有一定限制，同一个团体进去的人数不能超过限制人数。对于其他团体，只有等球场内的所有人都出来后其他人才能进去。再比如一个饭店，可以同时分批接待若干个代表团就餐，假设所有人的用餐标准都是一样的，那么在安排餐桌的时候，假如一张餐桌无人入座，那么哪个人先入座就把这张餐桌安排给该人所属的代表团，其他人不得入座（即使餐桌没有坐满）。只有当该桌人都就餐完毕都离开后其他尚未就餐人员才能再按代表团入座就餐。此类问题的共同特点就是同一类资源在任一时刻只能分配给具有同一性质（同一性别、同一代表团、同一个身份等）的若干进程。下面针对入厕问题用普通信号量解决这类问题，并且在之后给出一种新的信号量机制——带标记的信号量来解决此类问题。

6.2 普通信号量解法及存在的问题

通过分析上述的这个入厕问题，可以发现，这个问题中存在着两对互斥关系：一个是异性互斥，当厕所已经有人在入厕的时候，即使有空的位子，异性也不能进去。二是蹲位互斥，同一性别的人入厕时因蹲位限制而要做互斥。现在利用普通信号量解决这个问题，那么就需要设置两个信号量：一个是用来控制当前可用蹲位的多值信号量 Sem，该信号量的初值应该设为总的可用蹲位数 N，另一个是用于封锁有关操作的二值信号量 Lock，该信号量的初值设为真，需要对这些操作进行一个封锁，因为这些操作不可分的。

当有人要入厕时，首先需要判断一下厕所中是否有人，假如没有人，那么任何人可入厕，此时要设置如厕者的性别标记（利用 Tag 标记入厕者的性别）；假如有人并且还有空蹲位，则只有与已在厕者同性别的人才可以入厕；如果不是这两种情况，则要入厕者都需要等待。入厕流程的操作伪代码如下所示：

```
P（Lock）
if Sem==N then
    Wait(Sem);--任何人都可入厕
    Tag:=所要入厕者的性别
elseif(Sem>0) and (Tag==所要入厕者的性别）then
     Wait(Sem);--只有与正入厕者同性别的人才可入厕
else
      所要入厕者（挂起）排队等待
end if;
V(Lock);
```

当有人入厕完出来后，要首先判断是否有人等待，如无人等待，则释放该蹲位；否则，如果这个人出来时厕所中还有人，那么此时一定还有空的蹲位，可以让正在等待的同性别的一人入厕；假如该人出来时厕所里面没有人，则让等待中的异性的一人进入，并且置其性别标记，然后再让其他人入内，直到所有的蹲位都已占满。该过程的伪代码描述如下：

```
P(Lock)；
Signal(Sem)；
    if 相应等待队列非空 then
        if Sem==N then
            等待队列中选任一人进去入厕；
            Sem：=Sem-1;
            Tag：=所要人厕者的性别；
            while(Sem>0）and（等待队列中还有人的性别与 Tag 相同）LOOP
                该人进去入厕；一一其他与正入厕者同性别的人可同时人厕
                Sem:=Sem-1；
            end LOOP；
        elseif 等待队列中有人的性别==Tag then
              该人进去人厕(同性别的人才可以入厕)；
              Sem:=Sem-1；
        end if
    end if;
V(Lock);
```

分析上述两段程序，在整个程序中调用了四个操作。P 与 V 操作是单值信号量操作，用于封锁有关的操作，因为这些操作是不可分的。而 Wait 与 Signal 操作是多值信号操作，用于申请与释放资源。在此过程中与一般的信号量机制不同的是，不允许 Sem 取负值，不允许 Sem 取负值来表示正挂起在等待队列中的进程（待入厕者）的数目，这是因为由于两种互斥的存在，即使还有资源可用（空的蹲位），但是可能有异性进程处于等待状态。所以并不是 Sem>0 就可以用 Wait 操作来使要入厕者入厕，调用该操作是有条件

的（同性的进程才可以进入）；同样，对于 Signal 操作不具备唤醒等待进程的能力，所以在使用它释放资源后必须分情况处理等待进程。

在这个问题中，由于问题的特殊性，并没有完整 的使用 Wait 和 Signal 原语，因为在申请资源或者是释放资源都需要考虑进程的某个性质（性别），信号量 Sem 只反映了资源分配情况并没有反映出进程调用（包括等待）的具体情况。更为严重的是，在用于释放资源的程序段中没有通过信号原语就直接更新了信号变量的值，这破坏了信号量操作的完整性与一致性，可能带来严重的后果，对于申请与释放资源的操作变得很复杂、很容易出错。下面引入一种带标记的信号量机制来更好的解决此问题。

6.3 带标记信号量

带标记信号量的操作原语是 Tagged_Wait 与 Tagged_Signal，分别对应于普通信号量中的 Wait 与 Signal 原语。普通信号量只有一个分量，当它为正时表示还未被占用的资源数，当它为负时表示正在等待使用该类资源的进程数，当它为零时表示所有资源都已被占用并且没有其它进程在等待该类资源；而带标记信号量 S 有三个分量，分量 S.R 表示尚可使用的资源数（初值为所有资源数 N），分量 S.P 表示正在等待使用该资源的进程数（初值为零），另一个分量 S.T 是一个标记，标记正使用这类资源的使用者的身份（性别、所属代表团等）。当所有资源均未被使用时，该标记不起作用。这三个分量的值都不得在除 Tagged_Wait 与 Tagged_Signal 这两个操作原语之外的其它地方更新。

Tagged_Wait 原语用于申请资源，该原语有两个参数：一个是 Tag，表示调用进程的调用标记；另一个是变量参数 S，是所要作用的带标记信号变量。Tagged_Wait 原语的功能描述如下：

```
Tagged_Wait(Tag, S)：：=
    if S.R==N then
        S.R=S.R-1;
        当前调用进程继续执行;
        S.T：=Tag;
    elsif (S.R>0)and（S.T=Tag) then
        S.R:=S.R-1;
        调用进程继续执行;
    else
        挂起当前调用进程到相应队列中（并相应保存 Tag），
    endif,
```

Tagged_Signal 原语用于释放资源，有一个变量参数 S，代表带标记信号变量。该原语的功能描述如下:

```
Tagged_Signal(S)：=
    if S.P>0 then
        if S.R==N—1then 一一该类资源只被当前进程占用了一个，释放完后该类资源暂无进程占用从等待队列中取一进程执行;
        S.T：=Tag；一一将 S.T 置为刚从等待队列中取出的进程的 Tag 值
        while(S.R>0）and（等待队列中还有进程的 Tag 值等于 S.T)loop
            从等待队列中取该进程执行;
            S.R：=S.R-1;
        end loop,
```

```
        elseif(S.R>0)and（等待队列中有进程的 Tag 值等于 S.T)then
              从等待队列中取该进程执行；
        endif;
    else
        S.R:=S.R+1;
    endif;
```

6.4 应用

利用带标记信号量可以很容易解决入厕问题：当要入厕时只要调用 Tagged_Wait(Sex,Sem)原语，Sex 是要入厕者的性别，Sem 是表示厕所蹲位占用情况的带标记的信号量；当入厕完时只要调用 Tagged_Signal(Sem)原语即可。

6.5 带标记信号量与普通信号量的对比

对于带标记的信号量其实是信号量的一般形式,而普通信号量则是带标记的信号量的特殊情况。当 S.R 的初值为 1 时，带标记信号量就退化成了二值互斥信号量，此时其标记分量不起作用；对任一带标记信号量，当对他的使用都用一特定“空”标记值调用 Tagged_Wait 原语时，该信号量就退化成普通多值信号量。由此可见，带标记信号量是普通信号量的一般化，普通信号量则是带标记信号量的特例，而带标记的信号量的描述能力则比普通信号量强的多。

7 对操作系统未来发展的畅想

7.1 OS 的发展趋势

7.1.1 巨型微型化

巨型化指研制速度更快的、存储量更大的和功能更强大的巨型计算机。主要应用于天文、气象、地质和核技术、航天飞机和卫星轨道计算等尖端科学技术领域，研制巨型计算机的技术水平是衡量一个国家科学技术和工业发展水平的重要标志。与此同时，计算机的微型化已成为计算机发展的重要方向，各种笔记本电脑和 PDA 的大量面世和使用，是计算机微型化的一个标志。所以，在未来一定有多种多样规模大小的系统适应人类各种不同场景的需求。

7.1.2 智能网络化

网络化可以更好地管理网上的资源，它把整个互联网虚拟成一台空前强大的一体化信息系统，犹如一台巨型机，在这个动态变化的网络环境中，实现计算资源、存储资源、数据资源、信息资源、知识资源、专家资源的全面共享，从而让用户从中享受可灵活控制的、智能的、协作式的信息服务，并获得前所未有的使用方便性和超强能力。计算机智能化是使计算机具有模拟人的感觉和思维过程的能力。智能化的研究包括模拟识别、物形分析、自然语言的生成和理解、博弈、定理自动证明、自动程序设计、专家系统、学习系统和智能机器人等等。目前已研制出多种具有人的部分智能的机器人，可以代替人在一些危险的工作岗位上工作。计算机的硬件和软件以及操作系统的密切结合，同时保持特有的使用功能，为人们的生活带来更加需求的功能。改变传统的使用功能，趋向于一种更加智能化的发展。有人预测，家庭智能化的机器人将是继 PC 机之后下一个家庭普及的信息化产品。

7.1.3 便捷高效化

这种便捷性不单单是传统意义上的使用方便，更重要的体现整个操作系统具有更方便的制造技术，未来的发展重点是在节约资源和维持整个操作系统能够更加便捷的帮助人们使用。同时，由于未来生活的发展是一个充满更高科技化的过程，整个生活讲究的是效率，因此，计算机的发展一定朝着更高效力的方面发展，同时满足人们的各种生活需要，因此，计算机的操作系统肯定朝着人性化和高性能及高效益的方向发展，从而更加有利于计算机系统的发展。

7.1.4 稳定安全化

对于使用者来说，系统的安全性与稳定性是至关重要的，在未来的操作系统发展中，系统的安全稳定性是一大重要因素。

7.1.5 多媒体化

人们通过键盘、鼠标和显示器进行对文字、数字、图形和声音等文件的处理，由于数字化技术的发展进一步改进了计算机的表现能力，多媒体技术使信息处理的对象和内容发生了深刻变化。

7.2 对未来 OS 的畅想

7.2.1 系统虚拟化

未来的操作系统可能不会存放在本地，而是每个用户的操作系统存在云端，操作系统不再依赖于个人的显示器终端。在任何一台设备上，用户只需要登录自己的账户密码或者通过人脸、指纹、声音、虹膜等安全认证，就可以切入到自身使用的系统，查看系统的相关信息，并进行一系列的操作，实现真正的“云操作系统”。对于“云操作系统”来说个人的信息保护就是特别重要的，对于数据必须有很好的保护，在未来可以通过某些的加密技术进行对于数据的保护，例如量子加密技术，使得信息在传输过程中不会被人获得，只有秘钥的两头才能获取数据。

7.2.2 系统普及化、差异化

对于未来的操作系统要能够普及推广，应该具有易于操作的特点，使得此操作系统能够满足大多数用户的需求。而对于不同的人群，可以推广出不同的操作系统，或者建立一个操作系统的商店，通过下载和删除更改操作系统的功能的配置。并且操作系统还可以是针对用户进行专门定制的，对于不同的用户定制符合用户习惯的操作系统。

7.2.3 系统智能化

随着人工智能技术的发展，AI 将会成为人们生活的一部分，操作系统中的语音助理会实现真正的智能化，以至于实现智能 AI 操作系统。并且通过 AI 芯片，系统可以处理大量计算任务，避免任务堵塞，从而更高效的响应用户的一系列操作。

7.2.4 打通移动端、pc 的隔阂，支持无缝对接

未来的操作系统适用于所有场景。可以运行在智能手机、智能扬声器、计算机、智能手表、无线耳塞、汽车与平板电脑上。实现多端共生，互为外设。当使用某些软件时，电脑可通过数据线对手机进行操作。当统一操作系统之后，不再依赖于数据线或其他关联软件，只需通过信号的收发进行操作互联。因为所有的移动设备、智能家居都使用了相同的操作系统，所以可通过移动设备连接智能家居，实现远距离操控，实现一键万物互联。

计算机操作系统讨论课评分表

小组名称	**01 小组**			**小组自评分数**	**9**
小组人员	**乔翱**	**王紫晔**			
自评分数	**9**	**7**			
讨论题目	**感受进程的创新及对未来操作系统的畅想**				
评价内容（100 分）					
评价项目	**内容、特色**				**得分**
资料的查阅整理、研究报告及任务完成　30 分					
PPT 质量和讲解及演示情况　40 分					
提问交流，回答问题　30 分					
评审人		**总分**			