



计算机操作系统

6 并发程序设计 – 6.4 管程

6.4.2 霍尔管程

理解管程实现的霍尔方法

理解霍尔管程的条件变量

理解霍尔管程的程序过程

霍尔管程的实现方法

- 使用signal释放一个等待进程时，霍尔管程让执行signal的进程等待，直到被释放进程退出管程或等待另一个条件
- 霍尔管程基于PV操作原语实现
 - Wait和signal可以是程序过程
 - 可以用语言机制实现霍尔管程

互斥调用霍尔管程的信号量

TYPE interf = RECORD

mutex : semaphore;

//调用管程过程前使用的互斥信号量

next : semaphore;

//发出signal的进程挂起自己的信号量

next_count : integer;

//在next上等待的进程数

END;

互斥调用霍尔管程的框架

P(IM.mutex);

<过程体>;

if IM.next_count > 0 then V(IM.next);

else V(IM.mutex);

霍尔管程的条件变量

x_sem : semaphore;

// 与资源相关的信号量

x_count : integer;

// 在x_sem上等待的进程数

分配资源

霍尔管程的wait过程

```
procedure wait(var x_sem : semaphore,  
               var x_count : integer, var IM : interf);  
begin  
    x_count := x_count + 1;  
    if IM.next_count > 0 then V(IM.next);  
    else V(IM.mutex);  
  
    P(x_sem);  
    x_count := x_count - 1;  
end;
```

释放

霍尔管程的signal过程

```
procedure signal(var x_sem : semaphore,  
    var x_count : integer, var IM : interf);  
begin  
    if x-count > 0 then begin  
        IM.next-count := IM.next-count + 1;  
        V(x-sem);  
        P(IM.next); // 进入等待调用管程的队列  
        IM.next-count := IM.next-count - 1;  
    end;  
end;
```