

计算机操作系统

6 并发程序设计 - 6.4 管程 6.4.1 管程概述

> 掌握管程的基本概念 理解管程与条件变量 理解管程的执行模型

管程概念的提出 □

- 管程试图抽象相关并发进程对共享变量 访问,以提供一个友善的并发程序设计 开发环境
- 管程是由若干公共变量及其说明和所有访问这些变量的过程所组成
- 管程把分散在各个进程中互斥地访问公 共变量的那些临界区集中起来管理,管 程的局部变量只能由该管程的过程存取
- 进程只能互斥地调用管程中的过程

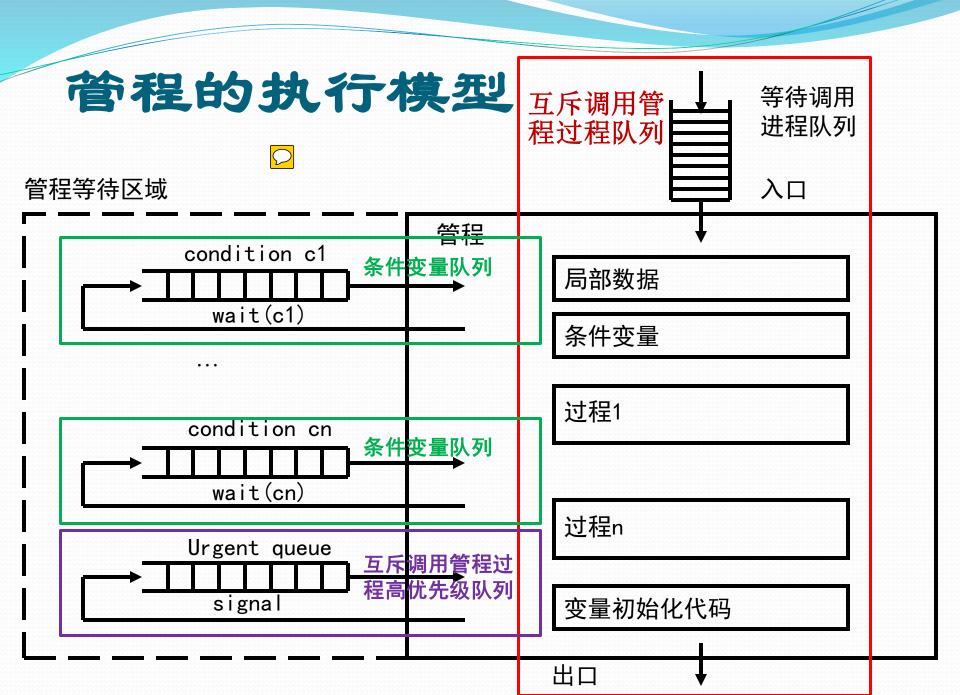
管程的基本形式

```
\bigcirc
```

```
TYPE〈管程名〉 = MONITOR
〈管程变量说明〉;
define ( (能被其他模块引用的) 过程名列表);
use ( (要引用的模块外定义的) 过程名列表 >;
procedure 〈过程名〉(〈形式参数表〉);
  begin
   〈过程体〉;
  end:
procedure 〈过程名〉(〈形式参数表〉);
  begin
   〈过程体〉;
  end;
begin
  〈管程的局部数据初始化语句〉;
end;
```

管程的条件变量

- 条件变量(condition variables): 当调用管程过程的进程无法运行时,用于阻塞进程的信号量
- 同步原语wait: 当一个管程过程发现无法继续时(如发现没有可用资源时),它在某些条件变量上执行wait,这个动作引起调用进程阻塞
- 同步原语signal: 用于释放在条件变量上 阻塞的进程



管程过程执行中signal的处理问题

- 当使用signal释放一个等待进程时,可能出现两个进程同时停留在管程内。解决方法:
 - 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
 - •被释放进程等待,直到执行signal的进程 退出管程或等待另一个条件
- 霍尔采用了第一种办法
- 汉森选择了两者的折衷,规定管程过程所执 行的signal操作是过程体的最后一个操作