第二章　进程的描述与控制

2.1 前驱图和程序执行

2.1.1 前趋图

2.1.2 程序的顺序执行

2.1.3 程序的并发执行

2.1.1 前趋图

1.前趋图的概念

前趋图：有向无循环图，记为DAG(Directed Acyclic Graph)，用于描述进程之间执行的前后关系。

结点：每个结点可用于描述一个程序段或进程，或一条语句；

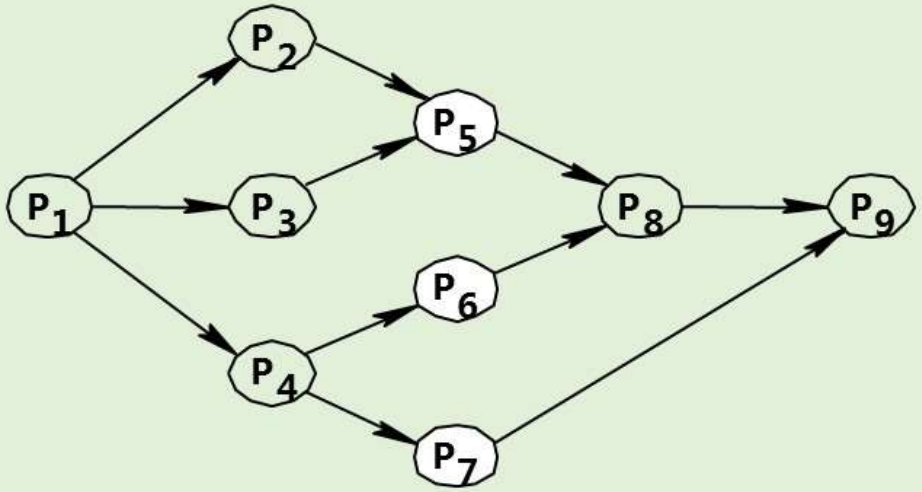
有向边：结点间的有向边则用于表示两个结点之间存在的偏序(Partial Order)或前趋关系(Precedence Relation) “→”。

→ ={(Pi, Pj) | Pi 在 Pj 开始之前完成}

直接前趋/直接后继：如果(Pi, Pj)∈→,可写成Pi→Pj，称Pi是Pj的直接前趋，而称Pj是Pi的直接后继。

初始结点/终止结点：在前趋图中，把没有前趋的结点称为初始结点(Initial Node)；把没有后继的结点称为终止结点(Final Node)。

重量(Weight)：每个结点还具有一个重量(Weight)，用于表示该结点所含有的程序量或结点的执行时间。



（a）具有九个结点的前趋图

（b）具有循环的图

前驱图中必不存在循环

2.1.2 程序顺序执行

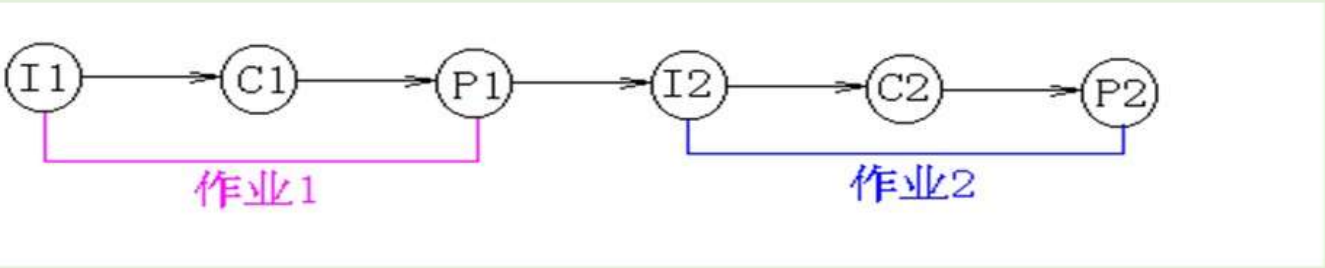
一、顺序程序及特点

1. 什么是程序的顺序执行

一个程序由若干个程序段组成，而这些程序段的执行必须按照严格的先后次序顺序的执行。程序的顺序执行也称为**顺序程序设计。**

顺序环境：

在单道计算机系统中只有一个程序在运行，这个程序独占系统中所有资源，其执行不受外界影响.



例： S1： a:=x+y;□

S2： b:=a−5;□

S3： c:=b+1;

7

2. 程序顺序执行的特点

（1）程序执行的顺序性

处理机的操作严格按照程序所规定的顺序执行。

（2）程序执行的封闭性

独占资源，执行过程中不受外界影响（由执行环境保证）

（3）程序执行结果的可再现性（确定性）

程序执行的结果与初始条件有关，而**与执行时间无关**。即只要程序的初始条件相同，它的执行结果是相同的，不论它在 什么时间执行，也不管计算机的运行速度。

（由封闭性保证）。

2.1.3 程序并发执行

1.并发环境：

在一定时间内物理机器上有两个或两个以上的程序同处于开始运行但尚未结束的状态，并且次序不是事先确定的。

2.程序并发执行（定义）

若干个程序段同时在系统中运行，这些程序的执行在时间上是重迭的，一个程序段的执行尚未结束，另一个程序段的执行已经开始，即使这种重迭是很小的，也称这几个程序段是并发执行的。

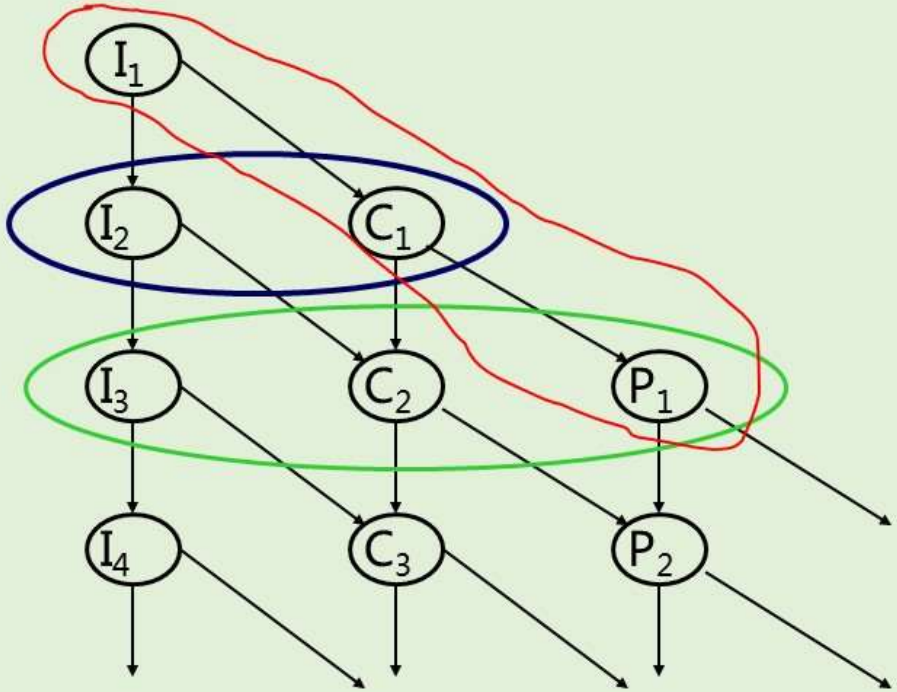
9

例：用下图说明在多道批处理系统中，大量操作执行的先后次序。

- **讨论：**

（1）哪些程序段的执行必须是顺序的？为什么？

（2）哪些程序段的执行是可并行的？为什么？



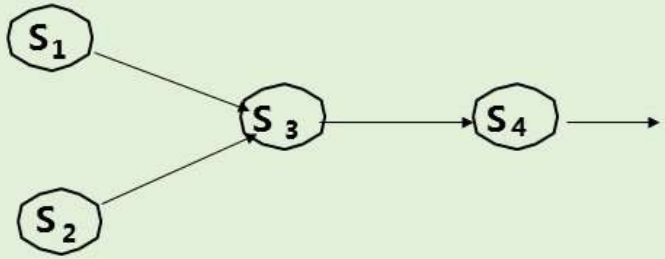
对于具有下述四条语句的程序段：

S1: a：=x+2□

S2: b：=y+4□

S3: c：=a+b□

S4: d：=c+b



程序并发执行的特征：

- **间断性**：相互制约关系，运行→暂停→运行
- **失去封闭性**：如果程序执行的结果是一个与时间无关的函数，即具有**封闭性**。若一个程序的执行可改变另一个程序的变量，程序执行的结果不仅依赖于程序的初始条件，还依赖于程序执行时的相对速度，在这种情况下结果是不确定的，即失去了程序的封闭性。
- **失去可再现性**：失去封闭性 - >失去可再现性；外界环境在程序的两次执行期间发生变化，失去原有的可重复特征。

·程序并发执行的条件

读集

·R(Pi)={a1,a2,…,am}表示程序Pi在执行期间需要参考的变量的集合

写集

·W(Pi)={b1,b2,…,bn}表示程序Pi在执行期间要改变的变量的集合

Bernstein条件

若两个程序P1、 P2能满足下述条件，它们便能并发执行，且具有可再现性。

$R(P1)\cap W(P2) \cup R(P2)\cap W(P1) \cup W(P1)\cap W(P2) = \{\}$

- 例：四条语句
  S1: a:=x+y
  S2: b:=z+1
  S3: c:=a-b
  S4: w:=c+1
- R(S1)={x,y}, W(S1)={a}
- R(S2)={z}, W(S2)={b}
- R(S3)={a,b}, W(S3)={c}
- R(S4)={c}, W(S4)={w}

由 Bernstein条件：

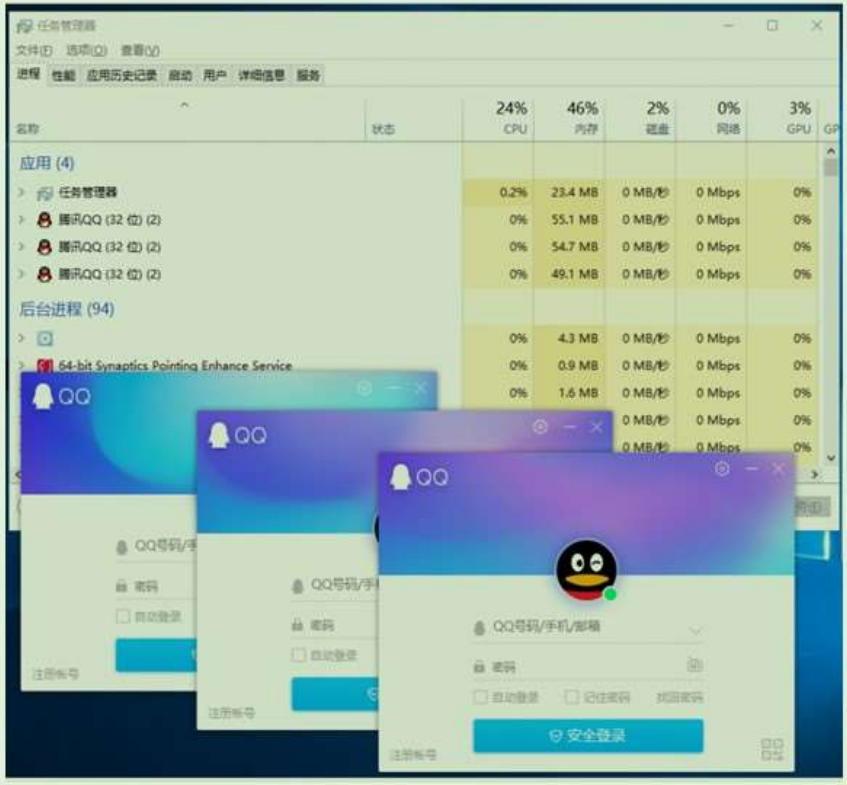
$R(S1) \cap W(S2) \cup R(S2) \cap W(S1) \cup W(S1) \cap W(S2) = \{\}$

$R(S1) \cap W(S3) \cup R(S3) \cap W(S1) \cup W(S1) \cap W(S3) = \{a\}$

.....

14

2.2 进程的描述

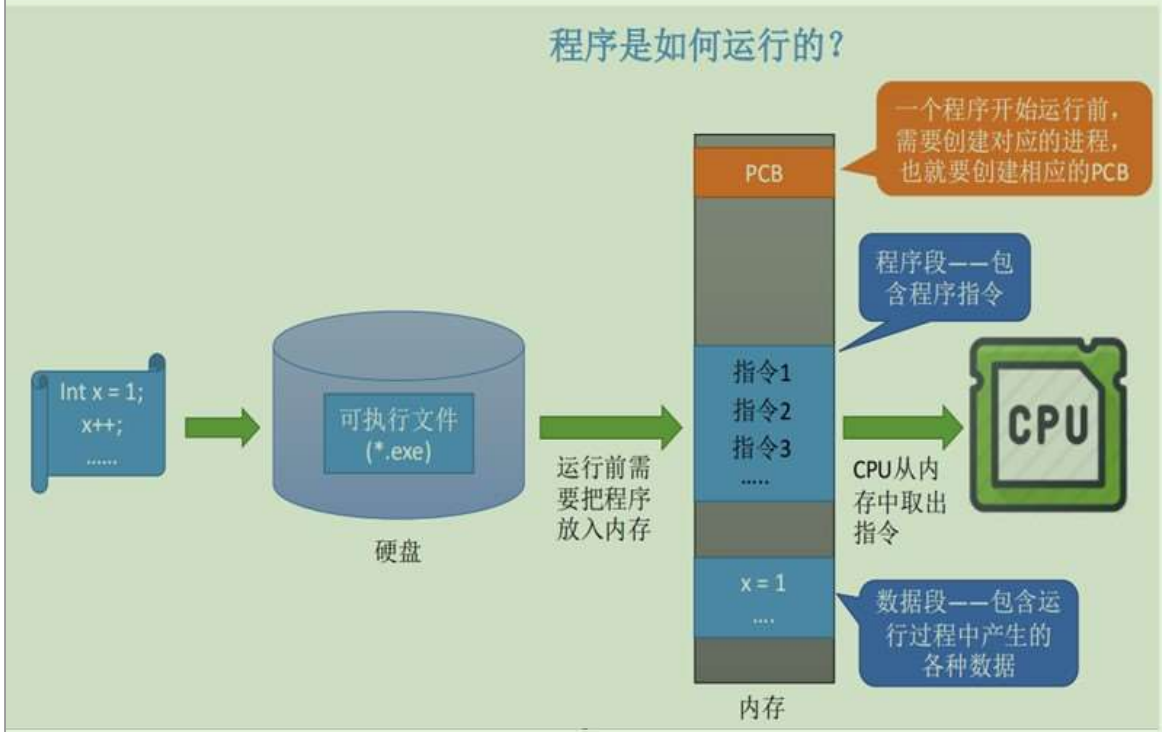


程序：是静态的，就是个存放在磁盘里的可执行文件，就是一系列的指令集合。

进程（Process）：是动态的，是程序的一次执行过程。

同一个程序多次执行会对应多个进程

- 当进程被创建时，操作系统会为该进程分配一个唯一的、不重复的“身份证号”——PID（Process ID，进程ID）
  - 操作系统要记录PID、进程所属用户ID（UID）
  - 还要记录给进程分配了哪些资源（如：分配了多少内存、正在使用哪些I/O设备、正在使用哪些文件）
  - 还要记录进程的运行情况（如：CPU使用时间、磁盘使用情况、网络流量使用情况等）



一个进程实体（进程映像）由PCB、程序段、数据段组成。
进程是动态的，进程实体（进程映像）是静态的。
进程实体反应了进程在某一时刻的状态（如：x++后，x=2）

程序段、数据段、PCB三部分组成了进程实体（进程映像）
引入进程实体的概念后，可把进程定义为：
进程是进程实体的运行过程，是系统进行资源分配和调度的一个独立单位。
注意：PCB是进程存在的唯一标志！

一个进程被“调度”，就是指操作系统决定让这个进程上CPU运行