第四章　习题及解答

4-3 什么是进程？进程与程序的主要区别是什么？

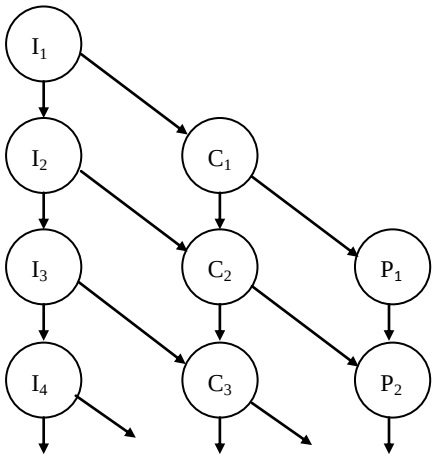
答：进程是一个具有一定独立功能的程序关于某个数据集合的一次活动。进程与程序的主要区别是。

(1) 程序是指令的有序集合，是一个静态概念。进程是程序在处理机的一次执行过程，是一个动态概念。进程是有生命期的，因创建而产生，因调度而执行，因得到资源而暂停，因撤消而消亡；

(2) 进程是一个独立的运行单元，是系统进行资源分配和调度的独立单元，而程序则不是。

(3) 进程与程序之间无一一对应关系。一个程序可以对应多个进程，一个进程至少包含一个程序。

4-4 下图标明程序段执行的先后次序。其中：I 表示输入操作，C 表示计算操作，P 表示打印操作，下角标说明是对哪个程序进行上述操作。请指明： 哪些操作必须有先后次序? 其原因是什么？ 哪些操作可以并发执行? 其原因又是什么？



答：(1) ① I_n、C_n 和 P_n 之间有先后顺序要求，这是由于程序本身的逻辑要求。② 使用同一设备的不同的程序段，如 $C_1 \cdots C_n$，$I_1 \cdots I_n$，$P_1 \cdots P_n$，之间有先后顺序要求，这是由于设备某一时刻只能为一个程序服务。

(2) 不同程序使用不同设备时，占用不同设备，无逻辑关系，可以并发执行，如 I_2 和 C_1；I_3、C_2 和 P_1。

4-9 某系统进程调度状态变迁图如图 4.28 所示，请说明：

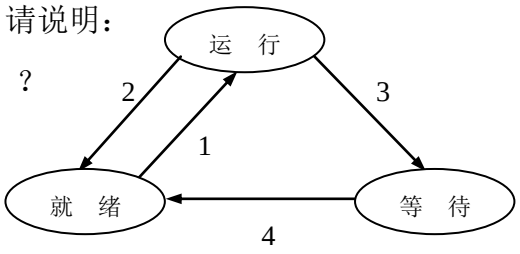


图 4.29

(1) 什么原因会导致发生变迁 2、变迁 3、变迁 4 ？

(2) 当观察系统中进程时，可能看到某一进程产生的一次状态变迁将引起另一进程作一次状态变迁，这两个变迁称为因果变迁。在什么情况下，一个进程的变迁 3 能立即引起另一个进程发生变迁 1 ？

(3) 下述因果变迁是否可能发生 ？如果可能的话，在什么情况下发生？

a. 2→1； b. 3→2； c. 4→1

(1) 什么原因会导致发生变迁 2、变迁 3、变迁 4 ？

答：发生变迁 2 的原因：时间片到

发生变迁 3 的原因：请求 I/O 或其他系统调用

发生变迁 4 的原因：I/O 完成或其他系统调用完成

(2) 在什么情况下，一个进程的变迁 3 能立即引起另一个进程发生变迁 1 ？

答：一个进程的变迁 3 能立即引起另一个进程发生变迁的条件是，就绪队列非空。

(3) 下列因果变迁是否可能发生？若可能，需要什么条件？

a. 2→1； b. 3→2； c. 4→1

答：a. 2→1 不需要条件，一定会发生。

b. 3→2 不是因果变迁。

c. 4→1 可能发生，条件：在可剥夺调度方式下，转变为就绪状态的进程优先级最高。

4-12 n 个并发进程共用一个公共变量 Q，写出用信号灯实现 n 个进程互斥时的程序描述，给出信号灯值的取值范围, 并说明每个取值的物理意义。

解：(1) 程序描述

```
main ( ){
  int mutex=1;   //公共变量 Q 的互斥信号灯//
  cobegin
```

P_1; P_2; … P_n;

coend

}

P_i(){

…

P(mutex);

使用 Q;

V(mutex);

…

}

(2) 信号灯值的取值范围：[1，– (n–1)]

(3) mutex 每个取值的物理意义：

mutex = 1　说明没有进程进入临界段执行；

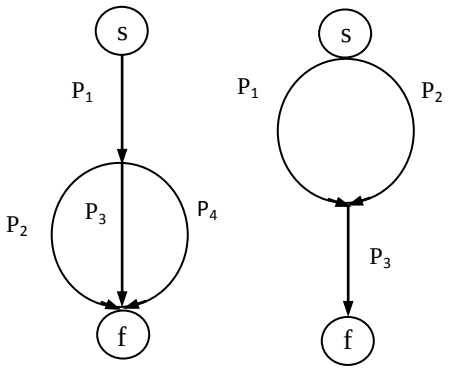
mutex = 0　说明有一个进程进入临界段执行；

mutex = –1　说明有一个进程进入临界段执行，另有一个进程正在等待进入；

⋮

mutex = – (n–1)　说明有一个进程进入临界段执行，另有(n–1)个进程正在等待进入。

4-14 图 4.30(a)、(b) 分别给出了两个进程流图。试用信号灯的 p、v 操作分别实现图 4.30(a)、(b)所示的两组进程之间的同步，并写出程序描述。



(a)　(b)

图 4.30

图(a) 解：

```
main(){
    int s12=0,s13=0,s14=0;   // s12 表示进程 P2 可否执行；s13 表示进程 P3 可否执行；s14 表示进程 P4 可否执行//
    cobegin
        P1; P2; P3; P4;
    coend
}
P1(){
    p1   execute;
    V(s12);
    V(s13);
    V(s14);
}

P2(){
    P(s12);
    p2   execute;
}

P3(){
    P(s13);
    p3   execute;
}

P4(){
    P(s14);
    p4   execute;
}
```

图(b)

```
main(){
    int s13=0,s23=0; // s13 表示进程 P1 执行完成否；s23 表示进程 P2 执行完成否 //
    cobegin
        P1; P2; P3;
    coend
}

P1(){
    P1  execute;
    V(s13);
}

P2(){
    P2  execute;
    V(s23);
}

P3(){
    P(s13);
    P(s23);
    P3  execute;
}
```

4-15 如图 4.32 所示，get、copy 和 put 三个进程共用两个缓冲区 s 和 t (其大小每次存放一个记录) ，get 进程负责不断地把输入记录输入缓冲区 s 中，copy 进程负责从缓冲区 s 中取出记录复制到缓冲区 t 中，而 put 进程负责从缓冲区 t 中取出记录打印。试用 PV 操作实现这三个进程之间的同步，并写出程序描述。

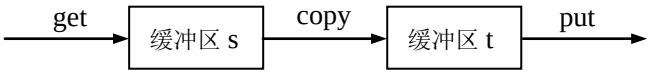


图 4.32

解：

```
main(){
    int s1=1,s2=0; // s1 表示缓冲区 S 是否为空，s2 表示是否已满//
    int s3=1, s4=0; // s3 表示缓冲区 T 是否为空，s4 表示是否已满//
    cobegin
        get;
        copy;
        put;
    coend
}

get(){
    while(1){
        P(s1);
        input data to buffer S;
        V(s2);
    }
}

copy (){
    while(1){
        P(s2);
        copy data from buffer S;
        V(s1);
        P(s3);
        input copy-data to buffer T;
        V(s4);
```

```
    }
}

put(){
    while(1){
        P(s4);
        output data to buffer S;
        V(s3);
    }
}
```

4-16 什么是进程的互斥与同步？同步和互斥这两个概念有什么联系和区别？

答：在操作系统中，当某一进程正在访问某一存储区域时，就不允许其他进程读出或者修改该存储区的内容，否则，就会发生后果无法估计的错误。进程之间的这种相互制约关系称为互斥。

所谓同步，就是并发进程在一些关键点上可能需要互相等待与互通消息，这种相互制约的等待与互通信息称为进程同步。

同步和互斥这两个概念都属于同步范畴，描述并发进程相互之间的制约关系。同步是指并发进程按照他们之间的约束关系，在执行的先后次序上必须满足这种约束关系。而互斥是同步的一种特例，是指并发进程按照他们之间的约束关系，在某一点上一个时刻只允许一个进程执行，一个进程做完了，另一个进程才能执行，而不管谁先做这个操作。

4-22 什么是线程？线程和进程有什么区别?

答：线程也称为轻量级进程，它是比进程更小的活动单位，它是进程中的一个执行路径，一个进程可能有多个执行路径，即线程。

线程和进程的主要区别如下。

(1) 线程是进程的一个组成部分，一个进程可以有多个线程，而且至少有一个可执行的线程。

(2) 进程是资源分配的基本单位，它拥有自己的地址空间和各种资源；线程是处理机调度的基本单位，它只能和其他线程共享进程的资源，而本身并没有任何资源。

(3) 进程的多个线程都在进程的地址空间内活动。这样，在以线程为单位进行处理机调度和切换时，切换时间较短；而以进程为单位进行处理机调度和切换时，由于涉及到资源转移及现场保护等问题，将导致切换时间变长和资源利用率下降。

(4) 线程和进程一样，都有自己的状态和相应的同步机制，但是，由于线程没有自己单独的程序和数据空间，因而不能像进程那样将程序和数据交换到外存去。

(5) 进程的调度和控制大多由操作系统的内核完成，而线程的控制既可以由操作系统内核完成，也可以由用户控制完成。

4-24 某系统的进程状态变迁图如图 4.36 所示 (设该系统的进程调度方式为非剥夺方式)。

(1) 说明一个进程发生变迁 3 的原因是什么？发生变迁 2、变迁 4 的原因又是什么 ？

(2) 下述因果变迁是否会发生，如果有可能的话，在什么情况下发生 ？

① 2→5； ② 2→1； ③ 4→5； ④ 4→2； ⑤ 3→5

(3) 根据此进程状态变迁图叙述该系统的调度策略、调度效果。

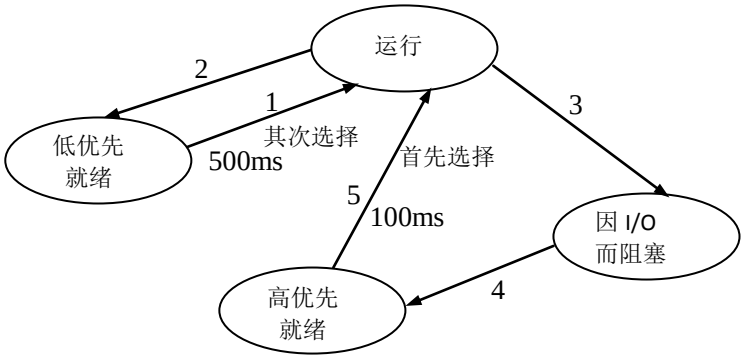


图 4.36

(1) 答：

发生变迁 3 的原因：当运行进程在执行过程中，需要等待某事件的发生才能继续向下执行时会发生变迁 3。

发生变迁 2 的原因：运行进程在分得的时间片 100ms 或 500ms 内未完成，当其时间片到时将发生变迁 2。

发生变迁 4 的原因：当等待进程等待的事件发生了，将会发生变迁 4。

(2) 答：

① 2→5 的因果变迁可能发生。条件是：高优先就绪队列非空。

② 2→1 的因果变迁可能发生，当运行进程的时间片到时发生的变迁 2，若此时高优先就绪队列为空，必然引起低优先就绪队列中的一个就绪进程被调度执行而发生变迁 1。

③ 4→5 的因果变迁不可能发生，因为采用的是非剥调度夺式。

④ 4→2 的因果变迁不可能发生。

⑤ 3→5 的因果变迁可能发生，条件是：高优先就绪队列非空。

(2) 答：

调度策略：首先调度高就绪队列中的进程 (一般是 I/O 型进程) 投入运行，给高优先就绪队列中的进程分配的时间片大小为 100ms。只有当高就绪队列中的所有进程全部运行完或因等待某事件发生处于阻塞状态，高就绪队列中没有进程可运行时，才调度低优先就绪队列中的进程 (一般是计算型进程) ，给低优先就绪队列中的进程分配的时间片大小为 500ms。若一个运行进程时间片 100ms 或 500ms 到时未完成就进入低优先就绪队列。若某进程在运行期间因等待某事件发生而进入阻塞队列，则当所等待事件完成后，它将进入高优先就绪队列。

调度效果：这种算法优先照顾了 I/O 量大的进程 (高优先级) ，但通过给计算型进程分配更长的时间片也适当照顾了计算型进程。