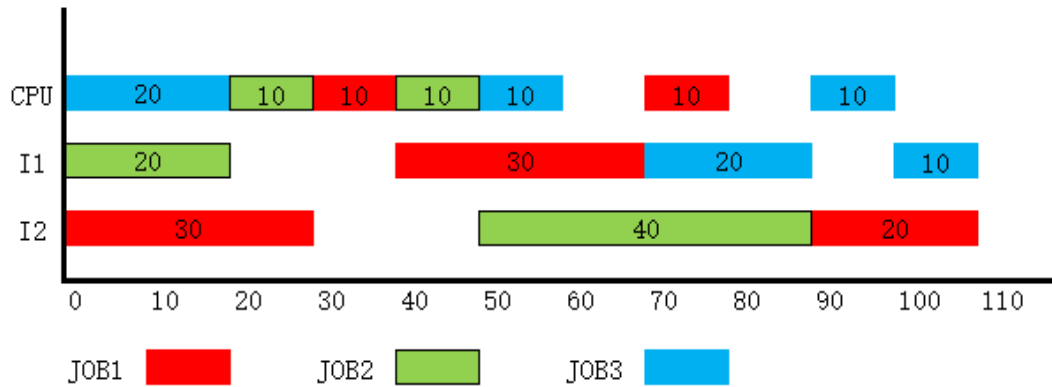


## 一、多道批处理

(共 6 分，不能清晰表达酌情扣分)



(共 2 分，错一个扣 1 分)

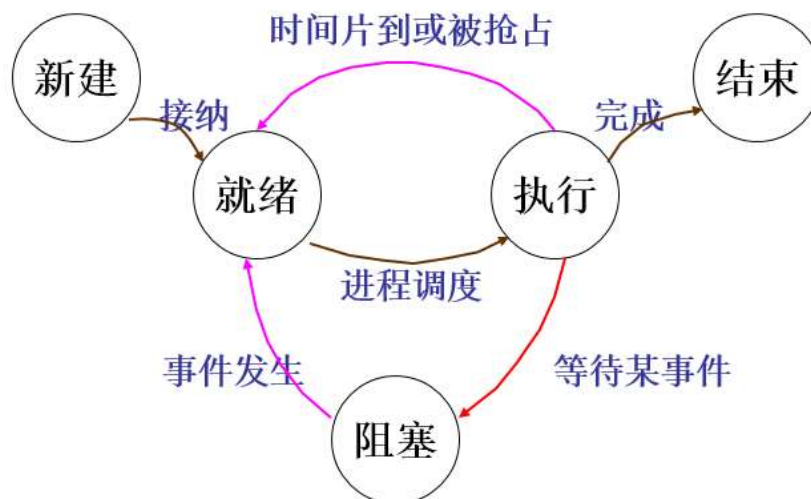
- (1) JOB1: 110ms  
JOB2: 90ms  
JOB3: 110ms

(共 2 分，错一个扣 1 分)

- (2) CPU 利用率:  $(20+10+10+10+10+10+10) / 110$   
I1 利用率:  $(20+30+20+10) / 110$   
I2 利用率:  $(30+40+20) / 110$

## 二、进程生命周期

(共 5 分，少一个状态扣 1 分，少一个箭头及原因扣 1 分)



(每小题 1 分，共 5 分。每个要点 1 分)

(1) 新建：进程正在创建中(作业调度，即程序或作业从外存调入内存)的状态。此时系统需要分配和建立 PCB 表项；建立资源表格并分配资源；加载程序并建立地址空间表。

(2) 就绪：进程已获得了除 CPU 以外的所有资源，等待分配 CPU 执行时进入的状态，通常由一个就绪队列进行管理。在下列情况下进程进入就绪态：一是新进程被接纳时；二是因 CPU 时间片到或被抢占时；三是陷入阻塞状态的进程因事件发生而等待 CPU 调度时。

(3) 执行：当一个进程获得必要的资源并正在处理机上执行的状态。通常 CPU 空闲时会从就绪队列队首调度一个进程，将其转入执行状态。

(4) 阻塞：由于发生某事件而暂时无法执行下去时进程所处的状态，通常由一个阻塞队列进行管理。当等待事件发生时，进程将进入就绪状态。

(5) 结束：进程执行完毕，释放所占资源的状态。导致进程结束的原因包括但不限于：正常完成、父进程终止、无可利用内存等。

### 三、银行家算法

(5 分，要求通过银行家算法矩阵，给出至少一个安全序列)

(1) 如下表所示 (推导过程略)，存在安全序列<P0, P3, P1, P2, P4>、<P0, P3, P4, P1, P2>， <P0, P3, P1, P4, P2>因此，该状态是安全的。

PROC 进程	MAX 总需求				NEED 待分配				ALLOCATION=MAX-NEED 已分配				AVAILABLE=WORK 可提供				WORK+ALLOCATION 回收后				次序
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	
P0	0	0	4	4	0	0	1	2	0	0	3	2	1	6	2	2	1	6	5	4	1
P1	2	7	5	0	1	7	5	0	1	0	0	0	1	9	9	10	2	9	9	10	4
P2	3	6	10	10	2	3	5	6	1	3	5	4	2	9	9	10	3	12	14	14	5
P3	0	9	8	4	0	6	5	2	0	3	3	2	1	6	5	4	1	9	8	6	2
P4	0	6	6	10	0	6	5	6	0	0	1	4	1	9	8	6	1	9	9	10	3

(3 分，结论正确 1 分，说明原因 2 分)

(2) 根据银行家算法检查，若给进程 P3 分配资源(1,2,2,2)，系统剩余资源数为(0,4,0,0)，则无法满足其他进程的需求，系统将进入不安全状态，因此，不能将资源分配给 P3。

(2 分，结论正确 1 分，说明原因 1 分)

(3) 不会立即死锁。不安全不等于立即死锁 (危险区≠死锁区)。

#### 四、处理机调度算法

(2分)

(1) 进程执行次序为:

先来先服务法	非抢占式的优先数法
P <sub>1</sub> 、P <sub>2</sub> 、P <sub>3</sub> 、P <sub>4</sub>	P <sub>1</sub> 、P <sub>2</sub> 、P <sub>4</sub> 、P <sub>3</sub>

(2)

(共4分。过程正确2分，结论正确2分)

等待时间即开始时刻。

先来先服务法:

执行次序	提交时刻	运行时间	开始时刻	完成时刻	周转时间	带权周转时间
1	0	20	0	20	20	1.00
2	0	15	20	35	35	2.33
3	0	10	35	45	45	4.50
4	0	12	45	57	57	4.75
进程平均等待时间			$(0+20+35+45) / 4 = 25$			
进程平均周转时间			$(20+35+45+57) / 4 = 39.25$			
进程平均带权周转时间			$(1+2.33+4.5+4.75) / 4 = 3.15$			

(共4分。过程正确2分，结论正确2分)

非抢占式的优先数法:

执行次序	提交时刻	运行时间	开始时刻	完成时刻	周转时间	带权周转时间
1	0	20	0	20	20	1.00
2	0	15	20	35	35	2.33
4	0	12	35	47	47	3.92
3	0	10	47	57	57	5.70
进程平均等待时间			$(0+20+35+47) / 4 = 25.5$			
进程平均周转时间			$(20+35+45+57) / 4 = 39.75$			
进程平均带权周转时间			$(1+2.33+4.5+4.75) / 4 = 3.24$			

## 五、信号量与进程同步

(共 4 分, 同步、互斥各 2 分)

(1) 分析: 本题可以归类为生产-消费问题。

座位类似于产品缓冲区; logout 进程被调度时, 释放出一个空座位; 当存在空座位时, login 进程可以被调度, 并减少一个空座位; login 和 logout 进程存在同步关系;

但多个进程不能同时修改空座位的数量 (临界资源); 故此刻存在互斥关系。

(2) 代码如下:

(2 分)

```
const int N=100;

int seats=N, readers=0, mutex=1;

int buff[]=new int[N]; //array

int in=0, out=0;        //pointer of buff
```

(2 分, 要求 P/V 成对出现且顺序正确)

```
void logout(){
    while(1){
        P(readers);

        P(mutex);

        next=buff[out]; //delete info;

        out=(out+1)%n;

        V(mutex);
```

```
        V(seats);  
    }
```

(2 分, 要求 P/V 成对出现且顺序正确)

```
void login(){  
    while(seats>0){        //leave if no seat  
        r=getreaderInfo(); //come in  
        P(seats);  
        P(mutex);  
        buff[in]=r;        //write down info  
        in=(in+1)%n;  
        V(mutex);  
        V(readers);  
    }  
}
```