

桌上有个能盛得下五个水果的空盘子。爸爸不停地向盘中放苹果或桔子，儿子不停地从盘中取出桔子享用，女儿不停地从盘中取出苹果享用。规定三人不能同时从盘子中取放水果。试用信号量实现爸爸、儿子和女儿这三个循环进程之间的同步。

```
semaphore empty=5, orange=0, apple=0, mutex=1;
```

```
Dad(){
```

```
    while(1){
```

```
        wait(empty);
```

```
        wait(mutex);
```

```
        将水果放入盘中;
```

```
        signal(mutex);
```

```
        if(放入的是桔子) signal(orange);
```

```
        else signal(apple);
```

```
    }
```

```
}
```

```
Son(){
```

```
    while(1){
```

```
        wait(orange);
```

```
        wait(mutex);
```

```
        从盘中取一个桔子;
```

```
        signal(mutex);
```

```
        signal(empty);
```

```
        享用桔子;
```

```
    }
```

```
}
```

```
Daughter(){
```

```
    while(1){
```

```
        wait(apple);
```

```
        wait(mutex);
```

```
        从盘中取一个苹果;
```

```
        signal(mutex);
```

```
        signal(empty);
```

```
        享用苹果;
```

```
    }
```

```
}
```



假设一个系统中有5个进程，它们的到达时间和服务时间如表3-1所示，忽略I/O以及其他开销时间，若分别按先来先服务(FCFS)、抢占的短作业优先(SJF)、时间片轮转(RR,时间片=1)进行CPU 调度，请给出各进程的完成时间、周转时间、带权周转时间、平均周转时间。

表 3-1 进程到达和需服务时间

进程	到达时间	服务时间
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

FCFS

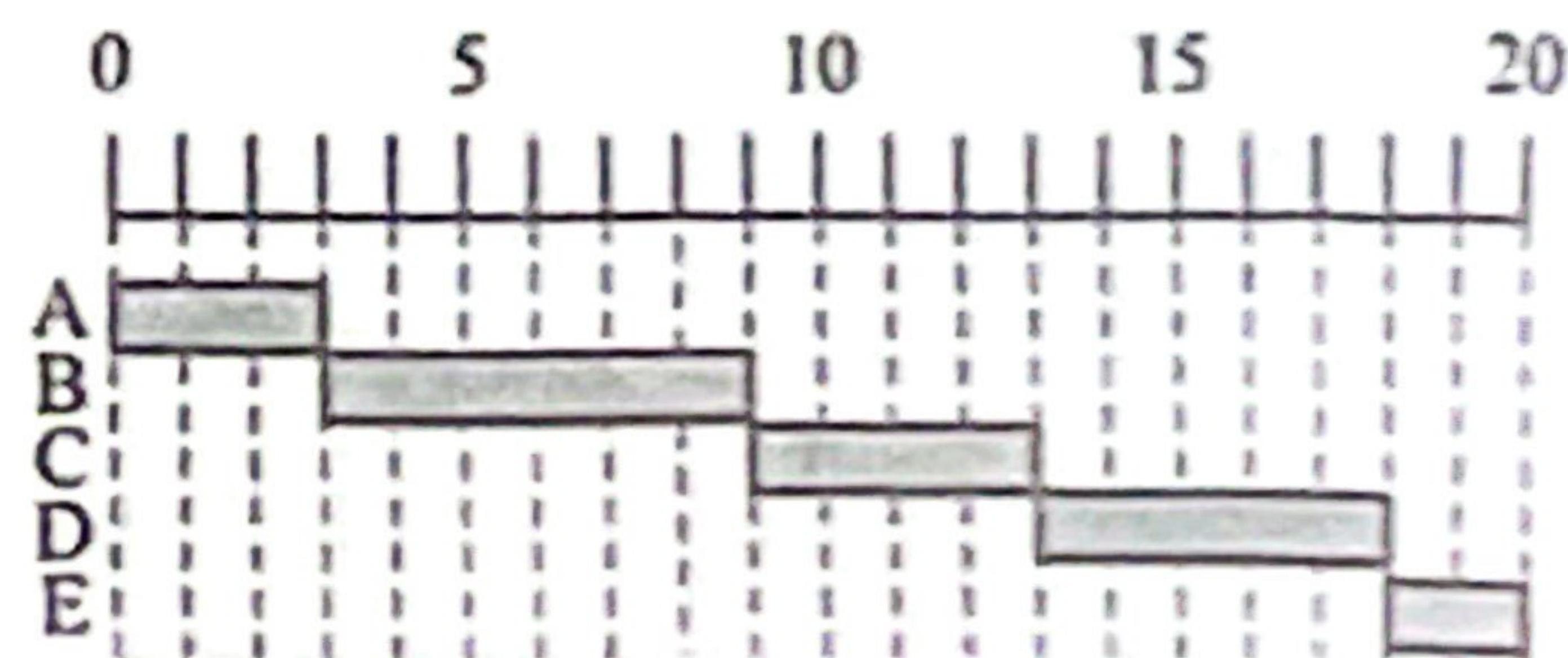




表 3-1 进程到达和需服务时间

进程	到达时间	服务时间
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

FCFS

SJF(非抢占)

SJF(抢占)

RR(q=1)

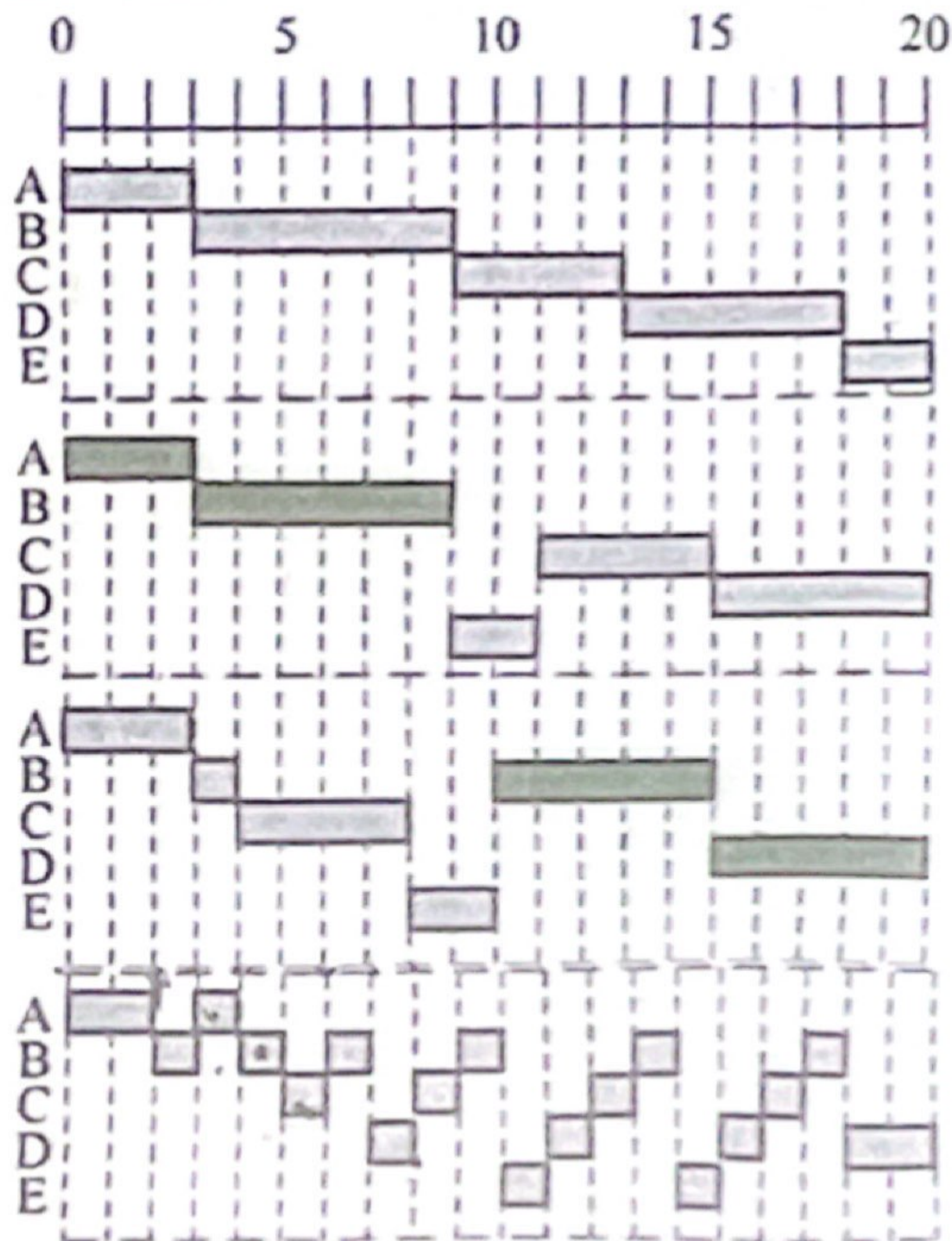




表 3-2 进程的完成时间和周转时间

	进 程	A	B	C	D	E	平均
FCFS	完成时间	3	9	13	18	20	
	周转时间	3	7	9	12	12	8.6
	带权周转时间	1.00	1.17	2.25	2.40	6.00	2.56
SJF(非抢占)	完成时间	3	9	15	20	11	
	周转时间	3	7	11	14	3	7.6
	带权周转时间	1.00	1.17	2.75	2.80	1.50	1.84
SJF(抢占)	完成时间	3	15	8	20	10	
	周转时间	3	13	4	14	2	7.2
	带权周转时间	1.00	2.16	1.000	2.80	1.00	1.59
HRRN	完成时间	3	9	13	20	15	
	周转时间	3	7	9	14	7	8
	带权周转时间	1.00	1.17	2.25	2.80	3.50	2.14