1 有5个进程Pa、Pb、Pc、Pd和Pe，它们依次进入就绪队列，它们的优先级和需要的处理机时间：

进程 处理机时间 优先级  
Pa 10 3  
Pb 1 1  
Pc 2 3  
Pd 1 4  
Pe 5 2  
忽略进程调度等所花费的时间，请回答下列问题：  
(1) 分别写出采用“先来先服务”和“非抢占式的优先数(优先数大优先级低)”调度算法选中进程执行的次序。  
(2) 分别计算出上述两种算法使各进程在就绪队列中的等待时间及两种算法下的平均等待时间。

(1) 采用“先来先服务”调度算法时，进程调度次序是：Pa→Pb→Pc→Pd→Pe。采用“非抢占式的优先数”调度算法时，进程调度次序是：Pa→Pb→Pe→Pc→Pd。

(2) 采用“先来先服务”调度算法时，各进程的等待时间如表2-15所示。采用“非抢占式的优先数”调度算法时，进程调度次序如表2-16所示。

表2-15 “先来先服务”各进程的等待时

进程 等待时间 运行时间

Pa 0 10

Pb 10 1

Pc 11 2

Pd 13 1

Pe 14 5

平均等待时间 (0+10+11+13+14)/5=9.6

表2-16 “非抢占式的优先数(优先数大优先级低)”各进程的等待时间

进程 等待时间 运行时间

Pa 0 10

Pb 10 1

Pe 11 5

Pc 16 2

Pd 18 1

平均等待时间 (0+10+11+16+18)/5=11

2 某寺庙,有小和尚和老和尚若干,有一个水缸,由小和尚提水入缸供老和尚饮用.水缸可以容纳10桶水,水取自同一口井中,由于水井口窄,每次只能容纳一个水桶取水.水桶总数为3个.每次入水、取水仅为一桶,且不可同时进行.试给出有关取水、入水的算法描述.

小和尚从井中取水并向缸中倒水为一个进程，而老和尚从缸中取水为另一个进程。

有关互斥的资源有：水井(一次仅允许一个水桶进出)；水缸(一次倒水、取水仅一个水桶)。分别为它们设置信号量mutexA、mutexB来实现互斥，初值均为1。

有关同步的问题是：3个水桶，无论是从井中取水还是倒水入缸或取水出缸都是一次一个，即为其设置信号量count，初值为3，抢不到水桶的进程只好等待。此外，设置信号量empty来控制入缸的水量，初值为10，当水缸满时不可入水；设置信号full控制出缸的水量，初值为0，当水缸空时不可出水。

mutexA=1, 实现对井的互斥

mutexB=l, 实现对缸的互斥

empty=10, 缸中能装多少水，初始是10桶

full=0, 缸中有多少桶水，初始是0桶

count=3; 表示有3个空桶可用

main()

{

Cobegin

小和尚i(i=A，B，…)打水;

老和尚j(j=A，B，…)取水;

Coend;

}

小和尚i(i=A，B，…)打水：

{

while(true)

{ P(empty); //看水缸满否，满则阻塞打水进程

P(count); //申请打水的桶

去井边

P(mutexA); //互斥使用水井，即不允许两和尚同时打水

从井中取水;

V(mutexA);

回寺庙

P(mutexB); //互斥使用水缸

送水入缸;

V(mutexB);

V(count); //归还水桶

V(full) //水缸又多一桶水

}

}

老和尚j(j=A, B，…)取水：

{while(true)

{ P(full); //看水缸是否有水，无水则阻塞取水进程

P(count); //申请取水的桶

P(mutexB); //互斥使用水缸

从缸中取水;

V(mutexB);

喝水

V(count) //归还水桶

V(empty); //缸中少了一桶水

}

}