2021-2022-2 B 卷参考答案

一、选择题（每题 1 分，共 25 分）

1.D	2.A	3.A	4.B	5.B	6.D	7.B	8.A	9.D	10.C
11.A	12.A	13.C	14.B	15.A	16.D	17.B	18.C	19.C	20.D
21.C	22.D	23.B	24.A	25.D					

二、综合题（共 75 分）

1、答:（共 8 分）

（1）推动批处理系统形成和发展的主要动力是”不断提高系统资源利用率“和”提高系统吞吐量”。它们主要表现在:脱机输入/输出技术的应用和作业的自动多度大大地提高了 I/O 的速度、I/O 设备与 CPU 并行工作的程度，减少了主机 CPU 的空闲时间；多道程序设计技术的应用更进一步提高了 CPU、内存和 I/O 设备的利用率和吞吐量。（4 分）

（2）推动分时系统形成和发展的主要动力则是“为了更好地满足用户的需要“。主要表现在 CPU 的分时使用缩短了作业的平均周转时间：人机交互能力的提供使用户能方便地直接控制自己的作业；主机的共享，使多个用户(包括远程用户)能同时使用同一台计算机，独立地、互不干扰地处理自己的作业。（4 分）

2、答:（共 12 分）

（1）根据题意，当酒吧老板提供两种物品时，必须被同一个音乐爱好者取走，我们应该把每两种物品看成是一个组合起来的临界资源。第 1 队的音乐爱好者等待的组合资源是音乐磁带和电池，设置一个信号量 S1；第 2 队的音乐爱好者等待的组合资源是随身听和电池，设置一个信号量 S2；第 3 队的音乐爱好者等待的组合资源是随身听和音乐磁带，设置一个信号量 S3；S1，S2, S3 的初值都为 0。只有一个音乐爱好者听完一首乐曲后，酒吧老板才能再次出售商品，为这种同步关系也必须设置一个初值为 0 的信号量 music_over。（2 分）

（2）semaphore　S1=S2=S3=0;

```
semaphore    music_over=0;          （1 分）
fan1( ){     （2 分）
        wait(S1);
        买音乐磁带和电池;
        听乐曲;
        signal（music_over）
}
fan2( ){       （2 分）
        wait(S2);
        买随身听和电池;
        听乐曲;
        signal（music_over）
}
fan3( ){          （2 分）
        wait(S3);
        买随身听和音乐磁带;
        听乐曲;
        signal（music_over）
```

```
}
boss(){            (3 分)
    while(1){
        提供任意两种物品出售;
        if (提供的是音乐磁带和电池)        signal (S1);
        else if (提供的是随身听和电池)        signal (S2);
        else signal (S3);
        wait (music_over);
    }
}
main()
{
  Cobegin
    fan1();fan1();fan3();boss();
coend
}
```

3、答: (共 12 分)

(1) 对资源分配图进行简化，能完全简化，没有死锁发生。

简化顺序: P3，P1，P2 或 P1，P3，P2 // (本小题共 3 分，是否发生死锁 1 分，判断过程 2 分)

(2) (本小题共 5 分，数据结构定义 2 分，伪代码描述 3 分)

死锁检测算法: Coffman 算法

①可利用资源向量 Available。长度为 m，描述系统中每类资源的当前可分配数量。

②分配矩阵 Allocation。一个 n×m 的矩阵，描述当前各进程已经分配到的各类资源的数量。

③资源请求矩阵 Request。一个 n×m 的矩阵，描述当前各进程对各类资源的请求数量。

死锁检测算法的流程描述如下:

①首先设置两个向量: 工作向量 Work，长度为 m，描述系统能够提供的各类资源的可用数量，初始化为 Work=Available；布尔向量 Finish，长度为 n，对 i=0,1,2,…,n-1，初始化时，若进程 i 的 Allocation[i]=0，则其 Finish[i]=True；否则 Finish[i]=False，。

② 寻找能同时满足以下两个条件的进程:

Finish[i]=False; $Request_i \leqslant Work$

如果找到，则转第③步，否则转第④步。

③执行如下操作: Work=Work + Allocation[i]，Finish[i]=True，再转回第②步。

④如果对所有 i (i=(0,1,2, …,n-1))，Finish[i]=True，则可判定系统没有发生死锁；否则系统有死锁发生，且如果 Finish[i]=False，则进程 P_i 死锁。

参考伪代码如下:

```
Work=Available;
Boolean Finish[n];
Deadlock=false;
for (i=0;i<=n-1;i++)
{
    If Allocation[i]==0   Finish[i]=True;
    Else Finish[i]=False;
```

```
    }
    for (i=0;i<=n-1;i++)
    {
       If (Finish[i]==false) and (Request_i ≤Work ) do
            {
                Work=Work+Allocation;
                Finish[i]=True;
            }
    }
    for (i=0;i<=n-1;i++)
    {
        If (finish[i]==false)   deadlock=true;
    }
  printf("是否发生死锁：%c", deadlock);
```

（3）（本小题共 4 分，死锁检测时机考虑因素 2 分，具体方案可以参考以下三种，合理就给分，方案和性能影响 2 分）

死锁检测时机需要综合考虑几个因素：①死锁出现的频繁程度；②发生死锁时受到死锁影响的进程数量；③进行死锁检测的系统开销。通常有以下几种思路：

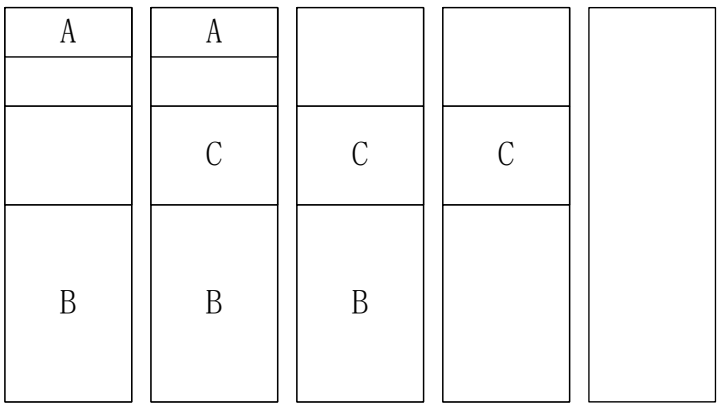
①每当进程请求资源且得不得满足时就做检测，因为死锁只可能在这种情况下发生，这时候检测能及早发现死锁，并及时解除。但系统中进程请求资源的频率很高，如果死锁发生的频繁程度并不高，则会浪费较多的 CPU 时间。

②周期性定时检测。系统可综合考虑前面提到的 3 个因素，设置一个合理的检测周期，每隔一段时间检测一次，这样既能有效控制检测的系统开销，又能相对及时地发现死锁。

③依据 CPU 利用率确定是否进行检测。系统为 CPU 利用率设置一个阈值，每当 CPU 利用率低于这个阈值时，就进行死锁检测，因为 CPU 利用率低有可能是某些进程发生死锁不能参与运行而造成的。

4、答：答：（共 10 分）

（1）（5 分）



、

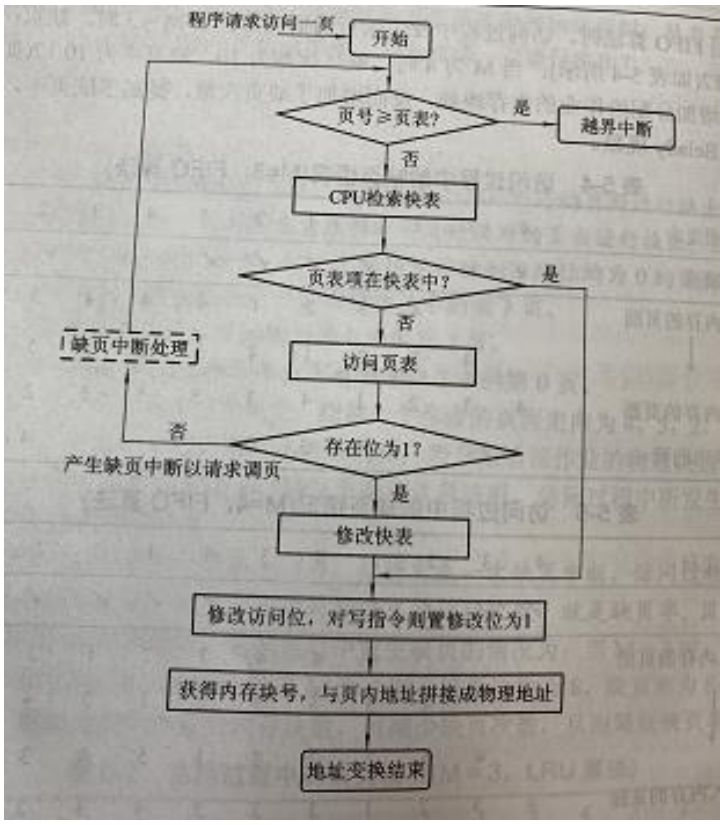
（2）合理即可（5 分）

5、答：（共 12 分）

（1）（6 分）根据题意，分配给作业的内存块数为 3，而页面的引用次序为：3、3、1、3、2、3、0、2、1、2、3、0、1、1。因此，可以计算出，采用 LRU 算法时，缺页次数为 8；采用 FIFO 算法时，缺页次数为 6。

LRU 算法用最近的过去作为预测最近的将来的依据，因为程序执行的局部性规律，一般有较好的性能，但实现时，要记录最近在内存的每个页面的使用情况，比 FIFO 算法苦难，其开销也大。有时，因页面的过去和未来的走向之间并无必然的联系，如上面，LRU 算法的性能就没有想象中那么好。

（2）（6 分）地址转换流程如下图所示。



6、答：（共 11 分）

（1）（4 分）如果盘块地址只需用 2 个字节来描述，则该磁盘系统中盘块的数目将小于等于 2^{16}，65536 块，故文件大小也不会超过 65536 块；而每个盘块中可存放 256 个盘块号，因此系统最多只要用到 2 次间接地址。实际上，使用一个一次间接地址和一个二次间接地址后，允许文件的最大长度已达 11+256+256*256 块，已经超出了该磁盘系统中实际的盘块数目。

（2）（8 分）根据题意，该文件的最后一个字节的字节偏移量为 18000000，而 18000000/512 的商为 35156，因此该文件的最后一块的逻辑块号为 35156。由于 10+170+170*170≤35156 ‹10+170+170*170+170*170*170 故该文件不仅需要使用 10 个直接地址项，还需要一次、二次、三次间接项。又因为 35156-（10+170+170*170）=6076。6076/（170*170）得到商为 0，余数为 6076，得知该文件在三次间接地址时还需要 1 个二次间接地址块；而余数 6076/170 得到商为 35、余数为 126，可知该文件在三次索引地址时还需要 36 个一次间接块，因次该文件需要：

三次间接块 1 个；

二次间接块 1+1=2 个；

一次间接块 36+170+1=207 个

数据块：（35*170+127）+170*170+170+10=35157 个。

故共需要 35367 个。

7、答:（共 10 分）

（1）因为文件一次写入不更改，采用连续存储结构更合适。寻道距离短且支持随机读取。（4 分）

（2）FCB 中需要（起始块号、块数）或者（起始块号、结束块号）字段（3 分）

（3）采用集中存放效率更好，因为查找文件只需访问 FCB 所在盘块。（3 分）