

2020-2021-1 B 答案及评分标准

一、选择题（每题 1 分，共 25 分） 得分：

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

二、综合题：

1、（本小题 12 分）

答：（1）这是典型的生产者和消费者问题，只对典型问题加了一个条件，只需在标准模型上新加一个信号量，即可完成指定要求。设置四个变量 mutex1、mutex2、empty 和 full，mutex2 用于一个控制一个消费者进程 一个周期(10 次)内对于缓冲区的控制，初值为 1，mutex1 用于进程单次互斥的访问缓冲区，初值为 1，empty 代表缓冲区的空位数，初值为 0，full 代表缓冲区的产品数，初值为 1000，具体进程的描述如下：（描述分析 2 分）

（2） #define N 1000 （定义 1 分）

#define M 10

#define MAXLEN 80

typedef struct{

 产品；

 }buffer；

semaphore mutex1={1,NULL};

semaphore mutex2={1,NULL};

semaphore empty={N,NULL};

semaphore full={0,NULL};

Message buffers[N];

int in =0, out=0;

buffer temp_p;

buffer temp_c[M];

cobegin

program produceri （4 分）

{

 while(1){

 produce a message in temp_p;

 P(empty);

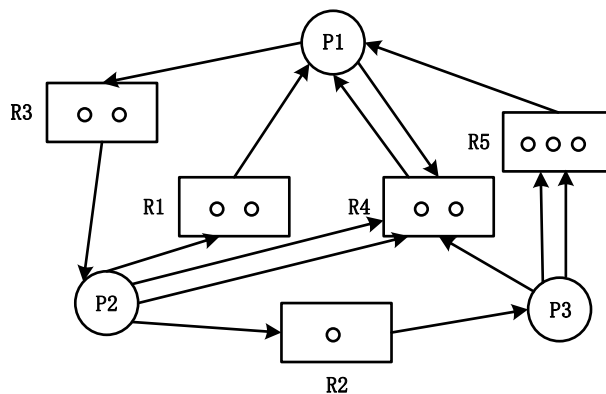
 P(mutex1);

```

        buffers[in] = temp_p
        in =( in +1)%N;
        V(mutex1);
        V(full);
    }
program consumerj (5分)
{
    while(1){
        P(mutex2);
        for(int i=0;i<10;i++){
            {
                P(full);
                P(mutex1);
                temp_c[i] = buffers[out];
                out =(out+1)%N;
                V(mutex1);
                V(empty);
                consume the message in
                temp_c[i];
            }
            V(mutex2);
        }
    }
}
coend

```

2、(本小题 10 分)



(1) 本小题共 4 分，是否发生死锁 2 分，判断过程 2 分
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 //

(2) 本小题共 6 分，数据结构定义 2 分，伪代码描述 4 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 $n \times m$ 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 $n \times m$ 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 $i=0,1,2,\dots,n-1$ ，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 ≤ 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 ($i=0,1,2,\dots,n-1$)，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 (Requesti ≤ 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 7 分)

(1) 打开谷歌地球应用程序时，操作系统要把程序的代码和数据读入到内存中，然后执行这个程序；(2 分)

(2) 用户单击界面中地球表面区域时，操作系统接收到这个“单击”信息，并将这个“单击”信息转交给应用程序；(2 分)

(3) 应用程序将用户单击的区域信息打包，发送给服务器；服务器查找到相关数据，如地形信息等，再反馈给用户端的应用程序；(2 分)

(4) 应用程序利用操作系统完成数据在显示器上的显示工作。(1 分)

4、(本小题 10 分)

(1) 由于计算机的逻辑地址空间和物理地址空间均为 64KB=2¹⁶B，按字节编址，且页(块)的大小为 1KB=2¹⁰B，所以计算机的逻辑地址结构和物理地址结构均为：

页(页框)号 (6 位)	页(块)内偏移量 (10 位)
--------------	-----------------

17CA H=(0001 0111 1100 1010)₂，所以 17CAH 对应的页号是(000101)₂=5。(3 分)

(2) 若采用先进先出(FIFO)置换算法，则置换装入时间最早的页，故 0 号页被置换，将 5 号页装入 7 号页框，所以 17CA H 对应的物理地址为(0001 1111 1100 1010)₂=1FCA H。(3 分)

(3) 若采用时钟 (CLOCK) 置换算法, 则从当前指针指示页框开始查找, 若其中页的访问位为 0, 则置换该页, 否则将访问位清零, 并将指针指向下一个页框, 继续查找。由于初始时内存中的 4 个页的访问位均为 1, 因此, 前 4 次查找并未找到合适的页, 但查找时已将对应页的访问位清零, 第 5 次查找时, 指针重新指向 2 号页框, 其中存放的 2 号页的访问位为 0, 故置换该页, 将 5 号页装入 2 号页框, 所以 17CA H 对应的物理地址为 (0000 1011 1100 1010)₂=0BCA H。(4 分)

5、(本小题 12 分)

(1)

FIFO 算法 (1 分)

1	2	2	2	4	4	4
4	4	5	5	5	2	2
3	3	3	1	1	1	5
	缺 页	缺 页	缺 页	缺 页	缺 页	缺 页

缺页 6 次, 缺页率为 6/12=50% (1 分)

LRU 算法 (1 分)

1	1	1	1	1	1
4	2	2	4	4	4
3	3	5	5	2	5
	缺 页	缺 页	缺 页	缺 页	缺 页

缺页 5 次, 缺页率为 5/12=41.67% (1 分)

(2) CLOCK 置换算法, 缺页 7 次 (2 分), 缺页率: 7/12=58% (2 分)

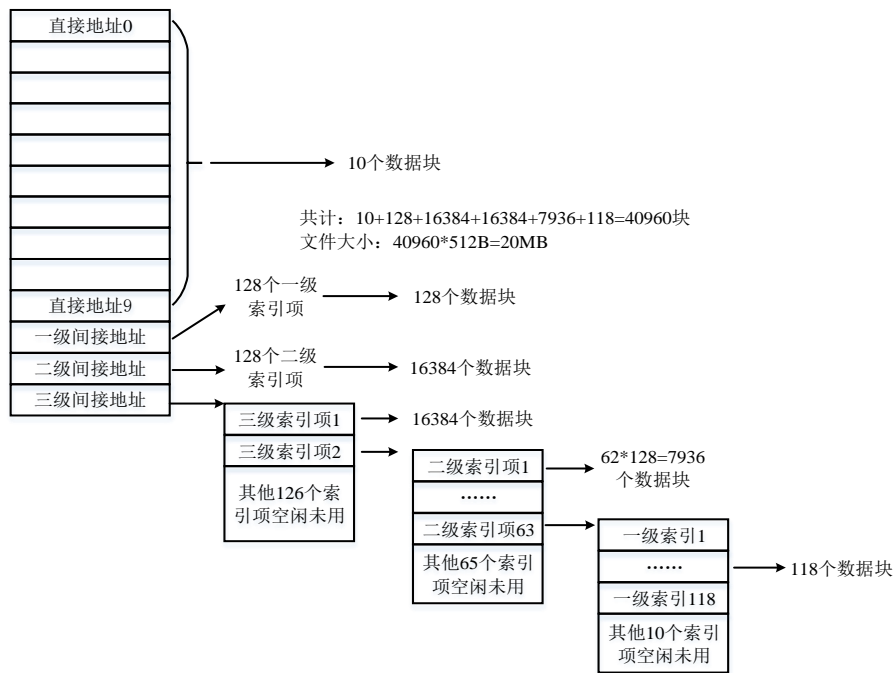
(3) FIFO 算法实现简单, 但未考虑页面的使用频率, 导致缺页率偏高; (2 分)

LRU 算法考虑页面的使用频率, 缺页率较低。(2 分)

6、(本小题 12 分)

(1) 文件的最大长度: $512 * (10 + 128 + 128 * 128 + 128 * 128 * 128)$ 字节 (3 分)

(2) 能正确画出结构图 5 分, $1 + (1 + 128) + (1 + 2 + 128 + 63) = 324$, 得到正确结果 4 分。



7、(本小题 12 分) 某磁盘大小为 64GB，磁盘上的磁盘块大小为 4MB，从 0 开始编号，每个磁道 16 个磁盘块。某文件顺序存储在 8 个磁盘块上，该 8 个磁盘块号分别是 100，160，280，320，90，590，620 和 230，且该文件的目录项所占的磁盘块号是 150，若最后一次磁盘块的访问是 60 号磁盘块。

(1) 若采用隐式链接，试计算读取该文件的寻道距离。

(2) 若采用单级索引分配方法，索引表存储在磁道号为 10 的磁盘块上，索引表表项占 4B。试计算读取该文件的寻道距离。

答案：

(1) 磁盘块个数: $64GB/4MB=16K$ (2 分)

文件占用的磁道号: $100/16=6$; $160/16=10$; $280/16=17$; $320/16=20$; $90/16=5$;

$590/16=36$; $620/16=38$; $230/16=14$; (2 分)

目录项所占磁道号: $150/16=9$; (2 分)

当前磁头所在磁道号: $60/16=3$;

寻道顺序: 3—9—6—10—17—20—5—36—38—14 (2 分)

(2) 寻道顺序: 3—9—10—6—10—17—20—5—36—38—14 (4 分)