

# 考试科目名称 操作系统 (A 卷) 参考答案

考试方式: 闭卷 考试日期 2020 年 8 月 18 日 教师 \_\_\_\_\_

系 (专业) \_\_\_\_\_ 年级 \_\_\_\_\_ 班级 \_\_\_\_\_

学号 \_\_\_\_\_ 姓名 \_\_\_\_\_ 成绩 \_\_\_\_\_

题号	一	二
分数		

得分	
----	--

 一、综合题 (共 42 分)

1. 假设一个计算机系统具有如下性能特征: (6 分)

- ✓ 处理一次中断, 平均耗用 1ms;
- ✓ 一次进程调度, 平均需要 2ms;
- ✓ 将 CPU 分配给选中的进程, 又需要平均 1ms。

再假设其定时器芯片每秒产生 100 次中断。试分析:

- (1) 操作系统将百分之几的 CPU 时间用于时钟中断处理;
- (2) 如果操作系统采用轮转法 (Round-Robin) 调度, 10 个时钟中断为 1 个时间片。那么, 操作系统将百分之几的 CPU 时间用于进程调度 (包括调度、分配 CPU 和引起调度的时钟中断处理时间)。

**参考答案:**

(1) 因为 1 秒钟中断 100 次, 所以中断周期是 10 ms。在每个周期内需要花费 1ms 进行中断处理, 因而系统将 CPU 时间的 10% 用于时钟中断处理。

(2) 轮转调度中, 每 10 个时钟周期为一个时间片, 因此时间片长度为 100ms。系统将 CPU 时间的 13% 用于进程调度, 即

$$(1\text{ms} \times 10 + 2\text{ms} + 1\text{ms}) / 100\text{ms} = 13\%$$

2. 假设一个单 CPU 系统, 以单道方式处理一个作业流, 作业流中有两道作业, 其占用 CPU 时间、输入卡片数、打印输出行数如下表所示, 其中卡片输入机速度为 1000 张/min (平均), 打印机速度为 1000 行/min (平均), 忽略读写盘时间。试计算: (6 分)

作业号	占用 CPU 计算时 (min)	输入卡片张数	输出行数
1	3	100	2000
2	2	200	600

(1) 不采用 SPOOLing 技术, 计算这两道作业的总运行时间 (从第一个作业输入开始, 到最后一个作业输出完毕);

(2) 如果采用 SPOOLing 技术, 计算这两道作业的总运行时间。

**参考答案:**

(1) 作业 1 输入 100 张卡片花 6 秒, 计算花 3 分, 打印 2000 行花 2 分。故合计花 5 分 6 秒。作业 2 输入 200 张卡片花 12 秒, 计算花 2 分, 打印 600 行花 36 秒。故合计花 2 分 48 秒。不采用 SPOOLing 技术, 两道作业的总运行时间=7 分 54 秒。

(2) 如果采用 SPOOLing 技术, 由于预输入和缓输出时间与其他任务重迭进行, 这两道作业的总运行时间应=5 分。

3. 在银行家算法中, 若出现下述资源分配情况, 试问: (8 分)

(1) 该状态是否安全?

(2) 如果进程 P2 提出请求  $Request_2(1, 2, 2, 2)$  后, 系统能否将资源分配给它?

(3) 试证明: 在  $n$  个进程共享  $M$  个资源的系统中, 如果对  $i=1, 2, \dots, n$ , 有  $Need(i) > 0$ , 且所有最大需求量之和小于  $m+n$ , 则该系统不会产生死锁。

	Allocation	Need	Available
P0	0032	0012	1622
P1	1000	1750	
P2	1354	2356	
P3	0332	0652	
P4	0014	0656	

参考答案:

(1) 安全; (需要给出分析过程)

(2) 不可以; (需要给出分析过程)

(3) 证明: 设每个进程对共享资源的最大需求量为  $x$  ( $0 < x \leq m$ ), 由于每个进程最多申请使用  $x$  个资源, 在最坏情况下, 每个进程都得到了  $(x-1)$  个资源, 并且都需申请最后一个资源。这时系统剩余资源数为:  $m-n(x-1)$ 。只要系统还有一个资源可用, 就可使其中的一个进程获得所需的全部资源, 该进程运行结束后释放出它所占用的资源, 其他进程的资源需求也可得到全部满足。因此, 当  $m-n(x-1) \geq 1$  时, 即  $x \leq (m+n-1)/n$  时系统不会发生死锁。进而可得系统中所有进程最大需求量之和  $nx \leq (m+n-1)$  时系统不会发生死锁。该题中, 所有进程最大需求量之和小于  $m+n$ , 所以, 该系统是死锁无关的。

4. 某系统采用可变分区存储管理技术, 某时刻在内存中有三个空闲区, 其首地址和大小分别是: 空闲区 1 (100KB, 10KB), 空闲区 2 (200KB, 20KB), 空闲区 3 (300KB, 15KB)。现有如下作业序列: 作业 1 要求 15KB, 作业 2 要求 16KB, 作业 3 要求 10KB。要求: (6 分)

(1) 画出该时刻内存分布图;

(2) 有首次适应算法和最佳适应算法分配, 画出此时的内存队列结构;

(3) 哪些算法能将该作业序列装入内存 (给出简要的分配过程)。

参考答案:

J1 要求 15KB、J2 要求 16KB、J3 要求 10KB

(1) 采用 first fit

J1 占用 200KB 开始的空闲区

J3 占用 100KB 开始的空闲区

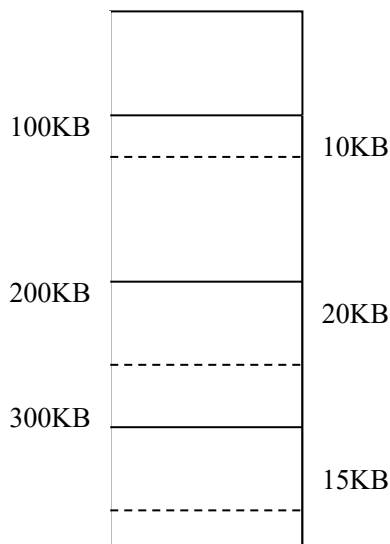
J2 无法分配

(2) 采用 best

J1 占用 300KB 开始的空闲区

J2 占用 200KB 开始的空闲区

J3 占用 100KB 开始的空闲区



5. 在文件系统中，会出现文件系统不一致性的现象：（6分）

（1）请简要解释这种现象产生的原因以及问题的严重性；

（2）为解决文件系统的不一致性问题，常采用一个实用程序检查文件系统。在进行了块的不一致性检查后，得到如下表所示的结果，请解释该文件中出现的每一种错误，并给出处理方法。

块号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
空闲块	1	1	1	0	1	0	1	1	1	0	1	0	1	1	1	2
分配块	0	0	1	1	0	1	0	0	0	2	0	0	0	0	0	0

**参考答案：**

（1）如果管理空闲块及分配/回收程序出现故障，则会导致不一致性错，或丢失盘块(均为0的情况)，或破坏信息(如重复分配)。

（2）正常情况，空闲块计数与分配块计数应互补。但表中有不一致处，块2重复出现，向系统报告出错。块9分配块重复出现，盘块被分了二次，严重错误，立即报告。块11两组计数中均未出现，报告给系统但影响不大，只是该块被隔离了。块15空闲块出现两次，可删去一个块15号。

6. 假设一个可移动磁头的磁盘具有200个磁道，其编号为0~199，当它刚结束了125道的存取，正在处理143道的服务请求，假设系统当前I/O请求队列如下：86，147，91，177，94，150，102，175，130，（10分）

（1）试对最短查找时间优先调度（SSTF）、扫描法（SCAN）、单向扫描（循环扫描）（C-SCAN）等3种磁盘I/O调度算法，分别给出对应的移动臂移动的总量；

（2）请用信号量或管程实现SCAN磁盘调度算法。

**参考答案：**

（1）磁头当前位于143道，且方向是向大的方向(对SCAN和C-SCAN有用)：

SSTF：143,147,150,130,102,94,91,86,175,177。（总量略）

SCAN：43,147,150,175,177,130,102,94,91,86。（总量略）

C-SCAN：143,147,150,175,177,86,91,94,102,130。（总量略）

（2）管程实现示例：（信号量实现示例略）

```

type diskschedule=monitor
var headpos:integer;
    direction(up,down);
    busy:boolean;
    S:array[0...199] of condition;
DEFINE request,return;
USE wait,signal,check,release;
procedure request(var dest:integer) {
    check(IM);
    if busy then wait(S[dest],IM);
    busy:=true;
    if (headpos<dest) or (headpos=dest&direction=up)
        then direction:=up;
        else direction:=down;
    headpos:=dest;

```

```

        release(IM);
    }
procedure return {
    var i:integer;
    check(IM);
    busy:=false;
    if direction=up
    then begin
        i:=headpos;
        while( i<200 &S[i]=0) do i:=i+1;
        if i<200 then signal(S[i],IM);
        else {
            i:=headpos;
            while(i≥0 &S[i]=0) do i:=i+1;
            if i≥0 then signal(S[i],IM);
        }
    }
    else {
        i:=headpos;
        while( I>0 &S[i]=0) do i:=i-1;
        if i≥0 then signal(S[i],IM);
        else {
            i:=headpos;
            while(i<200 &S[i]=0) do i:=i+1;
            if i<200 then signal(S[i],IM);
        }
    }
    release(IM);
    {
        headpos:=0;
        direction:=up;
        busy:=false;
        S:=0;
    }
main() {
cobegin
    process visit
    var k:integer;
    begin
        ...
        call diskschedul. Request(k);
        ...
        访问第k个柱面;
        ...
    end
end

```

```

        call diskschedul. Return;
        ...
    }
coend.

```

得分	
----	--

二、编程题（8 分）

1. 一个主修动物行为学、辅修计算机科学的学生参加了一个课题，调查花果山的猴子是否能被教会理解死锁。他找到一处峡谷，横跨峡谷拉了一根绳索（假设为南北方向），这样猴子就可以攀着绳索越过峡谷。只要它们朝着相同的方向，同一时刻可以有多只猴子通过。但是如果在相反的方向上同时有猴子通过则会发生死锁（这些猴子将被卡在绳索中间，假设这些猴子无法在绳索上从另一只猴子身上翻过去）。如果一只猴子想越过峡谷，它必须看当前是否有别的猴子在逆向通过。请使用信号量写一个避免死锁的程序来解决该问题。

**参考答案：**

```

var mutex, smutex, nmutex, smonkeycount, nmonkeycount: semaphore;
    smonkeycount:=0    /*由南向北攀绳索的猴子数量;
    nmonkeycount:=0    /*由北向南攀绳索的猴子数量;
    smutex:=nmutex:=1; /*南/北方向猴子间互斥信号量
    mutex:=1;          /*绳索互斥信号量
cobegin
    process southi (i=1,2,3,...) {
        P(smutex);
        if smonkeycount=0 then P(mutex);
        smonkeycount :=smonkeycount+1;
        V(smutex);
        Cross the cordage;
        P(smutex);
        smonkeycount :=smonkeycount-1;
        if smonkeycount=0 then V(mutex);
        V(smutex);
    }
    process northj (j=1,2,3,...) {
        P(nmutex);
        if nmonkeycount=0 then P(mutex);
        nmonkeycount :=nmonkeycount+1;
        V(nmutex);
        Cross the cordage;
        P(nmutex);
        nmonkeycount :=nmonkeycount-1;
        if nmonkeycount=0 then V(mutex);
        V(nmutex);
    }
coend.

```