



2. 一个具有二道作业的批处理系统，作业调度采用优先数调度算法（优先数越小，优先级越高），进程调度采用时间片轮转调度算法。（6分）

作业名	到达时间	估计运行时间（分钟）	优先数
Job1	9:00	30	4
Job2	9:10	25	1
Job3	9:30	15	5
Job4	9:40	10	3
Job5	9:50	20	2

试回答如下问题：

- （一） 列出所有作业进入主存时间与结束时间；
- （二） 计算作业的平均周转时间和平均带权周转时间；
- （三） 简要讨论批处理系统和分时系统的区别。

（一） Job1, 9: 00 – 9: 50。

Job2, 9: 10 – 10: 00。

Job3, 10: 20 – 10: 40。

Job4, 10: 00 – 10: 20。

Job5, 9: 50 – 10: 30。

（二） 平均周转时间 =  $(50 + 50 + 70 + 40 + 40) / 5 = 50$ 。

平均带权周转时间 =  $(50 / 30 + 50 / 25 + 70 / 15 + 40 / 10 + 40 / 20) / 5 = 43 / 5 = 2.8667$ 。

（三） 批处理系统：用户脱机工作、成批处理作业、多道程序运行(早期为单道程序运行)、作业周转时间长。

分时系统：同时性，多个用户同时；独立性，每个用户相对独立；及时性，快速响应用户请求；交互性，用户联机工作。

3. 设系统中有 P0-P4 共 5 个进程，A-D 四类资源，系统资源分配情况如下表所示：

进程	已分配	最大需求	空闲资源
	A B C D	A B C D	A B C D
P0	0 0 1 2	0 0 1 3	1 4 2 1
P1	1 0 0 0	1 7 5 1	
P2	1 2 1 2	2 2 1 6	
P3	0 1 1 2	0 1 3 3	
P4	0 0 1 0	0 3 3 0	

试回答如下问题：（6 分）

（一）当前系统状态是否安全（银行家算法）？

（二）假设此时进程 P1 提出资源申请（0，1，0，1），能够满足 P1 进程的资源申请吗（分配拒绝法）？

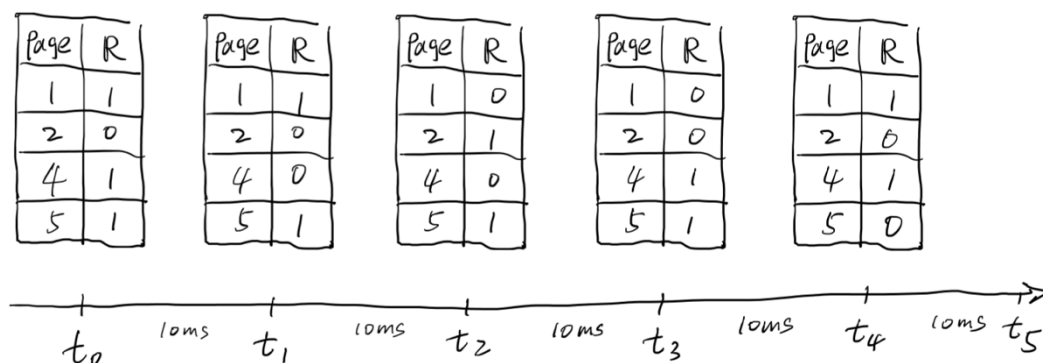
（三）假设系统满足了（二）中的请求，之后系统中各进程又发生了如下资源申请：P0 申请(0，0，0，1)，P1 申请(0，4，0，0)，P2 申请(1，0，0，1)，P3 申请(0，0，1，1)，P4 申请(0，1，1，0)，那么系统中是否有死锁发生，如果发生了死锁，则死锁涉及了哪些进程（死锁检测算法）？

（一）安全，存在安全序列 P0-P4-P3-P2-P1。

（二）不能，满足 P1 的请求之后不存在安全序列，进程死锁。

（三）发生了死锁，涉及进程 P0,P1,P2,P3。

4. 一个 32 位系统的计算机，具有 1GB 物理内存，其上的操作系统采用请求式分页存储管理技术，页面大小为 4KB，页表项占 4B。系统中某进程固定分配了四个页框，已驻留页面 1、2、4、5，其页面访问情况如下图所示：（8 分）



试回答如下问题：

- （一）如果采用一级页表，则页表最大包含多少个页表项？
- （二）如果采用二级页表，则 32 位逻辑地址该如何划分（页目录号、页号、页内偏移各占多少二进制位，给出计算过程）？相比一级页表，二级页表有何优势？
- （三）如果在  $t_4$  和  $t_5$  之间的某个时刻，进程需要访问逻辑地址 12300(十进制)，是否会发生缺页异常？如果发生缺页异常，请考虑采用最近没有使用替换算法(NRU)、最不常用替换算法(LFU)、老化算法(Aging)，分别会替换掉哪个页面？（设页面统计的时钟间隔为 10ms）
- （四）为了避免频繁缺页引发“颠簸”，系统中同时运行的进程需要满足什么条件？
- （五）假如统计结果表明进程平均占用 8MB 左右的内存空间，页面大小设为 4KB 是否合理（给出分析过程）？

- （一） $2^{32}/(4 \times 2^{10}) = 2^{20}$  个页表项。
- （二）页面大小  $4 \times 2^{10} = 2^{12}$ ，因此页内偏移 12 位，二级页表项可存放  $4 \times 2^{10}/4 = 2^{10}$  个，因此页目录号占 10 位，最后页目录号占  $32-12-10=10$  位。
- （三） $12300 = (0000\ 0000\ 0000\ 0000\ 0011\ 0000\ 0000\ 1100)_2$  B，因此页号为(11) B = 3，3 号页面不在页框中，因此缺页。NRU：2 或 5 号页面；LFU：2 号页面；Aging：2 号页面。
- （四）内存中工作集总尺寸不能超过 1GB（即物理内存大小）。
- （五）最佳页面尺寸应使函数  $f(p) = se/p + p/2$  取到最小值。进程平均占用空间  $s = 8 \times 2^{20}$  B，页表项大小  $e = 4$  B， $f(p) = 2^{25}/p + p/2$ ，求导  $f'(p) = 1/2 - 2^{25}/p^2 = 0$ ，得  $p = 2^{13}$  B，即 8KB。因此不合理。

5. 设某 UNIX 系统，文件系统的每个 inode 包含 10 个直接索引项和一、二、三级间接索引项各一个，物理块大小为 512B，每个索引项占 4B。每个目录项占 16B（包含文件名和 inode 号）。操作系统启动时，文件系统根目录内容预先加载至内存中。test 目录下仅有文件 demo.dat，demo.dat 文件大小为 6050B。（8 分）

试阅读如下代码并回答问题：

```
1. #include <stdio.h>
2. #include <unistd.h>
3. #include <fcntl.h>
4.
5. int main(void)
6. {
7.     int fd;
8.     char buffer[5000];
9.
10.    fd = open("/test/demo.dat", O_RDONLY);
11.    if (fork() == 0)
12.    {
13.        sleep(1);
14.        printf("child: %ld \n", read(fd, &buffer, 5000));
15.    } else {
16.        printf("father: %ld \n", read(fd, &buffer, 2000));
17.    }
18. }
```

- （一）上述代码若能正常执行完成，屏幕输出的内容是什么？
- （二）执行到第 10 行代码 open 函数时，试描述文件系统大致的工作过程，此过程中共计读取几次磁盘？
- （三）在不考虑磁盘缓冲的情况下，上述代码中父、子进程读文件时，文件系统各读取了几个物理块？其中有几个物理块父、子进程都读取了？
- （四）demo.dat 文件占多少磁盘空间？如果 demo.dat 文件需要不断增加内容，此文件理论上能增加到多大？
- （五）简述在 Shell 中执行“link /test/demo.dat /demo.dat”时，文件系统的主要操作步骤，并简要讨论此类文件共享方式的优缺点。

（一） father: 2000

child: 4050

- （二）共读取 3 次（1、test 目录 inode；2、demo.dat 目录项；3、demo.dat 的 inode）。
- （三）父进程 4 个（1-4），子进程 9 个（4-12）。  
有 1 个物理块父、子进程都读取了。
- （四）6656B。最大能增长到 1056837KB。
- （五）1、检索目录找到/test/demo.dat 所指向文件的索引节点编号；2、再次检索目录找到 /demo.dat 所指文件的父目录文件，并把上一步中获得的索引节点编号与/demo.dat 构成一个新目录项，加入到此目录文件中；3、将对应索引节点连接计数分量 i\_nlink 加一。

6. 设有一个包含了 32 个磁头(编号 0-31)和 200 个柱面(编号 0-199)的磁盘, 每磁道扇区数 200 个(编号 0-199), 每个扇区 512B, 磁盘的转速为 7200rpm (转每分钟)。试回答如下问题: (7 分)

- (一) 磁盘的容量是多少? 磁盘的读写速率是多少? 平均旋转等待时间是多少?
- (二) 如果采用位示图方式管理磁盘的空闲扇区, 位示图需要占多少扇区?
- (三) 若此时磁头位于 20 号柱面, 刚刚完成 19 号柱面访问, 依次到来如下柱面请求: 12, 25, 78, 32, 101, 175, 58, 197, 92, 如果采用先来先服务、最短查找时间优先、电梯调度算法, 请分别给出对应的柱面请求序列和移动臂移动的总量。
- (四) 为提高输入输出效率, 是否有必要对相同柱面不同扇区和不同磁头号的请求进行合并操作? (简要说明原因)

- (一) 磁盘容量  $640000\text{KB} = 625\text{MB}$ 。读写速率:  $12000\text{KB/s}$  ( $720000\text{KB/min}$ )。  
平均旋转等待时间:  $1 / 240\text{s} = 25 / 6\text{ms}$ 。
- (二)  $160000\text{B}$ , 313 扇区。
- (三) 先来先服务: 20-12-25-78-32-101-175-58-197-92, 移动 624 个柱面。  
最短查找时间优先: 20-25-32-12-58-78-92-101-175-197, 移动 217 个柱面。  
电梯调度算法: 20-25-32-58-78-92-101-175-197-12, 移动 362 个柱面。
- (四) 有必要。柱面间移动耗时较多, 将相同柱面请求合并处理, 能减少移动臂移动总量。

得分	
----	--

## 二、编程题（8）

1. 试用管程实现读者写者问题，即有若干读进程和写进程访问同一个文件，为防止文件读写错误，要求：1) 允许多个读进程同时读取文件内容；2) 任何时候只允许一个写进程修改文件内容；3) 写进程完成前不允许读进程读文件；4) 等所有已经开始读的进程完成后才允许写进程写。（需要定义管程 Monitor，引入适当的条件变量 Condition，以及条件变量上的 wait 和 signal 操作、定义读写进程并调用管程函数来实现上述同步）

```
Type RWMonitor = Monitor
{
    Cond write, read;
    int read_count = wait_count = write_count = 0;
    Define Start_write, End_write, Start_read, End_read;
    Procedure Start_write()
    {
        while(read_count > 0 or write_count > 0) wait(write);
        write_count ++;
    }
    Procedure End_write()
    {
        write_count --;
        while(wait_count > 0) {wait_count --;signal(read);}
        signal(write);
    }
    Procedure Start_read()
    {
        if(write_count > 0) {wait_count ++;wait(read);}
        read_count ++;
    }
    Procedure End_read()
    {
        read_count --;
        signal(write);
    }
}
```

```
Func Writer()
{
    RWMonitor.Start_write();
    Write_file();
    RWMonitor.End_write();
}
```

```
Func Reader()
{
    RWMonitor.Start_read();
    Read_file();
    RWMonitor.End_read();
}
```