

A 卷参考答案及评分标准:

一、选择题（每题 1 分，共 25 分）

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

二、综合题（共 75 分）

1、(6 分) 每个例子 2 分。

(1) 分时系统中的 CPU, 通过分时的原则把一个物理 CPU 虚拟为多台逻辑 CPU, 供多个终端用户使用; (2) 虚拟存储器: 通过只装入进程的部分内容, 从逻辑上对内存容量进行了扩充; (3) 虚拟设备技术: 通过 SPOOLing 技术, 把一台物理打印机虚拟成多台逻辑打印机, 共多个进程“同时”请求使用。

其他的应用举例合理即可。

2、(12 分)

(1)、(2): 每个调度算法每种时间各 2 分, 共 8 分

算法		P1	P2	P3	P4	P5	平均
RR	周转时间	19	2	7	4	14	9.2
	带权周转时间	1.9	2	3.5	4	2.8	2.84
优先级	周转时间	16	1	18	19	6	12
	带权周转时间	1.6	1	9	19	1.2	6.36

(3) 4 分:

性能分析 2 分: 静态优先级实现简单, 且能基本保证紧迫性任务得到及时处理; 但会造成低优先级任务长时间得不到调度运行, 导致产生饥饿现象; 因为是非抢占式, 所有不能保证高优先级任务得到最及时的处理。

改进 2 分: 比如让进程优先级随着等待时间延长动态提升; 让高优先级任务随着连续运行时间延长降低优先级; 采用抢占式优先级调度等, 合理即可。

3、9分：

(1) (3分)

4G 文件占用的盘块数量=4GB/4KB=1M 个

一个盘块可放置链表项共：4KB/4B=1K 个

该文件的链表项最少占用盘块数量：1M/1K=1K 个，最多分数在 1M 个块中（即每个 FAT 表块中 1 项）

最少需要访问磁盘次数为：1K+1 次。

最多需要访问磁盘次数为：1M+1 次

(2) (4分)

8MB 的文件占用的盘块数量=8MB/4KB=2K 块

需要 10 个直接索引(10 块), 1 个 1 级索引(1K 块), 1 个 2 级索引块(2K-1K-10)

Ext2 系统中，8MB 文件的最后一块使用了 2 级索引部分，因此需要读取 2 级索引、1 级索引、数据块，共访问 3 次磁盘。

(3) (2分)

索引文件的索引表应该存放在文件的头部。索引文件需要先查询索引表，再查询记录。由于文件绝大部分内容为记录数据，并使用 2 级索引保存，因此可认为访问记录时访问磁盘的次数为 3 次，可通过减少索引表的查询时间来提高访问效率。在 Ext2 系统中，文件的起始若干块会使用直接索引或 1 级/2 级索引，查询的速度较快，因此将索引表放置在文件头部有助于提高索引文件的访问速度。

4、12 分：

(1) (4分)

1) 3 分：一个目录盘块能够保存的目录项数量为 $4KB/(4+4)=512$ 项，因此 800 个文件需要 2 个目录盘保存。

打开文件时，首先由文件名计算 HASH 值，并通过该 HASH 值查找目录项得到文件的 FCB 编号，而目录项存放在两个块中，因此平均需要读取 $(1+2)/2=1.5$ 次；然后将 FCB 编号指向的 FCB 读入内存：1 次。

打开一个文件访问磁盘次数的均值= $1.5+1=2.5$ 次。

2) 1 分：如果 HASH 值可能重复，可选策略之一是对匹配成功的目录项读取 FCB，并对 FCB 中的文件名做字符串匹配，若字符串匹配失败，则在目录盘块中寻找下一个匹配项，并重复过程。

(2) (4分)

文件的打开操作具体是将文件的 FCB 载入内存。

在 root 的目录盘块中寻找 home 目录项（无需读盘）→载入 home 的 FCB→载入 home 目录文件的盘块并寻找 user 目录项→载入 user 的 FCB→载入 user 目录盘块并寻找 test.pak 目录项→载入 test.pak 的 FCB。共需要访问 5 次磁盘。

(3) (4分)

由于 user 目录盘块已经在内存中，因此可以从内存中直接搜索 work.pak 的目录项，并载入其 FCB，这里需要 1 次磁盘访问。 // 2 分

对于 3 级索引文件系统，读取文件的任意盘块需要 4 次磁盘访问。

因此一共需要 5 次磁盘访问。 //2 分

5、(12 分)

(1) 一个进程在内存中有 3 个页，且其中的 1 页用于存放程序，则用作存放数据的页面只有 2 个。数组 A 中有 10000 个整数，每页存放 200 个整数，故数组占用空间 50 页。

程序 A：程序 A 对数组 A 的访问是按行进行，即按照存储顺序进行。所以只有在访问每一页的第一个数据时会缺页，同一页的其他数据不会发生缺页，故程序 A 的缺页次数为 50。 //3 分

程序 B：程序 B 对数组 A 的访问是按列进行，而数组 A 每行有 100 个数据，每页可以存放 200 个数据，因此每页中有 2 个数据属于同一列，每次缺页中断调进一页时，只有其中的 2 个数据被赋予 0 值，即程序 B 对数组 A 每两次访问会遇到一次缺页。 $200/100=2, 10000/2=5000$, 故缺页次数为 5000 次。 //3 分

(2) 2 分：程序 A 中数据访问的局部性特征比程序 B 好。

(3) 4 分：合理即可：1) 合理设计程序，提高其内存访问的局部性；2) 引入页面缓冲置换的思想，让淘汰的页面再次驻留内存的机会。

6、(12 分)

//信号量设置正确 3 分：

信号量设置： semaphore wmutex, rmutex, maxcount=1,1,1000

整形变量： int count=0

//教师代码： 3 分

教师 i: {

P(wmutex);

登记成绩;

V(wmutex);

}

//学生代码： 6 分

学生 i: {

P(maxcount);

P(rmutex)

If (count==0) then P(wmutex);

count++;

V(rmutex);

查询成绩;

P(rmutex)

count--;

If (count==0) then V(wmutex);

V(rmutex);

V(maxcount);

}

7、12 分：

(1) (4 分) 页面大小为 4KB，则页内偏移为 12 位； //1 分

页表项大小为 8B，则每页可容纳 $4KB/8B=512$ 项= 2^9 项，

多级页表的级数= $(48-12)/9=4$ ，采用 4 级页表 //3 分

(2) (4 分) 若快表没有命中，则需要查找 4 级页表，所以内存平均访问时间为：

$$(10+100) *95\% + 5\% * (10+4*100+100) = 359.5 \text{ ns}$$

(3) (4 分) 系统采用 48 位虚拟地址，则虚拟地址空间为 2^{48}B ，每段最大为 4GB，则最多段数为： $2^{48}\text{B} / 4\text{GB} = 2^{16} = 65536$ 段； //2 分

每个段最大为 4GB，则段内地址位数为 32 位，段内页表级数为： $(32-12) / 9 = 3$ 级

段内采用 3 级页表。 //2 分