

## 试卷 A 参考答案及评分标准

### 一、选择题（每题 1 分，共 25 分）

得分：

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

### 二、综合题（共 75 分）

1、（6 分）缓和速度差异，减少 CPU 中断次数，均衡负载。

量化比较有道理即可

评分标准：每一要点 2 分，无详细说明扣 0.5 分。

2、（14 分）本题中可设置 4 对信号量 in1\_empty 和 in1\_full、in2\_empty 和 in2\_full、out1\_empty 和 out1\_full、out2\_empty 和 out2\_full，分别对应于相应的空闲缓冲区和缓冲区中的数据，它们的初始值为 1、0，相应的进程可以描述为：

semaphore in1\_empty=1,in1\_full=0,in2\_empty=1,in2\_full=0;

semaphore out1\_empty=1,out1\_full=0,out2\_empty=1,out2\_full=0;

//信号量完全设置正确 3 分，部分正确酌

情给分

main()

cobegin{

input () // 3 分

while (1) {

输入下一个数据;

P(in1\_empty);

将数据存入放到缓冲 IN1 中;

V(in1\_full);

输入下一个数据;

P(in2\_empty);

将数据存入放到缓冲 IN2 中;

V(in2\_full)

}

trans () //5 分

while (1) {

P(in1\_full);

```

    取出缓冲 IN1 中的数据;
    V(in1_empty);
    将取出的数据进行转换;
    P(out1_empty);
    将数据输入缓冲 OUT1 中;
    V(out1_full);
    P(in2_full);
    取出缓冲 IN2 中的数据;
    V(in2_empty);
    将取出的数据进行转换;
    P(out2_empty);
    将数据输入缓冲 OUT2 中;
    V(out2_full);
}
output ( ) // 3 分
while (1) {
    P(out1_full);
    取出缓冲 OUT1 中的数据;
    V(out1_empty);
    将取出的数据打印出来;
    P(out2_full);
    取出缓冲 OUT2 中的数据;
    V(out2_empty);
    将取出的数据打印出来;
}
}coend;

```

3、(10 分)

(1) //3 分

每秒产生 120 个时钟中断，每次中断的间隔时间是： $1/120(\text{秒})=8.3(\text{ms})$

其中中断处理耗时 500us，则开销为： $500\text{us}/8.3\text{ms}=6\%$

(2) //4 分

每一次进程切换需要 1 次调度、1 次切换，所以需要耗时：

$1\text{ms}+2\text{ms}=3\text{ms}$

每 24 个时钟为一个时间片： $=24*8.3\text{ms}=200\text{ms}$ ，一次调度切换所占

CPU 的时间比： $3\text{ms}/200\text{ms}=1.5\%$

(3) //3 分，

为了提高 CPU 的利用率，可考虑减少时钟中断的次数，延长中断的时间间隔；或者将每个时间片的中断数量加大；或者减少中断处理时间等等。

4、(11 分)

(1) //4 分

页面大小： $8\text{KB}=2^{13}\text{B}$ ，所以一个块中可以存放  $2^{10}$  个页表项，即逻辑地址结构中每个中间层页号（外部页内地址）需要占用 10 个二进制位，所以需要的页表级数为：

(64-13)/10=6 (向上取整)

(2) 700ns // 3 分

(3)  $85\% * (10+100) + 15\% * (10+700) = 200\text{ns}$  // 4 分

5、(12 分) // 每小題各 3 分

(1) 访问<0, 4>时, 对应页框号是 21, 因为起始驻留集为空, 0 号页是依次访问的第三个页面, 所以装入第三个空闲页框: 21

(2) 访问<1, 11>时, 对应页框号为 32, 因为 11>10, 所以发生第三轮扫描。页号为 1 的页框在第二轮已经处于空闲页框链表中。此刻该页又被重新访问, 因此应被重新放回到驻留集中。其页框号为 32。

(3) 访问<2, 14>时, 对应的页框号为 41, 因为第 2 页从来没有被访问过, 不在驻留集中, 所以从空闲页框链表中取出链首的页框, 页框号为 41。

(4) 适合, 程序的时间局部性越好, 则从空闲页框链表中被重新取回的机会就越大, 该策略的优势越明显。

6、11 分

(1) // 4 分

找到文件 A 的目录项需要 5 次读磁盘

文件 A 占用磁盘块= $600/2=300$  块, 所以总共需要  $300+5=305$  次

(2) // 4 分

找到文件 A 的目录项需要 5 次读磁盘, 读一块需要 1 次, 总共= $5+1=6$

(3) // 3 分

第一种方法, 引入索引节点, 则一个目录项为 16B, 则一个盘块可以存放  $512/16=32$  个目录项, 与原来的 4 个目录项相比, 读盘次数减少为 1/8, 或者其他合理说明;

第二种方法: 改变文件物理结构, 比如索引链接方法, 一个盘块中存放一个文件的一组盘块号, 再用一个指针指向存放下一组盘块号的盘块, 这样读出一个索引块, 就能得到文件的一组盘块号。

7、(11 分)

(1) 用位图表示磁盘的空闲空间, 假设每个字 32 位: // 3 分

共需:  $16384/32=512$  个字, 位图大小:  $512*4B=2KB$ , 刚好。

(2) 调度顺序为: 120, 150, 90, 50, 30 // 5 分

移动磁道数= $(120-100) + (150-120) + (150-90) + (90-50) + (50-30) = 170$

转速为 6000 转/分, 平均旋转延迟为 5ms, 总延迟时间=25ms;

转速为 6000 转/分, 读取一个磁道上的一个扇区的时间为: 0.1ms

所以总时间= $170+25+0.5=195.5\text{ms}$

(3) // 3 分

采用先来先服务算法 (FCFS) 更高效, 因为 Flash 半导体存储器的物理结构不需要考虑寻道时间和旋转延迟, 可直接按 I/O 请求的先后顺序服务。