

一、单选题 (20 分)

1. 具有同时性、独立性、及时性和交互性特点的操作系统是 (D)

- A. 批处理操作系统 B. 分布式操作系统
C. 实时操作系统 D. 分时操作系统

2. 关于 CPU 调度, 下面的论述中正确的是 (A)

A. 短期调度就是 CPU 调度, 负责从就绪队列中选择进程到 CPU 上运行。

B. 短期调度比长期调度相比执行频率低。

C. 长期调度是指选择某个符合条件的等待状态的进程都就绪队列中。

D. 短期调度算法有运行状态的用户进程负责调用执行。

(作业调度)
长期调度: 从存储设备的缓冲池中选进程进入内存执行, 执行效率低, 可控制多道程序设计的程度, 对 I/O 为主的进程和 CPU 为主的进程组合以保证 CPU 和外设的有效使用。

中期调度: 将进程换入换出 (交换) 以降低

短期调度: 选择进程进入 CPU 执行 (CPU 调度) 执行效率高

3. 下面 4 句话中说明了多道程序设计的优点的有 (C)

(1) 提高 CPU 的利用率

(2) 更好的利用内存

(3) 让用户方便的使用计算机

(4) 让 CPU 和设备并行工作

A. 1 个 B. 2 个 C. 3 个 D. 4 个

多道程序设计的优点,

① 提高 CPU 的利用率

② CPU 效率提升

③ 内存效率提升

④ CPU 和外设并行工作

4. 设从磁盘将一块数据传送到缓冲区经历的时间为 $80\mu s$, 将缓冲区中数据传送到用户区所用时间为 $40\mu s$, CPU 处理数据所用时间为 $20\mu s$, 则处理该数据, 采用单缓冲传送磁盘数据, 系统所用总时间为 (A)

$$80\mu s + 40\mu s = 120\mu s$$

A. $120\mu s$ B. $110\mu s$ C. $150\mu s$ D. $70\mu s$

5. 采用段式存储管理的系统中, 若地址用 24 位表示, 其中 8 位表示段号, 则每段的最大长度是 (B)

地址 24 位, 8 位是段号, 真正表示段内地址的只有 16 位

A. 2^{24} B. 2^{16} C. 2^8 D. 2^{32}

每段最大长度是 2^{16}

6. 设某个链式文件有 6 个逻辑记录组成, 每个逻辑记录的大小与磁盘物理块大小相等, 均为 512 字节, 并依此存放在 45、119、78、99、52、66 号盘块上。若要存取文件的第 1566 逻辑字节处的信息, 问需要访问的物理块号为 (B)。

A. 119 B. 99 C. 78 D. 52

$$1566 \div 512 = 3 \text{ 余 } 30$$

$$\text{由 } 512 \times 2 < 1566 < 512 \times 3$$

∴ 访问的是第 3 块

$$(1566 / 512 \text{ 商 } 3 \text{ 余 } 30)$$

分配给进程的主存页框数
II. 页框的大小
III. 编程技术

7. Spooling 系统提高了 (A) 的利用率。

- A. 独占设备 B. 共享设备 C. 虚拟设备 D. 用户设备

8. 系统某一时刻的进程资源图如图 a 所示, 请判断系统是否发生死锁 (B)。

- A. 有死锁 B. 无死锁 C. 不能确定 有环且单一资源分配

9. 一个进程释放一种资源将有可能导致一个或几个进程 (D)。

- A. 由就绪变运行 B. 由运行变就绪
C. 由阻塞变运行 D. 由阻塞变就绪

10. 关于文件的逻辑结构, 下面的论述错误的有 (A) 个。

- (1) 顺序文件中所有记录按照关键字的值进行排序。✓
(2) 索引顺序文件是在顺序文件上建立了索引—提升文件记录的查找速度。✓
(3) 对于文件建立穷举索引没有意义。✗
(4) 顺序文件在磁盘上连续存储可以得到比较好的存取效率。✓

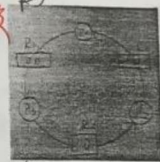
- A. 1 个 B. 2 个 C. 3 个 D. 4 个

二、简答题 (20 分)

1. (6 分) 有一个分页虚拟存储系统, 测得 CPU 和磁盘的利用率如下, 试指出每种情况所存在的问题和可采取的措施。

- (1) CPU 利用率为 13%。磁盘利用率为 97%;
(2) CPU 利用率为 87%。磁盘利用率为 3%;
(3) CPU 利用率为 13%。磁盘利用率为 17%;

- (1) 系统可能出现抖动; 可把暂停部分进程运行
(2) 系统运行正常; 可增加运行进程数以提高资源利用率
(3) 处理器和设备利用率均很低; 可增加并行运行的进程数



6. 下面选项中, 影响缺页中断率的因素是 (D) 、
- 分配给进程的主存页框数
 - 页框的大小

2. (5分) 在文件系统中, 与连续分配方法相比较, 链式分配有何优缺点?

链式分配的优点: 可以充分利用空闲物理块, 减少了磁盘空间碎片
文件的扩展不受磁盘空间大小的限制
缺点: 要寻找某个物理块必须从头找起, 且需要频繁地在非连续的物理块中移动磁头来读取数据

3. (5分) 试述什么是模式切换, 什么是进程切换, 以及两者之间的关系?

模式切换: 由CPU从核心态到用户态, 或从用户态到核心态
进程切换: 是指从一个进程上下文切换到另外的进程上下文

结论: 模式切换不一定导致进程切换, 进程切换一定发生模式切换

④ (4分) 一个有快表的页式存储系统, 设一次内存访问时间为1微秒, 内外存传送一个页面的平均时间为5毫秒, 如果快表命中率为75%, 缺页中断率为10%。忽略快表访问时间。试求内存的有效存取时间。

快表命中率为75%, 缺页中断率10%, 所以内存命中率为15%, 故内存的有效存取时间 = $1 \times 75\% + 2 \times 15\% + (5000 + 2) \times 10\% = 501.25 \mu s$

三、综合题 (50分)

1. (7分) 现有五个进程 P1、P2、P3、P4、P5 共享 A、B、C 这三类资源, 进程对资源的需求量和目前分配情况如下表所示。若系统还有剩余资源数分别为 A 类 2 个, B 类 1 个, C 类 1 个。按银行家算法回答下列问题:

进程	已占资源数			最大需求数		
	A	B	C	A	B	C
P1	1	2	1	8	6	4
P2	3	1	1	4	3	3
P3	4	1	3	10	1	3
P4	3	2	2	3	3	3
P5	1	1	3	5	4	6

剩余量 2 1 1
Need
P1 7 4 3
P2 1 2 2
P3 6 0 0
P4 0 1 1
P5 4 3 3
P4 → P2 → P5 → P1
→ P3
每完成一个, 释放所有已占有的

(1) 目前系统是否处于安全状态? 若是, 给出安全序列; 不是, 则说明原因。若是安全序列, 给出任何一个正确的安全序列即可。

P4 → P2 → P5 → P1 → P3

(2) 现在如果进程 P5 提出申请 (1,1,1) 个资源, 系统是否能为其分配资源? 为什么? 要求写出计算过程。
 若 P5 申请 (1,1,1) 则 P5 最大需求 (6,5,7)
 先与最大资源数相比, 若大于最大资源数则直接否认
 若小于, 则看当前序列是否满足, 满足则执行, 看是否满足序列
 若大于, 则等待直到满足为止, 再执行看是否满足序列
 可得安全序列 $P_4 \rightarrow P_2 \rightarrow P_3 \rightarrow P_1 \rightarrow P_5$
 系统可以为其分配资源。

2. (10分) 在单道批处理系统中有下列四个进程, 用先来先服务 (FCFS) 调度算法和最短作业优先调度算法 (SJF) 进行调度, 试问哪一种调度算法的性能好些? 并完成下表内容 (假设: 调度程序 10:00 开始调度)。

先来先服务 (FCFS) 调度算法

进程	提交时间	运行时间 (分钟)	开始时间	完成时间	周转时间 (分钟)
1	10:00	120	10:00	12:00	120
2	10:10	30	12:00	12:30	140
3	10:25	6	12:30	12:36	131
4	10:50	12	12:36	12:48	118

平均周转时间 $T = (120 + 140 + 131 + 118) / 4 = 127.25$ (分钟)

最短作业优先 (SJF) 调度算法

进程	提交时间	运行时间 (分钟)	开始时间	完成时间	周转时间 (分钟)
1	10:00	120	10:00	12:00	120
2	10:10	30	12:18	12:48	68
3	10:25	6	12:00	12:06	61
4	10:50	12	12:06	12:18	88

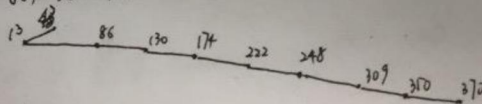
平均周转时间 $T = (120 + 158 + 101 + 88) / 4 = 116.75$ (分钟)

性能比较: 由上述所得平均周转时间, 可以看出最短作业优先 (SJF) 调度算法性能更好

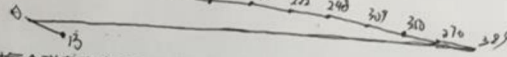
3. (8分) 假设一个磁盘驱动器有 400 个柱面, 从 0 到 399。驱动器正在为柱面号 (磁道号) 为 43 的一个请求服务, 且刚好完成的磁道请求是 25。按照 FIFO 的顺序, 即将到来的请求是 86、370、13、174、248、309、222、350、130。从现在磁头位置开始, 按照 SSTF、C-SCAN 的磁盘调度算法。要满足队列中即将到来的请求。要求:

(1) 分别给出响应请求的顺序 (要求图示)。

SSTF: 13, 86, 130, 174, 222, 248, 309, 350, 370



C-SCAN: 86, 130, 174, 222, 248, 309, 350, 370, 13



(2) 设寻道时每个磁盘移动需要 (磁头从一个磁道移动到另一个磁道) 6ms, 求采用 SSTF 和 C-SCAN 算法的寻道时间各是多少 ms? (C-SCAN 算法不要求计算从 399 号磁道返回 0 号磁道的时间)

$$\text{SSTF: } [(43-13) + (86-13) + (130-86) + (174-130) + (222-174) + (248-222) + (309-248) + (350-309) + (370-350)] \times 6 \times 10^{-3}$$

$$\text{C-SCAN: } [(86-43) + (130-86) + (174-130) + (222-174) + (248-222) + (309-248) + (350-309) + (370-350) + (13-370) + 13] \times 6 \times 10^{-3}$$

4. (8分) 一个页式存储器管理系统使用 LRU (最近最少使用) 页面替换算法, 页面大小为 1024 字节, 如果一个进程的页面访问序列为: 1、1、2、3、4、2、1、5、6、1、2、5, 分配给该进程的物理页框 (Frame) 数分别为 4, 且页框号分别为 20, 31, 2, 5. 要求:

(1) 计算访问过程中发生的缺页中断次数和缺页中断率?

(2) 如果接下来要访问逻辑地址为 3320, 试求其对应的物理地址是多少?

1	1	2	3	4	2	1	5	6	1	2	5
0	1	1	1	1	4	4	4	6	6	6	6
1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3

$$(2) 3320 \div 1024 = 3 \dots 248$$

$$\therefore \text{物理地址: } 5 \times 1024 + 248 = 5308$$

缺页中断次数: 7

$$\text{缺页中断率: } \frac{7}{12} \times 100\% = 58.3\%$$

5. (8分) 某进程的页表如下图所示 (已知 1 个 page 大小=1 个 Frame 大小=1024 字节)

页号 (Page)	页框号 (Frame)	中断位	辅存地址
0	8	1	1000
1	5	1	3000
2	7	1	5000
3		0	8000

其中, 中断位为 1 表示该页已经位于内存中. 请回答下列问题:

(1) 当进程执行指令 "Load 1, 1500" 时, 系统将怎样进行地址变换? 物理地址是多少?

1500 逻辑地址位于 1 号页, 5 号页框, 对应物理地址 $5 \times 1024 + 476 = 5196$

$$1500 / 1024 = 1 \dots 476$$

$$\therefore 1500 \text{ 位于 1 号页, 第 5 块, 物理地址 } 5 \times 1024 + 476 = 5196$$

- (2) 当执行指令 "Load 1,3600" 时, 将会发生什么现象? 给出原因。
 $7(3600/1024 = 3 \dots 528)$
 访问逻辑地址 3600 时, 3600 位于 3 号页面, 2 号中断位 0
 说明不在内存, 产生缺页中断, 从磁盘 8000 位置将页调入

6. (9分) 考虑一个索引节点所表示的 UNIX 文件的组织。假设在索引节点有 10 个直接块指针、一个一次间接块指针、一个二次间接块指针和一个三次间接块指针。假设系统块大小和磁盘块大小都是 4K, 且每个磁盘块地址用 32 位表示。请回答下列问题:

- (1) 计算此 UNIX 文件系统可以表示的最大文件的大小。要求有计算过程。 32 bit = 4 byte bit → byte

$$4K \times 10 + \frac{4K}{32} \times 4K + \frac{4K}{32} \times \frac{4K}{32} \times 4K + \frac{4K}{32} \times \frac{4K}{32} \times \frac{4K}{32} \times 4K$$

- (2) 假设索引节点已经读入内存, 如果有一个文件要访问其地址为 93160 处的数据, 需要启动磁盘几次? 请给出原因。

可以看出 $93160 \div 40K = 2316 \dots 40K < 93160 < 52K$

∴ 93160 处数据处于一次间接块指针中

故需启动磁盘 2 次