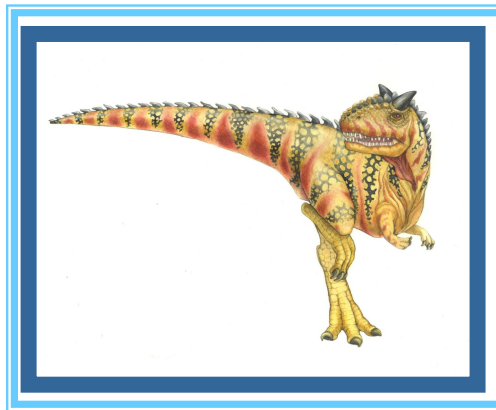


# Chapter 8: Main Memory

---





# Background

- **Program** must be brought (from disk) **into memory** and placed within a process for it to be run
  - 通常，程序以可执行文件格式保存在磁盘上
  - Main memory and registers are only storage CPU can access directly
  - Register access in one CPU clock (or less)
  - Main memory can take many cycles
  - **Cache** sits between main memory and CPU registers
- **多道程序设计模型**
  - 允许多个程序同时进入内存
- **每个进程有自己的地址空间**
  - 一个进程执行时不能访问另一个进程的地址空间
  - 进程不能执行不适合的操作





# Review

## ■ 逻辑地址，物理地址

- 逻辑地址：CPU执行程序看到的地址
- 物理地址：内存地址（内存是字节寻址）

## ■ 地址绑定

- 编译时，加载时，执行时
- 动态地址重定位（Relocation）
  - ▶ MMU 内存管理单元（CPU内）

## ■ 连续内存分配 Contiguous Allocation

- 固定分区（内碎片，Internal Fragmentation）
- 可变分区（外碎片，External Fragmentation）
  - ▶ 动态存储分配：首次适应，下次适应，最佳适应，最差适应
  - ▶ 紧缩 Compaction





# Review

## ■ 分页 Paging

- 页式地址变换
  - ▶ 逻辑地址：页号，页内偏移
  - ▶ 页表：逻辑页号与帧号的对应关系      page, frame
- 内碎片
- 页表结构
  - ▶ 分层页表 Hierarchical Page Table
  - ▶ 哈希页表 Hashed Page Table
  - ▶ 倒置页表 Inverted Page Table
- TLB: Effective Access Time, TLB hit ratio

## ■ 分段 Segmentation

- 段式地址变换
  - ▶ 逻辑地址：段号，段内偏移
  - ▶ 段表：base, limit





# 页式地址变换

- 虚拟地址（逻辑地址、程序地址）以十六进制、八进制、二进制形式给出
  - 将虚地址转换为二进制数
  - 分离出页号和位移量
    - ▶ 低位部分：位移量
    - ▶ 高位部分：页号
  - 将位移量直接复制到内存地址寄存器的低位部分
  - 以页号查页表，得到对应页装入内存的块号，并将块号转换为二进制数填入地址寄存器的高位部分，从而形成内存地址





# 页式地址变换

## ■ 虚拟地址以十进制给出

- 页号 =  $\text{int}(\text{虚地址} / \text{页大小})$  取整
- 位移量 =  $\text{虚地址} \bmod \text{页大小}$  取余

## ■ 以页号查页表，得到对应页装入内存的块号

## ■ 内存地址 = 块号 \* 页大小 + 位移量





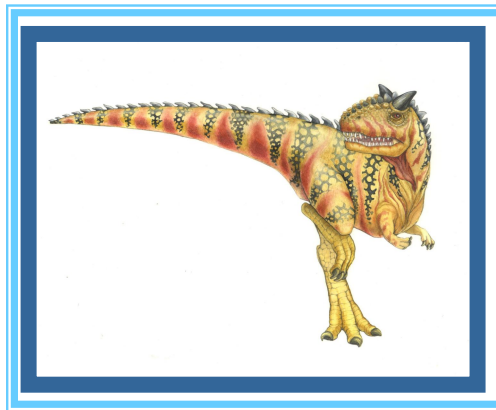
# 基本内存管理方案总结

固定分区	把可分配的内存空间分割成若干个连续区域，每一区域称为分区。每个分区的大小可以相同也可不同， <b>分区大小固定不变，每个分区装一个且只能装一个进程</b>
可变分区	根据进程的需求，把可分配的内存空间分割出一个分区，分配给该进程
页式	把用户程序地址空间分为大小相等的部分，称为 <b>页</b> 。内存空间按页大小划分为大小相同的区域，称为 <b>帧</b> （内存块，物理页面，页框）。 <b>以页为单位进行分配</b> ，逻辑上相邻的页，物理上不一定相邻
段式	用户程序地址空间按进程自身的逻辑关系划分为若干段，内存空间被动态得划分为长度不相同的区域（ <b>可变分区</b> ）。 <b>以段为单位分配内存</b> ，每个段在内存中占据连续空间，各段之间可以不连续存放
段页式	用户程序地址空间：段式 内存空间：页式 分配单位：页



# Chapter 9: Virtual Memory

---







# Background

- Physical memory space is always not large enough for us
  - Our programs are getting larger and larger
  - We want to run as many as possible programs at the same time

- 程序特点

- 程序通常具有处理异常错误条件的代码。
- 数组、链表等所分配的内存量通常多于实际需求量。
- 程序的某些选项和功能可能很少使用。

➡ 不需要将整个程序置于内存中

- Locality of reference 程序的局部性原理

- Within a period of time, the program execution is only **limited to a portion**. Accordingly, it accesses the storage space is also limited to **a certain area**.

➡ 不同时需要整个程序





# Review

## ■ 虚拟页式存储管理基本思想

- 进程开始之前，不是装入全部页面，而是装入一个或者零个页面
- 之后，根据进程运行的需要，动态装入其他页面（Demand paging）
- 当内存空间已满，而又需要装入新的页面时，则根据某种算法置换内存中的某个页面，以便装入新的页面（Page Replacement）

## ■ Demand Paging （请求调页）

- Demand paging = paging + page fault + page replacement
- 硬件支持

- ▶ 页表结构

页号	物理块号	状态位P	访问字段A	修改位M	外存地址
----	------	------	-------	------	------

- ▶ 缺页中断
- ▶ 地址转换





# Review

## ■ Page Fault

### ● 产生原因

- ▶ 所访问的虚拟页面没有调入物理内存（缺页异常）
- ▶ 页面访问违反权限（读/写，用户/内核）
- ▶ 错误的访问地址

### ● Performance of Demand Paging

- ▶ Effective Access Time (EAT) 有效访问时间

$$\text{EAT} = (1 - p) \times \text{memory access}$$

+  $p$  (page fault overhead

+ swap page out

+ swap page in

+ restart overhead)





# Review

- Page Replacement （页面置换）
  - FIFO
    - ▶ Belady现象
  - Optimal Page Replacement （最优置换）
  - Least Recently Used (LRU)
    - ▶ Counter
    - ▶ Stack
  - Reference-Bit
  - Second-Chance (Clock Algorithm)
  - Counting Algorithm
- Copy-on-Write
- Memory-Mapped Files





# Review

## ■ Allocation of Frames

- 固定分配&局部置换
- 可变分配&局部置换
- 可变分配&全局置换

## ■ **Thrashing** $\equiv$ a process is busy swapping pages in and out

- 页面在内存和磁盘间频繁调度，使调度页面所需时间比进程实际运行时间还多，这种现象称为抖动或颠簸

## ■ Working-Set

## ■ Allocating Kernel Memory

- Buddy System
- Slab Allocator

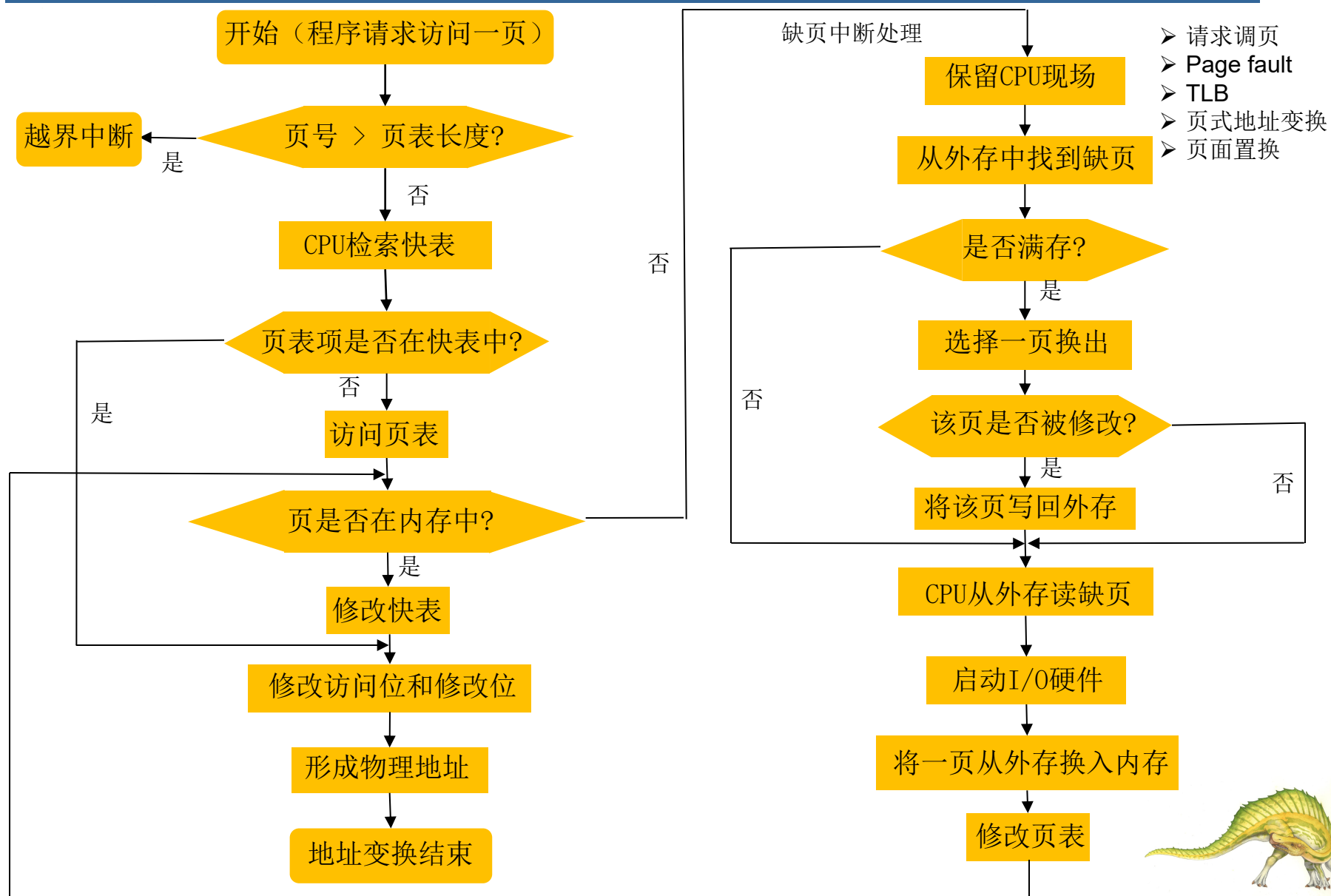
## ■ Other Issues

- Prepaging, Page Size, TLB Reach, Program Structure





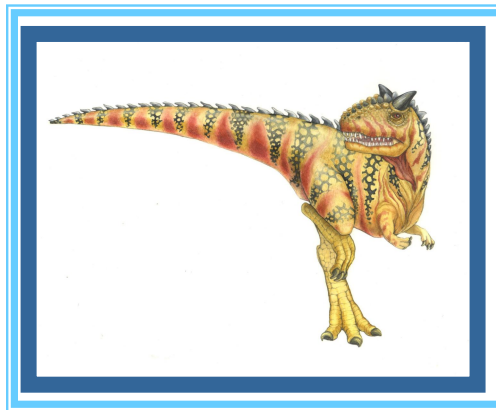
# 虚拟存储过程



# Chapter 10: File-System Interface

## Chapter 11: File System Implementation

---





# Review

- **文件：**一组带标识（标识即文件名）的、在逻辑上有完整意义的信息项的序列
- **文件基本操作**
  - Create、Read、Write、Seek、Delete、Truncate
- **文件的逻辑结构/文件访问方法**
  - 顺序访问、直接访问、索引访问
- **文件的物理结构/分配方法**
  - 连续分配
  - 链接分配
    - ▶ 文件分配表FAT
  - 索引分配







	优点	缺点
Contiguous Allocation	<ul style="list-style-type: none"><li>- 简单</li><li>- 支持顺序存储和随机存取</li><li>- 所需的磁盘寻道次数和寻道时间最少</li><li>- 可以同时读入多个块，检索一个块也很容易</li></ul>	<ul style="list-style-type: none"><li>- 文件不能动态增长</li><li>- 不利于文件插入和删除</li><li>- 外部碎片：紧缩技术</li></ul>
Linked Allocation	<ul style="list-style-type: none"><li>- 提高了磁盘空间利用率，不存在外碎片问题</li><li>- 有利于文件插入和删除</li><li>- 有利于文件动态扩充</li></ul>	<ul style="list-style-type: none"><li>- 存取速度慢，不适于随机存取</li><li>- 可靠性问题，如指针出错</li><li>- 更多的寻道次数和寻道时间</li><li>- 链接指针占用一定的空间</li></ul>
Indexed Allocation	<p>保持链接结构的优点，又克服了其缺点：</p> <ul style="list-style-type: none"><li>- 既能顺序存取，又能随机存取</li><li>- 满足了文件动态增长、插入删除要求</li><li>- 能充分利用外存空间</li></ul>	<ul style="list-style-type: none"><li>- 较多的寻道次数和寻道时间</li><li>- 索引表本身带来了系统开销, 如：内外存空间，存取时间</li></ul>



# Review

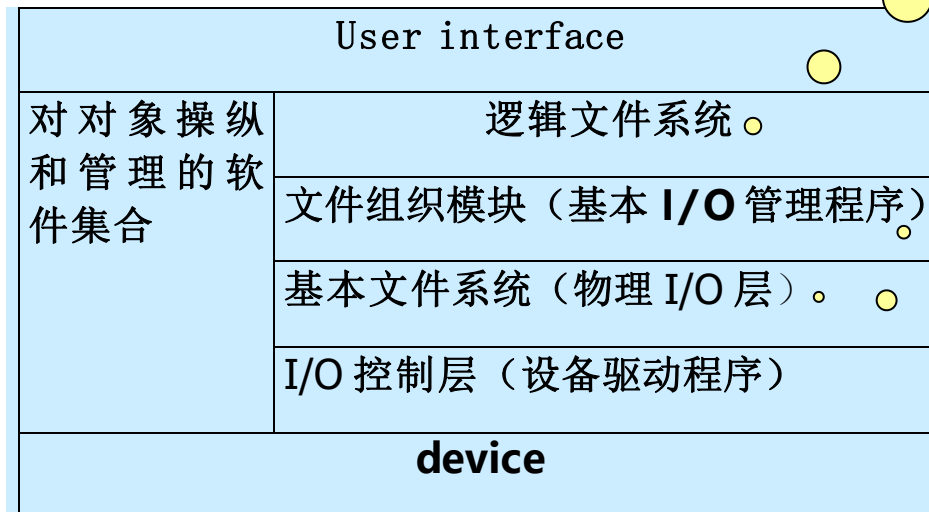
## ■ 目录结构

- 一级目录、二级目录、树形目录、无环图目录、通用图目录
  - ▶ Naming problem、Grouping problem、Searching problem

## ■ 文件共享与保护

- 用户分组
- ACL

## ■ 文件系统分层实现



管理文件目录,根据文件名得到该文件的相关信息,提供给下一层;文件的保护和安全

找到I/O设备;实现逻辑记录到数据块的映射;磁盘调度,性能优化

向驱动程序发出读/写数据块的命令

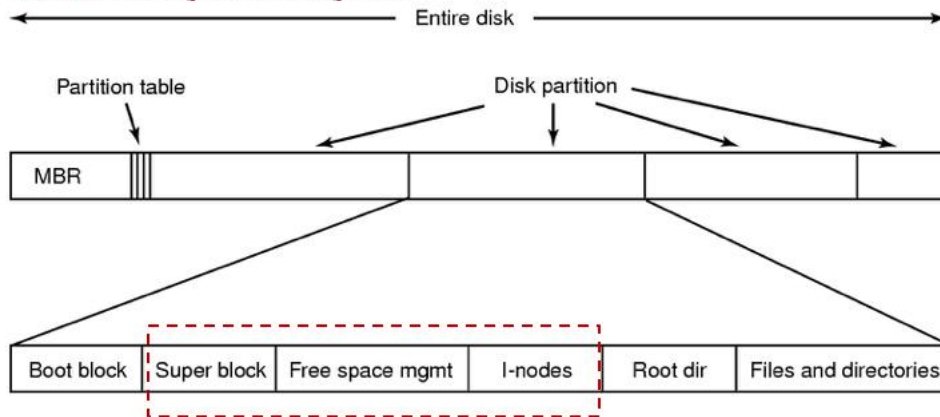




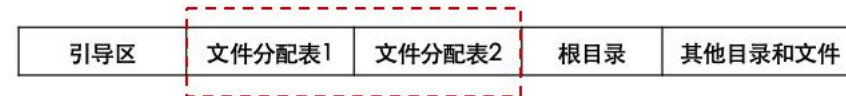
# Review

- 磁盘、内存 结构支持
  - File System Layout

## Unix file system layout



## Window FAT file system layout



- In-Memory File System Structures
  - ▶ The **system-wide open-file table**
  - ▶ The **per-process open-file table**





# Review

## ■ 磁盘、内存 结构支持

### ● In-Memory File System Structures

#### ■ System-wide open-file table

- 整个系统一张
- 放在内存：用于保存已打开文件的FCB

FCB( <u>iNode</u> )信息	引用次数	修改标记
-----------------------	------	------

#### ■ Per-process open-file table

- 每个进程一张
- 进程的FCB中记录了用户打开文件表的位置

文件描述符	打开方式	读写指针	系统打开文件表索引
-------	------	------	-----------

### ● $Open(F_i)$

- ▶ 根据文件名在文件目录中检索，并将该文件的目录项读入内存，建立相应的数据结构（系统打开文件表、用户打开文件表），为后续的文件操作做好准备。返回文件描述符/文件句柄

### ● $Close(F_i)$ :moves the content of entry $F_i$ in memory to directory structure on disk





# Review

---

## ■ 空闲空间管理

- 位图、空闲块链表、成组链接法

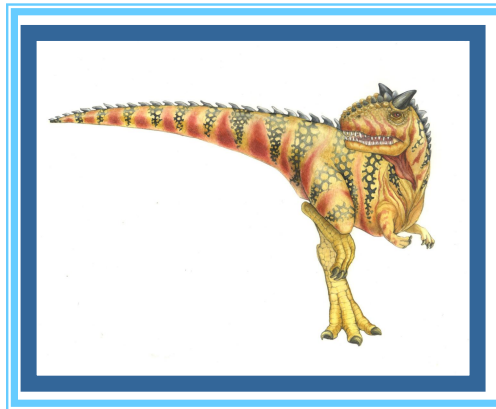
## ■ 效率与性能

- 目录项（FCB）分解、当前目录、合理分配磁盘空间
- 块高速缓存、提前读取
- 磁盘调度、RAID技术等



# Chapter 12: Mass-Storage Systems

---

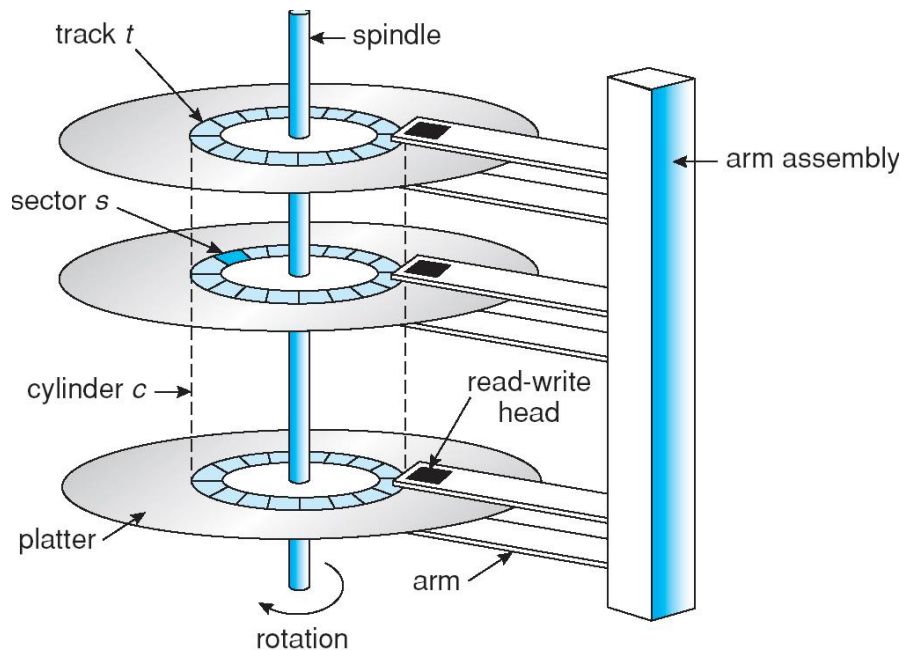




# Review

## ■ 磁盘结构

- 物理地址形式：磁头号（盘面号）、磁道号（柱面号）、扇区号
- Positioning time
  - ▶ seek time 寻道时间
  - ▶ rotational latency 旋转延迟



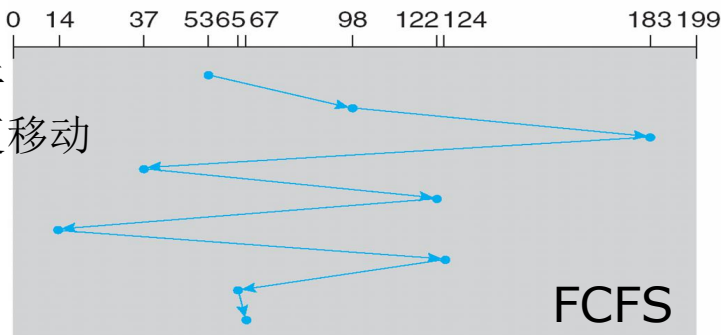


# Review

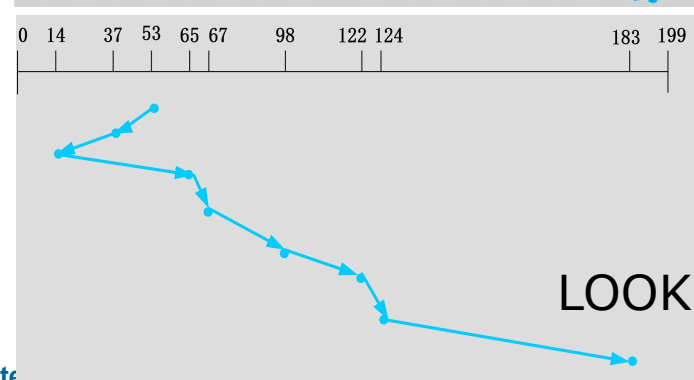
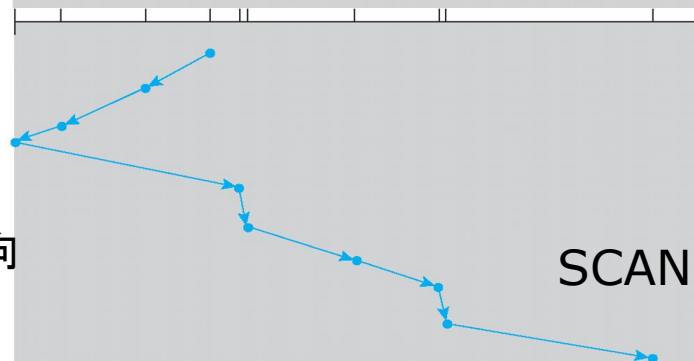
## ■ 磁盘调度

queue = 98, 183, 37, 122, 14, 124, 65, 67  
head starts at 53

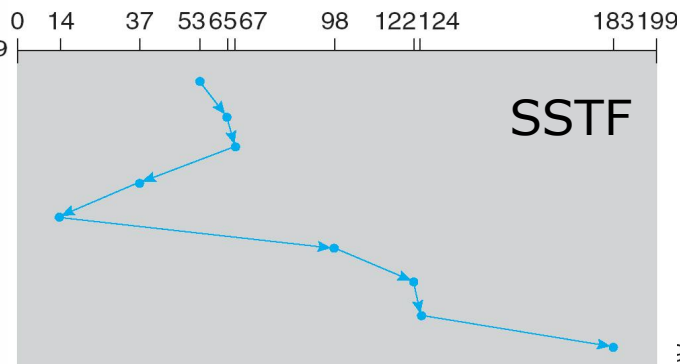
- 简单公平  
- 磁头反复移动



折中权衡  
距离、方向

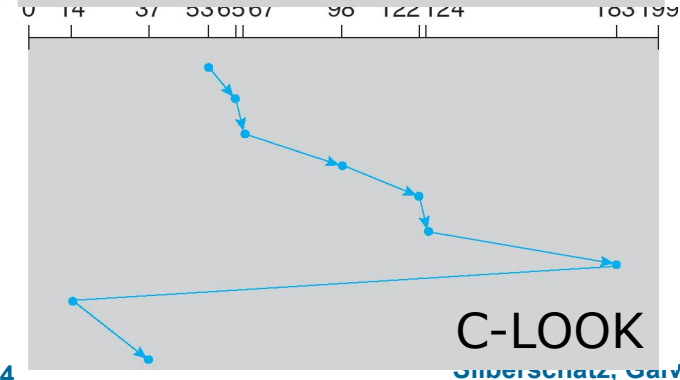
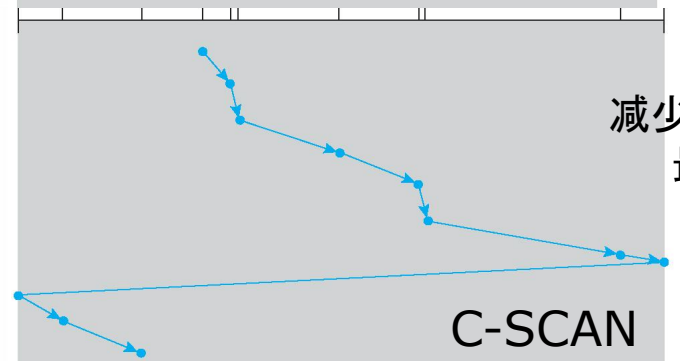


queue = 98, 183, 37, 122, 14, 124, 65, 67  
head starts at 53



- 改善了磁盘平均服务时间  
- 饥饿

减少了新请求的最大延迟







# Review

---

## ■ 磁盘管理

- 磁盘格式化
- 启动
- 坏块管理

## ■ 交换空间管理

- 大小、位置、何时分配

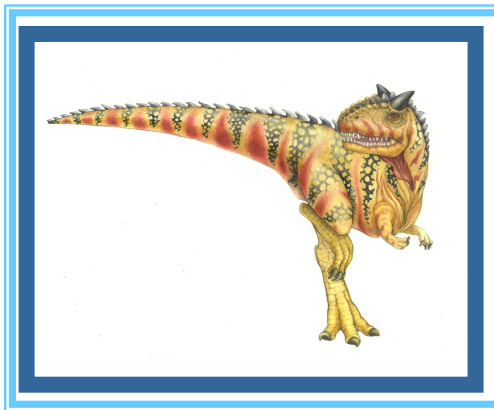
## ■ RAID

- 通过冗余提高可靠性；通过并行提高性能



# Chapter 13: I/O Systems

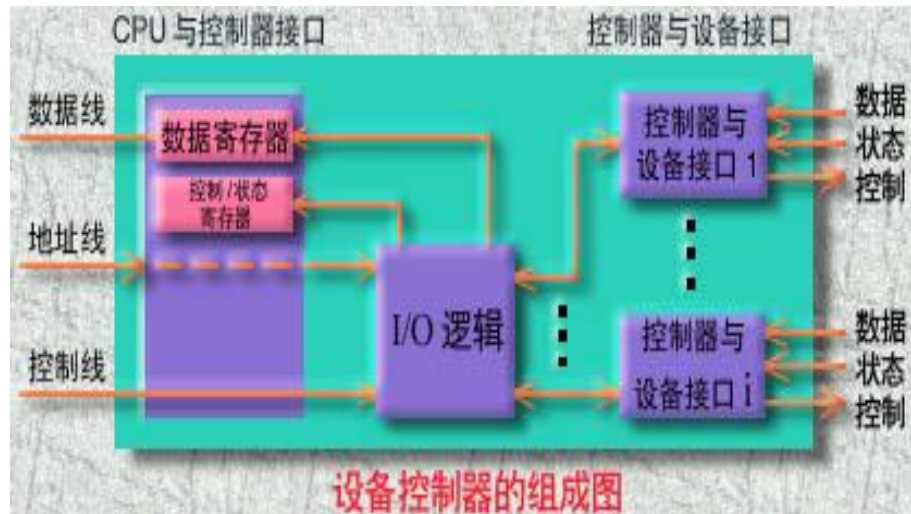
---





# Review

- **I/O设备**由机械和电子两部分组成：
  - (1) 机械部分：**设备本身**
  - (2) 电子部分：**设备控制器**（或适配器）
    - （端口）地址译码
    - 接收主机发来的数据和控制信号/向主机发送数据和状态信号
    - 将计算机的数字信号转换为机械能识别的模拟信号，或反之
    - 实现设备内部硬件缓冲、错误检查等
- 设备不直接与CPU进行通信，而是与设备控制器通信
- 每个I/O设备通过设备控制器与计算机的数据总线和地址总线相连接





# Review

## ■ Software I/O Approaches

CPU与I/O分离 性能

I/O控制方式：主机与I/O设备之间的数据交换方式

(1) CPU直接控制外围设备



Programmed I/O

(2) 增加了控制器或I/O部件，CPU使用非中断的**可编程I/O**



- CPU开始从外设设备接口的具体细节中分离出来

(3) 与(2)相同，但采用**中断**方式



- CPU无需花费等待执行I/O操作所需的时间，效率提高

(4) I/O部件通过**DMA**直接控制存储器



- 可以在没有CPU参与的情况下，从该内存中移出或往内存中移入一块数据，仅仅在传送开始和结束时需要CPU干预

(5) **I/O**部件增强为一个单独的**处理器**，有专门为I/O设计的指令集

- CPU指导I/O处理器执行内存中的I/O程序，I/O处理器没有CPU干涉的情况下取指令并执行这些指令





执行输入输出系统调用，  
**I/O数据格式化**，为  
**Spooling**输入输出做准备

**独立于设备的  
软件实现**

：设备命名  
、设备保护  
、缓冲技术  
、设备分配  
与释放

I/O  
请求

用户进程

独立于设备的软件

设备驱动程序

中断处理程序

硬件

I/O完  
成后的  
回答

设置设备寄  
存器、检查  
设备的执行  
状态

I/O软件的层次

**I/O完成时**，唤醒  
设备驱动程序进程  
，进行中断处理

物理I/O操作





## ■ Device Drivers: The software that talks to the controller

- 接收来自与设备无关的上层软件的抽象请求，并执行这个请求
- 每个设备控制器都设有一个或多个设备寄存器，用来存放向设备发送的命令和参数。设备驱动程序负责释放这些命令，并监督它们正确执行。即，**设备驱动程序是处理或操作设备控制器的软件。**
- 它们是内核中具有高特权的、**驻留内存的底层硬件处理例程。**

## ■ Device-Independent I/O Software

- Uniform interfacing for device drivers
- Buffering
- Error reporting
- Allocating and releasing dedicate device
  - ▶ LUT, SDT, DCT, COCT, CHCT

## ■ User-Space I/O Software

- Format, Spooling





## ■ User-Space I/O Software

- Format

- Spooling - hold output for a device

- ▶ It is one way OS can **coordinate concurrent output**.
- ▶ If device can serve only one request at a time.
  - i.e., Printing
    - » OS intercepts all output to a printer and spools them to **a separate disk file**.
    - » When an application finishes printing, the spooling system queues the spool files for output to the printer one by one.





- 名词解释 15分
- 填空题 15分
- 判断/选择题 10分
- 简答题 20分      4
- 计算题  $6*5+10=40$ 分





# Review

■ 1. 下列说法正确的有 ( B )

## 先进先出 (FIFO) 页面置换算法会产生Belady现象

## II 最近最少使用 (LRU) 页面置换算法会产生Belady现象

III 在进程运行时，若它的工作集页面都在虚拟存储器内，则能够使该进程有效地运行，否则会出现频繁的页面调入/调出现象

IV 在进程运行时，若它的工作集页面都在主存储器内，则能够使该进程有效地运行，否则会出现频繁的页面调入/调出现象

●A. 仅 I、III

B. 仅 I、IV

●C. 仅 II、III

D. 仅 II、IV





# Review

- 2. 在一个请求分页系统中，采用LRU页面置换算法时，加入一个作业的页面走向为：1, 3, 2, 1, 1, 3, 5, 1, 3, 2, 1, 5。当分配给该作业的物理块数分别为3和4时，在访问过程中所发生的缺页率为（ C ）
- A. 25%, 33%
  - B. 25%, 100%
  - C. 50%, 33%
  - D. 50%, 75%





# Review

- 3. 设有8页的逻辑空间，每页有1024B，他们被映射到32块的物理存储区中。那么，逻辑地址的有效位是（ C ）位，物理地址至少是（ C ）位
  - A. 10, 11
  - B. 12, 14
  - C. 13, 15
  - D. 14, 16

## ■ 第3题分析：

- 页大小、页号位数、物理块数、页内偏移地址、逻辑地址位数、物理地址位数之间的联系。

因为8页= $2^3$ 页，所以表示页号的地址有3位，又因为每页有1024B= $2^{10}$ B，所以页内偏移地址有10位，所以逻辑地址总共有13位；又因为页面的大小和物理块的大小是一样，所以每个物理块也是1024B，而内存至少有32块物理块，所以内存大小至少是 $32 \times 1024B = 2^{15}B$ ，所以物理地址至少要15位，不然无法访问内存的所有区域。故选C。





# Review

■ 4. 在某页式存储管理系统中，页表内容见图。若页面的大小为4KB，则地址转换机构将逻辑地址0转换成的物理地址是（ A ）

- A. 8192
- B. 8193
- C. 2048
- D. 2049

■ 第4题分析：

■ 本题中页的大小为4KB，从表中可知，每个页存储在一个块中，即每个块大小为4KB。逻辑地址0对应的页号为0，表中对应的块号为2。第0块物理地址范围为0~4095；第1块物理地址范围为4096~8191；第2块物理地址范围为8192~12287。本题容易错在逻辑地址、物理地址、块号都是从0开始编址的，而不是1。

表 3-10 页表内容

页 号	块 号
0	2
1	1
3	3
4	7





# Review

- 5. 不会产生内部碎片的存储管理是（ **B** ）
  - A. 分页式存储管理
  - B. 分段式存储管理
  - C. 固定分区式存储管理
  - D. 段页式存储管理
- 6. 要保证一个程序在主存中被改变了存放位置后仍能正确地执行，则对主存空间应采用（ **B** ）技术
  - A. 静态重定位
  - B. 动态重定位
  - C. 动态分配
  - D. 静态分配





# Review

- 7. 作业在执行中发生缺页中断，经操作系统处理后应让其执行（ **B** ）指令
  - A. 被中断的前一条
  - B. 被中断的那一条
  - C. 被中断的后一条
  - D. 启动时的第一条
- 8. （ **B** ）存储管理方式提供一维地址结构
  - A. 分段
  - B. 分页
  - C. 分段和段页式
  - D. 以上都不对





# Review

- 9. 在虚拟页式存储管理方案中，（ A ）完成将页面调入内存的工作
  - A. 缺页中断处理
  - B. 页面淘汰过程
  - C. 工作集模型应用
  - D. 紧缩技术利用
- 10. 在段页式分配中，CPU每次从内存中取一次数据需要（ C ）次访问内存
  - A. 1
  - B. 2
  - C. 3
  - D. 4
- 第10题分析：
- 在段页式分配中，取一次数据时先从内存查找段表，再访问内存查找相应的页表，最后拼成物理地址后访问内存，共需要3次内存访问。





- 11. 一台计算机有4个页框。装入时间、上次引用时间、他们的R（读）与M（修改）位见表（时间单位：一个时钟周期），请问FIFO、LRU和第二次机会算法将分别替换哪一页？

页使用情况表

页	装入时间	上次引用时间	R	M
0	126	279	0	0
1	230	260	1	0
2	120	272	1	1
3	160	280	1	1







- 11. 一台计算机有4个页框。装入时间、上次引用时间、他们的R（读）与M（修改）位见表（时间单位：一个时钟周期），请问FIFO、LRU和第二次机会算法将分别替换哪一页？

页	装入时间	上次引用时间	R	M
0	126	279	0	0
1	230	260	1	0
2	120	272	1	1
3	160	280	1	1

- 第11题分析：

- 1) FIFO算法淘汰最先进入内存的页。由题表所示可知，第2页最先进入内存（装入时间最小），故FIFO算法将淘汰第2页。
- 2) LRU算法淘汰最近最久未用的页。根据题表所示，最近最久未使用的页（上次引用时间最小）是第1页，故LRU算法将淘汰第1页。
- 3) 第二次机会算法时淘汰一个自上一次对它检查以来没有被访问过的页。根据题表所示可知，自上一次对它检查以来只有第0页未被访问过（R和M均为0），故第二次机会算法将淘汰第0页。





- 12. 已知系统为32位实地址，采用48位虚拟地址，页面大小4KB，页表项大小为8B；每段最大为4GB。

1) 假设系统使用纯页式存储，则要采用多少级页表，页内偏移多少位？

### ■ 第12题分析：

1) 已知页面大小 $4KB=2^{12}B$ ，即页内偏移量的位数为12。采用48位虚拟地址，故虚页号为 $48-12=36$ 位。

页表项的大小为8B，则每页可容纳 $4KB/8B=512=2^9$ 项。

那么所需多级页表的级数 $=\lceil 36/9 \rceil = 4$ ，故应采用4级页表。





- 12. 已知系统为32位实地址，采用48位虚拟地址，页面大小4KB，页表项大小为8B；每段最大为4GB。

2) 假设系统采用一级页表，TLB命中率为98%，TLB访问时间为10ns，内存访问时间为100ns，并假设当TLB访问失败后才开始访问内存，问平均页面访问时间是多少？

### ■ 第12题分析：

2) 系统进行页面访问操作时，首先读取页面对应的页表项，有98%的概率可以在TLB中直接读取到（10ns），然后进行地址变换，访问内存读取页面（100ns），所需时间为10ns+100ns+100ns=210ns。页表平均访问时间为：

$$[98\% * 110 + (1 - 98\%) * 210]ns = 112ns$$





- 12. 已知系统为32位实地址，采用48位虚拟地址，页面大小4KB，页表项大小为8B；每段最大为4GB。

3) 如果是二级页表，页面平均访问时间是多少？

### ■ 第12题分析：

3) 二级页表的情况下，TLB命中的访问时间还是110ns，未命中的访问时间加上一次内存访问时间，即 $210\text{ns} + 100\text{ns} = 310\text{ns}$ ，那么平均访问时间为：

$$[98\% * 110 + (1 - 98\%) * 310]\text{ns} = 114\text{ns}$$





- 12. 已知系统为32位实地址，采用48位虚拟地址，页面大小4KB，页表项大小为8B；每段最大为4GB。

4) 上题中，如果要满足访问时间 $\leq 120\text{ns}$ ，那么命中率需要至少多少？

- 第12题分析：

4) 本问是在小题3基础上提出的，假设快表命中率为 $p$ ，则应满足：

$$[p * 110 + (1 - p) * 310]\text{ns} \leq 120\text{ns}$$

求解不等式，得 $p \leq 95\%$





- 12. 已知系统为32位实地址，采用48位虚拟地址，页面大小4KB，页表项大小为8B；每段最大为4GB。

5) 若系统采用段页式存储，则每用户最多可以有多少个段？段内采用几级页表？

### ■ 第12题分析：

5) 系统采用48位虚拟地址，虚拟地址空间为 $2^{48}\text{B}$ ，每段最大为4GB，那么最大段数 $=2^{48}\text{B}/4\text{GB}=2^{16}=65536$ 。

$4\text{GB}=2^{32}\text{B}$ ，即段内地址位数为32；

页表项的大小为8B，则每页可容纳 $4\text{KB}/8\text{B}=512=2^9$ 项；页面大小 $4\text{KB}=12$ 位

段内采用多级页表，那么多级页表级数 $=[(32-12)/9]=3$ ，故段内采用3级页表。





- 13. 某虚拟存储器的用户空间共有32个页面，每页1KB，主存16KB。假定某时刻系统为用户的第0, 1, 2, 3页分配的物理块号为5, 10, 4, 7，而该用户作业的长度为6页，试将十六进制的虚拟地址0A5C, 103C, 1A5C转换成物理地址。

- 解答：

可知系统逻辑地址有15位，其中高5位为页号，低10位为页内偏移。物理地址有14位，高4位为块号，低10位为块内偏移。

1) 0A5C=000 1010 0101 1100，页号00010为2，页号合法。从页表得知对应块号为4，即0100；与页内偏移拼接形成物理地址010010 0101 1100，即125C。

2) 103C，页号为4，页号合法，但该页未装入内存，产生缺页中断

3) 1A5C，页号为6，为非法页号，故产生越界中断。





# Review

- 1. 操作系统为了管理文件，设计了文件控制块（FCB），FCB的建立是（ **A** ）。
- A. 调用create()时
  - B. 调用open()时
  - C. 调用read()时
  - D. 调用write()时







# Review

- 2. 文件的顺序存取是（ **B** ）。
  - A. 按终端号依次存取
  - B. 按文件的逻辑号逐一存取
  - C. 按物理块号依次存取
  - D. 按文件逻辑记录大小逐一存取

- 第2题解析：

顺序存取文件是按其在文件中的逻辑顺序依次存取的，只能从头往下读。在4个选项中，只有逻辑号跟逻辑顺序的意思最接近，故选B。





# Review

■ 3. 无结构文件的含义是（ C ）。

- A. 变长记录的文件
- B. 索引文件
- C. 流式文件
- D. 索引顺序文件

■ 第3题解析：

无结构文件是指由字符流构成的文件，故又称为流式文件。





# Review

- 4. 文件系统中设立打开(open)系统调用的主要目的是（ **B** ）。
- A. 把文件从辅存读到内存
  - B. 把文件的控制信息从辅存读到内存
  - C. 把文件的FAT表信息从辅存读到内存
  - D. 把磁盘文件系统的控制管理信息从辅存读到内存

■ 第7题解析：

打开文件是指系统将指定文件的属性（包括该文件在外存上的物理位置）从外存复制到内存打开文件表的一个表目中，并将该表目的编号（或称为索引）返回给用户。





# Review

- 5. 下面关于文件的叙述中，错误的是（ **ABCD** ）
- A. 打开文件的主要操作是把指定文件复制到内存指定的区域
  - B. 对一个文件访问，常由用户访问权限和用户优先级共同限制
  - C. 文件系统采用树形目录结构后，对于不同用户的文件，其文件名应该不同
  - D. 为防止系统故障造成系统内文件受损，常采用存取控制矩阵方法保护文件

■ 第5题解析：

- A. 将目录放到打开文件表中。
- B. 由用户访问权限和文件属性共同限制。
- C. 可相同，也可不同。
- D. 常用备份方法保护文件，而存取控制矩阵的方法是用于多用户之间的存取权限保护





# Review

- 6. 如果文件采用直接存取方法，且文件大小不固定，则应采用（ **B** ）物理结构。
  - A. 直接
  - B. 索引
  - C. 随机
  - D. 顺序
  
- 7. 用户在删除某文件的过程中，操作系统不可能执行的操作是（ **A** ）。
  - A. 删除此文件所在的目录
  - B. 删除与此文件关联的目录项
  - C. 删除与此文件对应的文件控制块
  - D. 释放此文件关联的内存缓冲区





# Review

- 8. 下列文件物理结构中，适合随机访问且易于文件扩展的是（ B ）。
- A. 连续结构
  - B. 索引结构
  - C. 链式结构且磁盘块定长
  - D. 链式结构且磁盘块变长

■ 第8题解析：

根据外存储分配方法，**链式存储结构**将文件按照顺序存储在不同盘块中，因此**适合顺序访问，不适合随机访问**（需从文件头遍历所有盘块）；连续结构（数据位置可计算得到）和索引结构（只需访问索引块即可知道数据位置）适合随机访问。但连续结构如果要在中间增加数据，则要整体移动后面的所有数据，因此不适合文件的动态增长；而索引结构适合随机访问，因为索引结构可以单独将新增数据放在一个新盘块，只需修改索引块即可。





# Review

- 9. 在磁盘上容易导致外碎片发生的物理文件结构是（ **B** ）。
- A. 链接
  - B. 连续
  - C. 索引
  - D. 索引和链接





■ 10. 下面关于目录检索的说法中，正确的是（ C ）。

- A. 由于散列法具有较快的检索速度，因此现代操作系统中都用它来替代传统的顺序检索方法
- B. 在利用顺序检索法时，对属性目录应采用文件的路径名，应从根目录开始逐级检索
- C. 在利用顺序检索法时，只要路径名的一个分量名未找到，便应停止查找
- D. 在顺序检索法的查找完成后，即可得到文件的物理地址







- 11. 考虑一个文件存放在100个数据块中。文件控制块、索引块或索引信息都驻留内存，那么如果（ **B** ），不需要做任何磁盘I/O操作。
- A. 采用连续分配策略，将最后一个数据块搬到文件头部
  - B. 采用单级索引分配策略，将最后一个数据块插入文件头部
  - C. 采用隐式链接分配策略，将最后一个数据块插入文件头部
  - D. 采用隐式链接分配策略，将第一个数据块插入文件尾部





# Review

- 12. 某文件系统物理结构采用采用三级索引分配方法，如果每个磁盘块大小为1024B，每个盘块索引号占用4B，请问在该文件系统中，最大文件的大小最接近的是（ **B** ）。
- A. 8GB
  - B. 16GB
  - C. 32GB
  - D. 2TB

■ 第12题解析：

每个盘块为1024B，索引号为4B，因此，每个索引块可以存放256个索引号，三级索引块可以管理文件的大小为： $256 \times 256 \times 256 \times 1024\text{B}$ 约等于16GB。





# Review

- 13. 存放在某个磁盘上的文件系统采用混合索引分配方式，其FCB中共有13个地址项，第0~9个地址项为直接地址，第10个地址项为一次间接地址，第11个地址项为二次间接地址，第12个地址项为三次间接地址。假设每个盘块的大小为512B，若盘块号需要占3B，而每个盘块最多存放170个盘块地址，则：

(1) 该文件系统允许文件的最大长度是多少？

- 第13题解析：

(1) 该文件系统的一个文件的最大长度可达：

$$\begin{aligned} 10 + 170 + 170 * 170 + 170 * 170 * 170 &= 4942080 \text{ 块} = 4942080 * 512 \text{ B} \\ &= 2471040 \text{ KB} \end{aligned}$$





# Review

- 13. 存放在某个磁盘上的文件系统采用混合索引分配方式，其FCB中共有13个地址项，第0~9个地址项为直接地址，第10个地址项为一次间接地址，第11个地址项为二次间接地址，第12个地址项为三次间接地址。假设每个盘块的大小为512B，若盘块号需要占3B，而每个盘块最多存放170个盘块地址，则：

(2) 将文件的第5000B，15000B，150000B转换成物理块号和块内偏移。

- 第13题解析：

(2)  $5000/512$ ，商为9，余数为392。即对应的逻辑块号为9，块内位移为392。由于 $9 < 10$ ，因此可直接从该文件的FCB的第9个地址项处得到物理盘块号，块内偏移为392。

$15000/512$ ，商为29，余数为152。即对应的逻辑块号为29，块内位移为152。由于 $10 < 29 < 10 + 170$ ，而 $29 - 10 = 19$ ，因此可从FCB里第10个地址项得到一次间接地址块的地址，并从一次间接地址块的第19项中获得对应的物理盘块号，块内位移为152。

$150000/512$ ，商为192，余数为496。即对应的逻辑块号为292，块内位移为496。由于 $10 + 170 < 192 < 10 + 170 + 170 * 170$ ，而 $192 - (10 + 170) = 112$ ， $112/170$ 得到商为0，余数为112，因此可从FCB第11个地址项得到二次间接地址块的地址，并从二次间接地址块的第0项中获得一个一次间接地址块的地址，再从该一次间接地址块的第112项中获得对应的盘块号，块内位移为496。





# Review

- 13. 存放在某个磁盘上的文件系统采用混合索引分配方式，其FCB中共有13个地址项，第0~9个地址项为直接地址，第10个地址项为一次间接地址，第11个地址项为二次间接地址，第12个地址项为三次间接地址。假设每个盘块的大小为512B，若盘块号需要占3B，而每个盘块最多存放170个盘块地址，则：

(3) 假设某个文件的FCB已在内存，但其他信息均在外存，为了访问该文件中某个位置的内容，最少需要几次访问磁盘？最多需要几次访问磁盘？

- 第13题解析：

最少需要1次。即可通过在内存的FCB得到直接地址，通过直接地址进行访盘。

最多需要4次。分别为访问三次间接地址、二次间接地址、一次间接地址及读文件盘块。





# Review

- 1. 磁盘调度算法中，（ **B** ）算法可能会随时改变移动臂的运动方向。
- A. 电梯调度
  - B. 最短寻道时间优先
  - C. 扫描
  - D. 单向扫描





# Review

---

- 2. 下列算法中，用于磁盘调度的是（ C ）。
- A. 时间片轮转法
  - B. LRU算法
  - C. 最短寻道时间优先算法
  - D. 高优先级算法





# Review

- 3. 一个磁盘的转速是7200 r/min, 每个磁道有160个扇区, 每个扇区为512B, 那么理想情况下, 其数据传输率为 ( C )。
- A. 7200X160 KB/s
  - B. 7200 KB/s
  - C. 9600 KB/s
  - D. 19200 KB/s







# Review

- 4. 如果当前读写磁头正在53号柱面上执行操作，依次有4个等待访问的请求，柱面号依次为98、37、124、65，当采用（ C ）算法时，下一次磁头才可能到达37号柱面。
- A. 先来先服务
  - B. 最短寻道时间优先
  - C. 电梯调度（初始磁头移动方向向着小磁道方向）
  - D. 循环扫描（磁头移动方向向着大磁道方向）





- 5. 若8个字（字长32位）组成的位示图管理内存，假定用户归还一个块号为100的内存块，它对应的位示图的位置为（ **B** ）。假定字号、位号、块号均从1开始算起，而不是从0开始。
- A. 字号为3，位号为5
  - B. 字号为4，位号为4
  - C. 字号为3，位号为4
  - D. 字号为4，位号为5





- 6. 下列选项中，不能改善磁盘设备I/O性能的是（**B**）。
- A. 重排I/O请求次序
  - B. 在一个磁盘上设置多个分区
  - C. 预读和滞后写
  - D. 优化文件物理的分布





# Review

■ 7. 假设磁盘的每个磁道分为9个块，现一文件有A, B, ...I共9个记录，每个记录的大小与块的大小相等，设磁盘转速为27ms/转，每读出一块后需要2ms的处理时间，若忽略其他辅助时间，试问：

1) 如果这些记录被顺序存放于同一磁道上，文件处理程序顺序读取且顺序处理记录，处理文件要多长时间？

2) 文件处理程序顺序读取且顺序处理记录，记录如何存放可使文件的处理时间最短？

■ 1) 由题目所给条件可知，磁盘转速为27ms/转，因此读出1个记录的时间是  $27 \text{ ms} / 9 = 3 \text{ ms}$ 。读出并处理记录A需要  $3 \text{ ms} + 2 \text{ ms} = 5 \text{ ms}$ ，此时读写头已转到了记录B的中间，因此为了读出记录B，必须再转将近一圈（从记录B的中间到记录B，需要25 ms）。后续7个记录的读取及处理与此相同，但最后一个记录的读取与处理只需5 ms，于是处理9个记录的总时间为  $8 \times (25 + 3 + 2) \text{ ms} + (3 + 2) \text{ ms} = 245 \text{ ms}$ 。这里将旋转一周的时间算在了前一个读取单元中，即将“读取A，旋转”看作一个单元，因此前8个的处理时间为30 ms，最后一个为5 ms。若将旋转算在后一个读取单元，则处理A为5ms，后面8个为 30ms，结果相同。





# Review

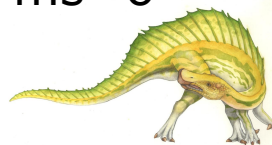
- 7. 假设磁盘的每个磁道分为9个块，现一文件有A, B, ...I共9个记录，每个记录的大小与块的大小相等，设磁盘转速为27 ms/转，每读出一块后需要2 ms的处理时间，若忽略其他辅助时间，试问：

2) 文件处理程序顺序读取且顺序处理记录，记录如何存放可使文件的处理时间最短？

- 2) 由于读出并处理一个记录需要5 ms，当读出并处理记录A时，不妨设记录A放在第1个盘块中，读写头已移动到第2个盘块的中间，为了能顺序读到记录B，应将它放到第3个盘块中，即应将记录按如下顺序存放，见下表。

盘块	1	2	3	4	5	6	7	8	9
记录	A	F	B	G	C	H	D	I	E

这样，处理一个记录并将磁头移动到下一记录的时间为3 ms+2 ms+1 ms=6 ms。所以，处理9 个记录的总时间为6×8 ms+5 ms=53 ms。





# Review

- 1. 缓冲技术的缓冲池通常设立在（ **A** ）中
  - A. 主存
  - B. 外存
  - C. ROM
  - D. 寄存器
- 2. 下面设备中属于共享设备的是（ **C** ）
  - A. 打印机
  - B. 磁带机
  - C. 磁盘
  - D. 磁带机和磁盘
- 3. 在采用SPooling技术的系统中，用户暂时未能打印的数据首先会被送到（ **A** ）存储起来
  - A. 磁盘固定区域
  - B. 内存固定区域
  - C. 终端
  - D. 打印机





# Review

- 4. 下列有关设备独立性的说法中，正确的是（ B ）
  - A. 设备独立性是指I/O设备具有独立执行I/O功能的一种特性
  - B. 设备独立性是指用户程序独立于具体物理设备的一种特性
  - C. 设备独立性是指能够实现设备共享的一种特性
  - D. 设备独立性是指设备驱动程序独立于具体物理设备的一种特性
  
- 5. 系统管理设备是通过一些数据结构来进行的，下面的（ A ）不属于设备管理数据结构。
  - A. FCB
  - B. DCT
  - C. SDT
  - D. COCT





# Review

- 6. 操作系统I/O子系统通常由4个层次组成，每一层明确定义了与邻近层次的接口，其合理的层次组织排列顺序是（ A ）
  - A. 用户级I/O软件、设备无关软件、设备驱动程序、中断处理程序
  - B. 用户级I/O软件、设备无关软件、中断处理程序、设备驱动程序
  - C. 用户级I/O软件、设备驱动程序、设备无关软件、中断处理程序
  - D. 用户级I/O软件、中断处理程序、设备无关软件、设备驱动程序
- 7. 用户程序发出磁盘I/O 请求后，系统的正确处理流程是（ B ）。
  - A. 用户程序→系统调用处理程序→中断处理程序→设备驱动程序
  - B. 用户程序→系统调用处理程序→设备驱动程序→中断处理程序
  - C. 用户程序→设备驱动程序→系统调用处理程序→中断处理程序
  - D. 用户程序→设备驱动程序→中断处理程序→系统调用处理程序







# Review

- 8. 如果I/O设备与存储设备间的数据交换不经过CPU来完成，则这种数据交换方式是（ C ）
  - A. 程序查询方式
  - B. 中断方式
  - C. DMA方式
  - D. 外部总线方式
  
- 9. 在如下几种类型系统中，（ B ）采用忙等待I/O 是合适的。
  - a. 专门用来控制单I/O 设备的系统
  - b. 运行一个单任务操作系统的个人计算机
  - c. 作为一个负载较大的网络服务器的工作站
  - A. a
  - B. a, b
  - C. b, c
  - D. c

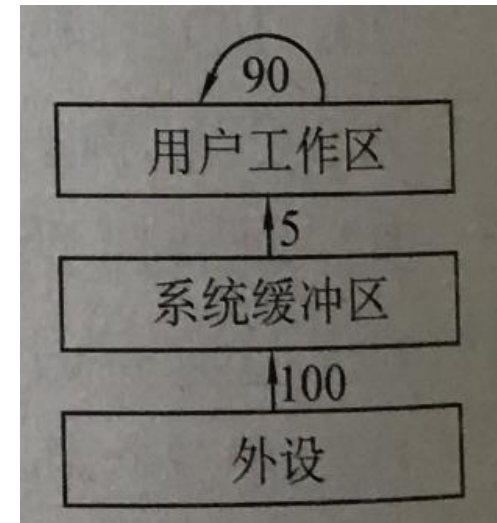




# Review

■ 10. 设系统缓冲区和用户工作区均采用单缓冲，从外设读入1个数据块到系统缓冲区的时间为100，从系统缓冲区读入1个数据块到用户工作区的时间为5，对用户工作区中的1个数据块进行分析的时间为90。进程从外设读入并分析2个数据块的最短时间是（ C ）

- A. 200      B. 295      C. 300      D. 390



第10题分析：

■ 数据块1从外设到用户工作区的总时间为105，在这段时间中，系统缓冲区均为数据块1占据，因此对数据块2无法进行操作。但在数据块1在用户工作区进行分析处理时（时间90），系统缓冲区已经空闲下来，可以让数据块2从外设读入到系统缓冲区，相当于数据块2的整个处理时间比串行执行时节省了90的时间。又1个数据串行执行时，所需时间为 $100+5+90=195$ ，第二块的处理时间为 $195-90=105$ ，合计 $195+105=300$ ，即进程从外设读入并分析2个数据块的最短时间为300，答案为C。





- 11. 什么是DMA方式？它与中断方式的主要区别是什么？
- DMA是Direct Memory Access的缩写，也就是直接存储器访问。DMA是用DMA控制器来控制一个数据块的传输，而CPU只需在一个数据块传输的开始阶段设置好传输所需的控制信息并在传输的结束阶段做进一步处理即可的传输控制方式。其基本思想是在I/O设备和内存间开启一个可以直接传输数据的通路。

中断驱动I/O控制方式是每个数据传输后即发出中断，而DMA方式是在一批数据传输完毕后才中断；中断驱动I/O控制方式的传输是由CPU控制的，而DMA方式中只有数据块传输的开始和结束阶段在CPU控制下，在传输过程中都是由DMA控制器控制的。所以DMA方式相比于中断方式，通过硬件的增加大大减少了中断的次数。





- 12. 考虑56kbit/s调制解调器的性能，驱动程序输出一个字符后就阻塞，当一个字符打印完毕后，产生一个中断通知阻塞的驱动程序，输出下一个字符，然后再阻塞。如果发消息，输出一个字符和阻塞的时间总和为0.1ms（中断），那么由于处理调制解调器而占用的CPU的时间比率是多少？假设每个字符有一个开始位和一个结束位，共占10位。

- 解答：

一个字符占10位，因此在56kbit/s的速率下，每秒传送 $56000/10=5600$ 字符，即产生5600次中断。每次中断需0.1ms，故处理调制解调器占用CPU时间总共为 $5600 \times 0.1\text{ms} = 560\text{ms}$ ，占 $560\text{ms}/1\text{s} = 56\%$  CPU时间。

