网络空间安全 薛凡豪 20211120138

# 第四章 文件管理

第四章 文件系统 的作业如下：p273：1、2、3（1）-3（4）、4（1）-4（9）  
5（1）、5（4）、5（5）、5（8）

第1题、

1. 覆盖：覆盖是一种内存管理技术，用于处理程序大小超过可用内存容量的情况。通过将程序分割成逻辑上相关的模块，并在执行过程中动态加载和卸载这些模块，只在需要时将其加载到内存中，从而有效地利用有限的内存资源。

2. 对换：对换是一种内存管理技术，用于将进程或程序从主存储器（RAM）移出到辅助存储器（通常是硬盘）中，以释放内存空间给其他进程使用。当需要执行的进程不能完全驻留在内存中时，操作系统会将其暂时交换出去，并在需要时再将其换入内存。

3. 页面淘汰：页面淘汰是虚拟内存管理中的一项关键任务。当需要将新的页面加载到内存中时，但内存已满时，操作系统必须选择一个页面将其替换出去。页面淘汰算法用于决定哪些页面应该被替换出内存，以便为新的页面腾出空间。

4. 快表：快表是一种高速缓存，用于加速虚拟内存到物理内存的地址转换过程。在虚拟内存系统中，TLB存储了最近使用的一些虚拟页和对应的物理页的映射关系，以避免每次访问内存时都需要进行完整的地址转换。

5. 地址重定位：地址重定位是一种内存管理技术，用于将程序在内存中的实际物理地址与程序中使用的逻辑地址进行映射。通过地址重定位，操作系统可以将程序加载到任意可用的物理内存地址上，实现进程的独立运行和保护。

6. 抖动：抖动是指系统过度交换（对换）页面导致性能严重下降的现象。当系统中的进程需要的页面数量超过了可用内存容量，导致频繁的页面调度和交换，造成大量的磁盘访问和延迟，使系统效率急剧下降。

7. 内碎片和外碎片：内碎片是指在内存分配中，已经被分配给进程但未被有效利用的内存空间。外碎片是指在内存分配中，已经被分配给进程之外的空闲内存块，但由于它们的分布不连续或太小而无法被利用。内碎片和外碎片都会浪费内存资源，降低内存利用效率。

第2题

1-5:DBDAB

6-10:CCBBD

第3题

1. 越界终端
2. 最佳适应
3. 操作系统
4. 加速虚拟地址到物理地址的转换过程

第4题

(1)正确。

(2)不正确。虚拟系统中，虽然进程可以拥有较大的虚拟地址空间，但是实际的物理内存仍然是有限的，因此仍然存在内存限制。

(3)不正确。OPT（最佳置换）算法是一种理论上的最优算法，但由于无法预知未来的页面访问模式，实际上无法在实际系统中实现。大多数虚拟系统采用其他页面淘汰算法，如LRU（最近最久未使用）算法。

(4)不正确。页面淘汰所花费的时间属于系统开销，因为它涉及到页面的读写和调度操作。

(5)不正确。在页式存储管理中，用户不需要手动将自己的程序划分成相等的页，这是由操作系统自动进行的。

(6)不正确。存储保护的目的是保护进程之间和操作系统之间的内存空间，以防止相互干扰和越权访问，而不是限制内存的分配。

(7)不正确。最近最少使用（LRU）页面调度算法是指淘汰最近未使用的页面，而不是使用次数最少的页面。

(8)不正确。可重入程序是指可以被多个进程同时调用而不会产生冲突的程序，与程序共享无直接关系。

(9)不正确。在请求分页存储管理系统中，页面大小的增加会导致每个进程所需的页数减少，但并不意味着缺页中断次数会减少一半。缺页中断次数受到多种因素的影响，包括程序访问模式和内存使用方式等。

第5题

（1）页表是一个数据结构，用于记录虚拟页面和物理页面之间的映射关系。每个进程都有自己的页表，其中的每个表项对应着虚拟地址空间中的一页。页表的作用包括：虚拟地址转换和内存保护。块表是一个数据结构，用于记录物理块的使用情况和对应的虚拟页面。每个物理块都有一个对应的块表项，用于记录该块是否被分配给某个虚拟页面。块表的作用包括：页面分配、页面淘汰、空闲块管理

（4）缺页中断与一般中断在引发原因、处理机制、频率、响应时间和处理方式等方面存在明显的区别，主要是因为缺页中断是由于虚拟内存的页面调度机制而产生的一种特殊中断。

（5）根据最先适应分配算法，进程会被分配给满足其大小要求的第一个空闲区。现在来逐个判断这5个进程是否可以全部装入主存：

1. 进程J1需要13K，可以分配给第一个空闲区45K。

2. 进程J2需要36K，可以分配给第一个空闲区剩余的32K（45K - 13K）。

3. 进程J3需要108K，无法分配给第一个空闲区32K，继续寻找下一个空闲区。

4. 进程J3需要108K，可以分配给第二个空闲区40K。

5. 进程J4需要43K，可以分配给第一个空闲区剩余的19K（32K - 13K）。

6. 进程J5需要195K，无法分配给第一个空闲区19K，继续寻找下一个空闲区。

7. 进程J5需要195K，无法分配给第二个空闲区40K，继续寻找下一个空闲区。

8. 进程J5需要195K，无法分配给第三个空闲区15K，继续寻找下一个空闲区。

9. 进程J5需要195K，可以分配给第四个空闲区200K。

因此，采用最先适应分配算法无法将这5个进程按照J1-J5的次序全部装入主存，因为进程J3无法分配到合适的空闲区。

要使主存利用率最高，可以采用最优适应分配算法。最优适应分配算法会选择与进程大小最接近的空闲区进行分配。在这个例子中，最优适应分配算法的执行顺序如下

1. 进程J1需要13K，可以分配给第一个空闲区45K。

2. 进程J2需要36K，可以分配给第二个空闲区40K。

3. 进程J3需要108K，可以分配给第四个空闲区200K。

4. 进程J4需要43K，可以分配给第一个空闲区剩余的32K（45K - 13K）。

5. 进程J5需要195K，可以分配给第四个空闲区剩余的92K（200K - 108K）。

通过最优适应分配算法，这5个进程可以按照J1-J5的次序全部装入主存，实现最高的主存利用率。

（8）UNIX采用的交换空间管理策略是基于页面的虚拟内存管理。

UNIX采用页面交换的管理策略可以实现扩展可用的内存空间、提高系统的灵活性、实现虚拟内存管理

# 第五章 文件系统

第5章 文件系统 作业：1、2、3（1）-3（4）、4（1）-4（9）、5（1）、5（4）、5（5）、5（8）

第1题

（1）文件：在计算机系统中，文件是指存储在持久性存储介质上的一组相关数据的集合。文件可以包含文本、图像、音频、视频等各种类型的数据。文件是计算机存储和组织数据的基本单位，通过文件可以持久地保存和访问数据。

（2）文件目录：文件目录是文件系统中用于组织和管理文件的一种数据结构。它是一个包含文件和目录项的表格或树状结构，记录了文件在存储介质上的位置和相关属性信息。文件目录提供了一种逻辑视图，使用户可以根据文件名或路径来查找和访问文件。

（3）目录文件：目录文件是指文件系统中的一种特殊文件，用于存储文件目录的内容。目录文件包含了文件和目录项的信息，可以通过读取目录文件来获取文件系统中的文件结构和相关属性。目录文件通常具有特定的格式和组织方式，可以按照层次结构组织文件和子目录。

（4）文件控制块 (FCB)：文件控制块是操作系统中用于管理文件的数据结构。每个文件在文件系统中都有对应的文件控制块，用于记录文件的属性、存储位置、权限、大小、时间戳等信息。文件控制块中还包含了与文件相关的操作和指针，如读取、写入、删除等操作的方法和指向文件数据的指针。文件控制块是操作系统对文件进行管理和访问的重要数据结构之一。

第2题

1-5：AABBA

6-10：BBCAC

第3题

1. 流式文件、记录式文件
2. 顺序存取、随机存取
3. 绝对路径、相对路径
4. **位示图法、空闲表法、链表法、索引节点法。**

第4题

（1）错误。同一文件系统中是允许文件同名的，但在同一目录下不允许文件同名，否则会引起混乱。

（2）正确。

（3）错误。文件的索引表通常不会全部存放在文件控制块中，而是存放在磁盘的其他区域或数据结构中。

（4错误。流式文件是指按顺序读取和写入的文件，它们有结构和记录的概念。

（5）错误。文件目录并非必须常驻内存，它通常存放在磁盘或其他外存储介质上，而在需要时可以加载到内存中进行操作。

（6）错误。打开文件是指操作系统为进程分配资源并建立文件与进程之间的连接，通常会创建一个与文件相关的数据结构，如文件描述符或文件句柄，而不是创建一个文件控制块。

（7）错误。磁盘上物理结构为连接结构的文件可以支持随机存取。

（8）正确。文件目录一般存放在外存储介质中，如磁盘上的某个区域。

（9）正确。在对文件进行读写操作之前，通常需要先打开文件以获取对文件的访问权限和建立与文件的连接。

第5题

（1）

1. 卷：卷是存储系统中的一个逻辑单元，通常是由一个或多个物理设备（如磁盘驱动器）组成的。它被用于存储和管理数据，可以被格式化为文件系统，并提供存储空间供文件存储和组织。卷可以是物理上的独立设备，也可以是逻辑上的划分。

2. 块：块是文件系统中的最小存储单位。它表示存储介质（如磁盘）上的连续一段固定大小的数据。块通常由扇区组成，扇区是存储介质上的物理单位。块的大小在不同的文件系统中可以有所不同，常见的块大小为4KB或8KB。

3. 记录：记录是文件系统中的逻辑单位，它是由一组相关数据字段组成的数据单元。记录的大小可以根据应用程序和文件类型的需求而有所变化。在数据库系统中，记录通常用于表示实体或数据项。

4. 文件：文件是文件系统中的基本组织单位，它是一组相关数据的命名集合。文件可以包含一个或多个记录，记录可以按照不同的格式和结构组织。文件是用户和应用程序对数据进行组织、存储和访问的逻辑单位。

关系：

一个卷可以包含一个或多个文件，卷提供了文件存储的逻辑容器。

文件由一系列块组成，块是存储和管理文件数据的最小单位。

文件可以按照记录进行组织，每个记录包含一个或多个字段。

记录的大小和组织方式由文件的类型和应用程序的需求决定。

总的来说，卷是一个存储的逻辑单元，文件是逻辑组织的基本单位，块是文件存储的最小单位，而记录是文件中数据的逻辑单位。它们在文件系统中相互关联，形成了数据的层次化结构。

（4）

文件的逻辑结构是指文件内部数据的组织方式，它定义了文件中数据元素之间的关系和排列方式。文件的逻辑结构决定了对文件进行读取、写入和操作时所采用的方法和算法。

常见的文件逻辑结构包括顺序结构、索引结构、随机结构、紧凑结构

（5）

文件的物理结构是指文件在存储介质上的实际存储方式，它定义了文件数据在物理存储介质（如磁盘、固态硬盘等）上的布局和组织方式。

常见的文件物理结构包括顺序结构、连接结构、索引结构、散列结构

（8）

文件目录是文件系统中用于组织和管理文件的一种数据结构，它记录了文件的元数据和存储位置信息，充当了文件系统的索引和导航功能。文件目录用于存储文件的名称、大小、类型、权限、创建时间、修改时间等属性信息，以及文件在存储介质上的物理位置或指针。

文件目录中常包含以下信息：

1. 文件名

2. 文件属性

3. 文件位置

4. 文件关联

常用的目录结构形式有以下几种：

1. 层次目录结构

2. 扁平目录结构

3. 索引目录结构

4. 直接目录结构

# 第6章 设备管理

第6章 设备管理 作业： 3、4、6、7、9

3.通常由接口电路、控制逻辑、数据缓冲区、设备接口组成

4.

* 中断触发或异常发生：在计算机系统中，当发生与正在执行的程序无关的事件或外部设备请求时，会触发中断；而异常是指程序执行过程中出现的错误或不正常的情况，例如除零错误、内存访问错误等。
* 中断/异常请求识别：中央处理器（CPU）会不断检测是否有中断或异常请求发生。通过检查特定的中断/异常信号或相关的标志位，CPU能够确定是否有中断或异常请求需要处理。
* 中断/异常处理程序调用：当中断或异常请求被识别后，CPU会根据中断/异常的类型，通过中断向量表或异常向量表找到对应的中断/异常处理程序的入口地址。
* 上下文保存：在调用中断/异常处理程序之前，CPU会保存当前执行程序的上下文信息，包括程序计数器（PC）、寄存器内容、标志位等。这样做是为了保护当前程序的执行状态，以便在中断/异常处理程序执行完毕后能够正确恢复。
* 中断/异常处理程序执行：CPU跳转到中断/异常处理程序的入口地址开始执行对应的处理代码。中断/异常处理程序会根据具体的中断/异常类型进行相应的处理操作，例如读取外部设备数据、处理错误、进行错误恢复等。
* 上下文恢复：当中断/异常处理程序执行完毕后，CPU会根据之前保存的上下文信息，恢复之前被中断的程序的执行状态。这包括恢复程序计数器、寄存器内容、标志位等。
* 重新执行中断/异常指令：在上下文恢复后，CPU会继续执行被中断的程序，从被中断处或异常处重新执行。

6.

I/O控制发展的主要推动因素主包括：需求增长、技术进步、多样化的设备类型、提高系统性能、用户体验要求

7.

* DMA（直接内存访问）是一种数据传输技术，它可以绕过CPU，直接在主存和I/O设备之间进行数据传输，从而提高数据传输效率。下
* 初始化DMA控制器：首先，CPU通过向DMA控制器发送指令和参数来初始化DMA控制器。这些参数包括源地址、目的地址、传输长度等。
* 配置DMA通道：DMA控制器通常有多个DMA通道，每个通道可以独立进行数据传输。在需要进行数据传输的设备和主存之间选择一个空闲的DMA通道。
* 启动DMA传输：一旦DMA控制器被配置好，CPU发出启动信号，告诉DMA控制器开始数据传输。DMA控制器将根据配置的参数控制数据传输的流程。
* 直接内存访问：DMA控制器在传输数据时绕过CPU，直接从设备读取数据并写入主存，或者从主存读取数据并写入设备，而无需CPU的干预。DMA控制器使用总线控制信号来访问主存，并且可以在数据传输过程中进行地址递增、数据缓存等操作。
* 数据传输完成中断：一旦DMA传输完成，DMA控制器会发出一个中断信号，通知CPU传输完成。CPU可以通过查询DMA控制器的状态来确认传输是否成功，并继续处理其他任务。
* 整个DMA的工作流程可以概括为初始化DMA控制器、配置DMA通道、启动DMA传输、直接内存访问以及传输完成中断。通过使用DMA技术，可以大大提高数据传输的效率，减少CPU的负载，提升系统的性能。

9

引入缓冲区的好处包括：

* 提高效率：通过将数据暂存到缓冲区中，可以减少数据传输的次数。当数据量较大时，通过一次性传输整个缓冲区的数据，可以减少传输的开销和延迟。
* 平衡数据速率：当数据源和目的地的数据速率不匹配时，缓冲区可以作为一个缓冲媒介，平衡数据的输入和输出速率，避免数据丢失或传输阻塞。
* 提高并发性能：缓冲区可以提供数据的异步传输，即数据可以在缓冲区中存储和处理，而不需要实时进行数据传输。这样可以提高系统的并发性能，允许多个任务同时进行数据处理。

在UNIX System V中，常见的缓冲区有以下几种：

* 标准I/O缓冲区：提供了标准输入输出函数（如printf、scanf）的缓冲区支持。它使用流（FILE）作为中间层，通过缓冲区提高了I/O操作的效率。
* 文件系统缓冲区：用于缓存磁盘上的文件数据，以减少对磁盘的频繁读写操作。文件系统缓冲区可以提高文件系统的性能，并提供对文件的快速访问。
* 进程间通信缓冲区：用于进程间通信机制（如管道、消息队列、共享内存等）中的数据传输。进程间通信缓冲区允许进程将数据存放在其中，并由系统进行数据传输。