**作业一**

**请论述进程引入对于以多道批处理为开端的现代操作系统的重要性和必要性，再从同步，互斥，死锁等角度，对如何保证进程高效安全工作浅谈自己的认识。**

进程引入对于以多道批处理为开端的现代操作系统具有重要性和必要性的原因主要包括以下几点：

1. **提高系统利用率：**进程引入使得操作系统能够同时处理多个程序，从而提高系统资源的利用率。通过将多个作业同时加载到内存中执行，系统可以充分利用CPU、内存和其他资源，提高整体的吞吐量和效率。

2. **实现并发操作：**进程引入使得多个程序可以同时执行，实现了并发操作。操作系统可以轮流分配CPU时间片给各个进程，使它们看起来在同时执行。这对于提高系统的响应速度、用户体验和系统的并发性能至关重要。

3. **提供进程间通信和协作机制：**进程引入为不同的进程之间提供了通信和协作的机制。进程可以通过共享内存、消息传递等方式进行信息交换和共享资源，实现进程间的协作和数据交互，从而满足复杂应用程序的需求。

对于进程的高效安全工作，需要考虑以下几个关键点：

1. **同步机制：**在多个进程并发执行的情况下，可能会存在多个进程同时访问共享资源的竞态条件。为了避免数据的不一致性和错误，需要采用适当的同步机制，如互斥锁、信号量、条件变量等，来保护共享资源的访问顺序和互斥访问。

2. **互斥机制：**互斥是指在同一时刻只允许一个进程访问共享资源。通过引入互斥机制，如互斥锁，可以确保多个进程不会同时访问共享资源，从而避免数据的不一致性和冲突。

3. **死锁避免和处理：**死锁是指多个进程因竞争资源而陷入互相等待的状态，无法继续执行的情况。为了保证进程的高效安全工作，需要采取措施来避免和处理死锁。常见的方法包括资源分配策略、死锁检测和恢复、资源预申请等，以保证系统能够正常运行并尽量避免死锁的发生。

进程引入对于现代操作系统的重要性和必要性在于提高系统的利用率、实现并发操作以及提供进程间通信和协作机制。同时，为了保证进程的高效安全工作，需要采取适当的同步机制、互斥机制和死锁处理策略，以确保系统的稳定性和数据的一致性。

**作业二**

**请归类并列举典型的内存分配存储方式，浅谈如何提高内存的使用效率，并对比硬盘存储空间分配存储有哪些共性和特性**

1. **内存分配管理** 
   1. **连续分配存储管理** 
      1. 单一连续分区分配
      2. 固定分区分区分配
      3. 动态分区分区分配
      4. 动态可重定位分区分配
   2. **离散分配存储管理** 
      1. 分页存储管理
      2. 分段存储管理
      3. 段页式存储管理
2. **如何提高内存的使用效率**：

1.**内存管理算法优化**：选择合适的内存分配算法和回收策略，以最大限度地减少内存碎片问题和提高内存利用率。例如，可以使用更高效的动态分区分配算法，或者采用内存池技术减少频繁的内存分配和释放操作。

2.**内存复用**：尽量复用已分配的内存，避免频繁的内存申请和释放操作。可以使用对象池、缓存技术等手段，将已经分配的内存对象重复利用，减少内存分配和释放的开销。

3.**垃圾回收**：对于使用堆式分配的内存管理方式，垃圾回收是一种重要的技术手段。通过自动回收不再使用的内存，可以避免内存泄漏和减少手动内存管理的复杂性。

1. **磁盘存储空间管理方式** 
   1. **连续分配**：一个文件存储在一个连续的存储空间中
   2. **链接组织方式** 
      1. 显式链接：保存一个文件分配表到内存中，它可以不按照顺序访问
      2. 隐式链接：文件目录项含有指向文件的第一个存储盘块和最后一个存储盘块的指针，每一个盘块都包含指向下一个盘块的指针，这些都保存在磁盘中，按照顺序访问，访问速度慢
   3. **NTFS的组织方式**
   4. **FAT的组织方式**
   5. **索引组织方式** 
      1. 单级索引组织
      2. 多级索引组织
      3. 增量式索引组织

**硬盘存储空间的分配存储与内存分配存储存在一些共性和特性：**

1. **分配方式**：硬盘存储空间通常采用文件系统进行管理，文件系统将硬盘空间划分为不同的文件块或簇，按需分配给文件。类似于动态分区分配的概念，文件系统会根据文件的大小和空闲空间的情况，动态分配合适大小的存储空间。

2. **碎片问题**：硬盘存储空间也存在内部碎片和外部碎片问题。内部碎片是指文件分配的存储空间大于文件实际所需的空间，导致空间浪费；外部碎片是指硬盘上存在一些零散的未分配空间，无法被有效利用。

3. **存储管理**：硬盘存储空间的管理通常由文件系统负责，包括空间的分配、回收和碎片整理等操作。文件系统会维护存储空间的元数据信息，并提供相应的接口和算法来管理和操作存储空间。

4. **持久性：**相比于内存存储，硬盘存储具有持久性，可以长期保存数据。即使系统断电或重启，数据仍然可以从硬盘中恢复。

总之，内存分配存储和硬盘存储空间分配存储都需要考虑空间的分配和回收、碎片问题以及存储管理等方面。不同的是，内存分配更加关注实时性和性能，而硬盘存储则注重数据的持久性和长期保存。

# 作业三

**操作系统主要管理计算机的哪些事务？总结其中涉及分配调度的事务，并简介其中的算法；总结并介绍设计虚拟的事务；总结及介绍涉及中间件的事务**

**操作系统主要管理计算机的事务如下：**

* **进程管理**：操作系统负责进程的创建，撤销；对诸多进程的运行进行协调，实现进程之间的信息交换，以及按照一定的算法把处理机分配给进程

**分配调度事务：**

* 1. **对作业的调度方式：**
     + **先来先服务**

先来先服务调度算法：系统按照作业到达先后的次序来进行调度，既它是优先考虑等待时间最长的作业，而不管该作业的执行时间长短

* + - **短作业优先**

短作业优先调度算法：系统按照作业的执行时间从就绪队列中调度一个或多个进程，既他是优先考虑作业的执行时间的长短，而不是考虑作业的等待时间

* + - **优先级调度**

优先级调度算法：由外部赋予作业的优先级，系统根据该优先级进行调度

* + - **高响应比优先调度**

高响应比优先调用算法：即考虑作业的执行时间，又考虑作业的等待时间，既照顾短作业，也不会导致长作业的等待时间过长，优先权=（等待时间+要求服务时间）/要求服务时间

* 1. **对进程的调度方式**
     + **时间片轮转**

时间片轮转：为就绪队列的每个进程运行相同的时间片

* + - **多级反馈队列调度算法**

多级反馈队列：设置多个队列，优先级按照1到n的优先权排列，每个队列都采用FCFS的算法，最后按照队列的优先级进行调度

* + - **基于公平原则的调度算法**
  1. **实时调度算法**
     + **最早截止时间EDF算法**（抢占式，非抢占式）

根据任务的截止时间确定任务的优先级，任务截至的时间愈早，其优先级愈高

* + - **最低松弛度优先LLF算法**（抢占式，非抢占式）

根据任务的紧急程度确定优先级，任务紧急程度愈高，赋予该任务的优先级愈高 松弛度= 必须完成的时间 - 当前时间 - 其本身的运行时间

* **存储管理**：操作系统负责分配、回收和管理内存资源，以便进程能够访问和使用内存。

**分配调度**：

* 1. **程序（经过编译——链接——装入）的装入**：
     + **以绝对方式装入**

用户程序经过编译后，将产生绝对地址（物理地址）的目标代码

* + - **可重定向式装入**

用户程序经过编译之后所产生的若干个模块，他们的起始地址通常是0开始的，程序中的其他地址都是相对于起始地址计算的

* + - **动态运行时装入**

在把装入模块装入内存后，并不立即把装入模块中的逻辑地址转换为物理地址，而是把这种地址转换推迟到程序真正要执行的时候才进行

* 1. **程序的链接：**
     + **静态链接**

在程序运行之前，先将各个模块及他们所需的库函数链接成一个完整的模块

* + - **装入时链接**

边装入边链接

* + - **运行时动态链接**

对某些模块的链接推迟到程序执行时运行

* 1. **程序的装入：**
     + **连续分配存储管理：**
       - **单一连续分配**

整个内存的用户空间由一个程序独占

* + - * **固定分区**

将整个内存分为多个固定大小的区域，每个区域只装入一个程序

* + - * **动态分区**

根据进程的实际需要，动态的为进程分配内存

* + - * 1. **分配内存**

**基于顺序搜索的动态分区分配算法**

首次适应

循环首次适应

最佳适应

最坏适应

**基于索引的动态分区分配算法**

快速适应

伙伴系统

哈希算法

* + - * 1. **回收内存**
      * **动态可重定向分区**

**紧凑**：将多个小分区拼接成一个大的分区

**动态重定向**：将原来的起始地址替换为新的起始地址

* + - **离散式分配管理**：
      * **分页存储管理**

使用固定大小的页框和页表来进行地址转换和内存分配。

* + - * **分段存储管理**

将进程的地址空间划分为逻辑上相关的段，并使用段表进行地址映射和分配。

* + - * **段页式存储管理**

结合以上两者

**虚拟存储管理：**

将主存（RAM）和辅助存储设备（如硬盘）结合起来，为进程提供了一个似乎是无限大的地址空间，称为虚拟地址空间。虚拟存储管理的主要目标是提供比实际可用物理内存更大的地址空间，并有效地管理和调度内存的分配和回收。

* 1. **虚拟地址空间**：每个进程在执行时拥有自己的虚拟地址空间，通常是一个连续的地址范围。这个地址空间包括了代码、数据、堆栈等段，并且通常比实际的物理内存要大。
  2. **分页技术**：虚拟存储管理通过将进程的虚拟地址空间划分成固定大小的页（通常为4KB或更大），并将其映射到实际的物理页框上来实现。这样，当进程访问一个虚拟页时，操作系统会将相应的物理页框调入主存，并建立虚拟页与物理页框之间的映射关系。
  3. **页面置换算法**：由于虚拟地址空间比物理内存大，不是所有的页都可以同时存放在主存中。因此，当物理内存不足时，需要使用页面置换算法来选择哪些页应该被替换出去，腾出空间给新的页。常见的页面置换算法有最近最久未使用（LRU）、先进先出（FIFO）和时钟页面置换算法等。
  4. **页面调度和请求调页**：当进程访问一个不在主存中的页时，会触发一个页面调度过程，即将相应的页面从辅助存储器中调入主存。请求调页是一种基于页面错误的调度方式，只有在访问的页面不在主存中时才触发页面调度。
  5. **页面表**：操作系统使用页面表来管理虚拟地址到物理地址的映射关系。每个进程都有自己的页面表，用于记录虚拟页与物理页框之间的映射关系。页面表的条目包含了虚拟页号和对应的物理页框号。

**中间件**：

在文件管理中，中间件事务通常指与文件系统相关的中间件组件，如文件系统驱动程序。文件系统驱动程序是操作系统和底层文件系统之间的接口，它负责管理文件系统的操作，包括文件的创建、读取、写入和删除等。中间件事务确保文件管理的正常操作和与底层文件系统的交互。

* **设备管理**：完成用户进程提出的I/O请求，提高cpu与I/O设备的利用率

**分配调度：**

**常见的设备分配算法：**

* 1. **先来先服务 (FCFS)**：按照设备请求的先后顺序进行调度。当一个进程请求设备时，如果设备空闲，则立即分配给该进程；如果设备正忙，则将该进程加入设备等待队列，按照先来先服务的原则进行调度。
  2. **最短作业优先 (SJF)：**选择估计完成时间最短的设备请求进行调度。根据设备请求的预估执行时间，选择下一个执行的请求是估计时间最短的，以最小化等待时间。
  3. **优先级调度**：为每个设备请求分配一个优先级，并根据优先级选择下一个执行的请求。可以根据进程的重要性、优先级或其他因素来确定设备请求的优先级。
  4. **时间片轮转调度**：将设备时间划分为固定大小的时间片，每个设备请求分配一个时间片，当时间片用完后，轮到下一个请求执行。该算法保证了设备请求公平地获得使用权，但可能导致一些请求等待时间较长。
  5. **最短剩余时间优先 (SRTF)**：在设备请求等待队列中，选择剩余时间最短的请求进行调度。当有新的请求到达或当前请求完成时，重新评估所有请求的剩余执行时间，选择最短的请求执行。
  6. **最高响应比优先 (HRRN)**：根据设备请求的等待时间和执行时间，计算响应比，并选择响应比最高的请求进行调度。响应比定义为 (等待时间 + 执行时间) / 执行时间，该算法优先选择等待时间较长且执行时间较短的请求。

**虚拟事务：**

将一台物理I/O设备虚拟成多台逻辑设备，允许多个用户共享一台设备

技术 ： 假脱机（Spooling） 技术

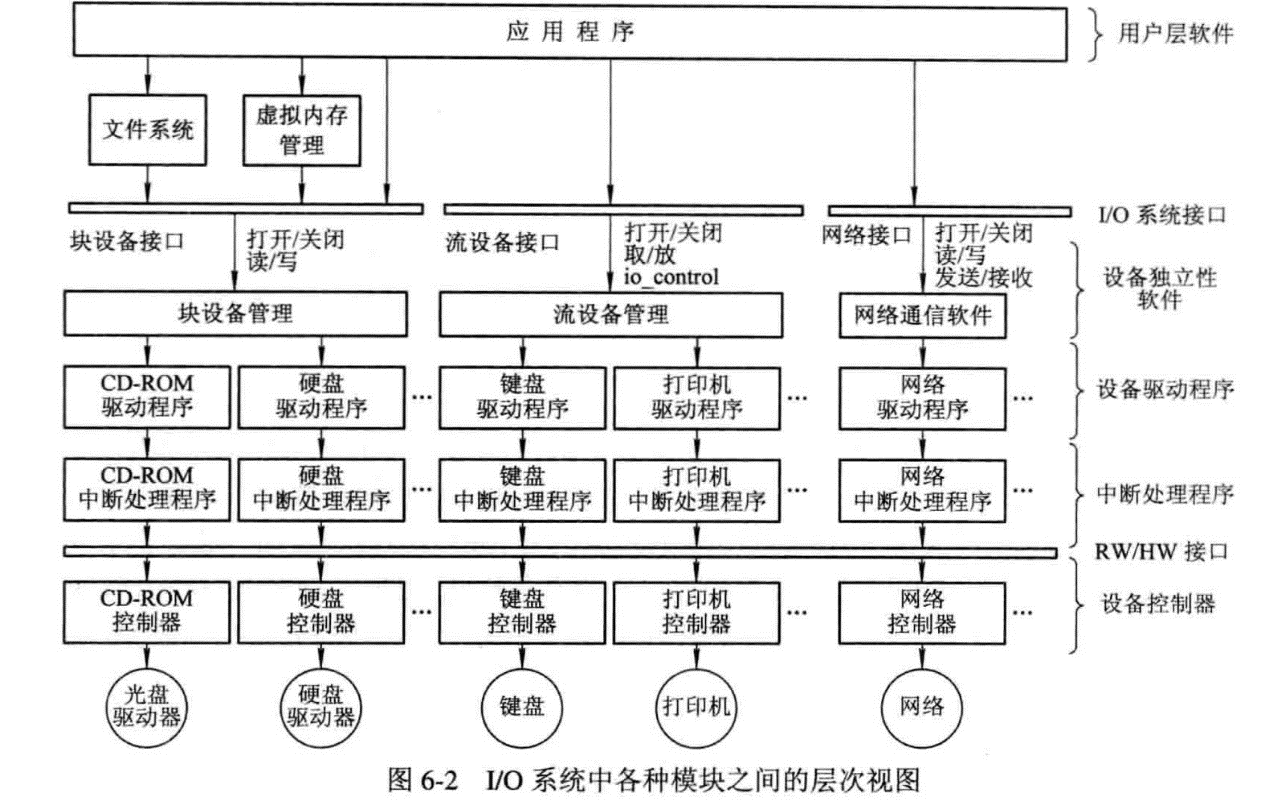
假脱机（Spooling）技术，用于将输入/输出（I/O）任务从实际的设备（如打印机或磁盘驱动器）解耦，并通过使用中间存储区（称为"虚拟设备"或"作业池"）来实现对这些任务的缓冲和调度。

假脱机技术的主要目的是提高系统的效率和吞吐量，避免进程或用户需要等待设备的繁忙状态，同时允许多个任务并行执行。以下是假脱机技术的关键特点和工作原理：

* 1. **缓冲区：**假脱机技术使用缓冲区作为虚拟设备来存储输入或输出数据。这些数据可以是打印作业、磁盘写入或其他需要在设备上执行的任务。
  2. **作业池：**假脱机技术使用作业池作为存储输入/输出任务的队列。任务按照先进先出的原则排队，并在设备空闲时进行调度和执行。
  3. **并行处理：**通过使用作业池，假脱机技术允许多个任务并行处理。当一个任务完成时，下一个任务可以立即开始执行，而不需要等待实际设备的可用性。
  4. **资源调度：**假脱机技术负责调度作业池中的任务，并将它们传送给实际的设备进行处理。这种调度可以根据优先级、时间片轮转或其他调度算法来进行。
  5. **错误处理：**假脱机技术可以处理设备错误或故障的情况。当出现错误时，它可以重新调度任务或通知用户进行相应的处理。

常见的应用假脱机技术的场景是打印作业的处理。打印任务可以先存储在作业池中，而不必等待实际打印机空闲。假脱机技术将任务按顺序传送给打印机，并在打印机完成当前任务后立即开始下一个任务的处理。这样，用户可以继续进行其他工作，而不必等待打印作业完成。

**中间件：**



* 1. **设备控制器**：负责控制一个或多个I/O设备，以实现I/O设备和计算机之间的数据交换
  2. **RW/HW接口**：用于连接软件与硬件
  3. **中断处理程序**： 处于I/O系统的底层，直接与硬件进行交互
  4. **设备驱动程序：** 处于次底层，是进程和控制器之间的通信程序，功能：将上层发来的抽象I/O请求，转换为对I/O设备的具体命令和参数，并把它装入到设备控制其中的命令和参数寄存器中
  5. **设备独立性软件：**包括设备命名、设备分配、数据缓冲等软件
  6. **I/O系统接口：**在I/O系统与高层接口中，根据设备类型的不同，又进一步分为若干个接口。主要包括： a. 块设备接口 数据的存取和传输都是以数据块为单位的设备。基本特征是传输速率较高、可寻址。磁盘设备的I/O常采用DMA方式。 b. 流设备接口 数据的存取和传输是以字符为单位的设备。如键盘、打印机等。基本特征是传输速率较低、不可寻址，常采用中断驱动方式。
  7. **网络通信接口** a. 通过某种方式，把计算机连接到网络上。 b. 操作系统必须提供相应的网络软件和网络通信接口，使得计算机能通过网络与网络上的其它计算机进行通信，或上网浏览。
* **文件管理**：

对用户文件和系统文件进行管理以方便用户使用，并保证文件的安全性

**分配调度**：

* 1. **连续分配：**连续分配是最简单的文件分配策略，它将文件存储在磁盘上连续的空闲区域中。文件在磁盘上占据一段连续的存储空间，方便读取和写入。然而，连续分配可能导致外部碎片问题，即磁盘上存在大量不连续的空闲空间，无法容纳大文件。
  2. **链接分配**：链接分配通过使用文件的链接列表来存储文件的数据。每个文件包含一个链接列表，其中每个链接指向磁盘上存储该文件的数据块。链接分配解决了连续分配中的外部碎片问题，但可能会导致访问文件时的额外开销，因为需要遍历链接列表。
  3. **索引分配**：索引分配使用索引节点（inode）来管理文件的存储。每个文件都有一个唯一的索引节点，该节点包含文件的元数据（如文件大小、权限等）和指向实际数据块的指针。索引节点中的指针可以是直接指针、间接指针或双重间接指针，以支持不同大小的文件。索引分配具有较低的访问开销，但会占用额外的磁盘空间来存储索引节点。
  4. **FAT（文件分配表）**：FAT是一种文件系统中常用的分配调度策略，例如FAT16和FAT32。FAT使用文件分配表来管理磁盘上的文件存储。文件分配表是一个表格，每个表项表示磁盘上的一个簇（cluster）。文件在磁盘上被分割成多个簇，并通过文件分配表中的指针链来跟踪文件的簇分布。FAT具有较低的存储开销和访问开销，但可能会产生碎片问题。

在文件管理的调度方面，涉及到文件的读取和写入操作。调度算法的目标是提高文件系统的性能和资源利用率。一些常见的文件调度算法包括：

* 1. **先来先服务（FCFS）**：按照请求的先后顺序进行调度，即先到达的请求先执行。
  2. **最短寻道时间优先（SSTF）**：选择离当前磁道最近的请求进行调度，以最小化磁头移动时间。
  3. **扫描（SCAN）**：磁头按照一个方向（如内向外或外向内）移动，执行请求，直到达到磁道的边界，然后改变方向继续执行。
  4. **循环扫描（C-SCAN）**：类似于扫描算法，但当到达磁道的边界时，磁头立即返回到另一端，形成一个循环。

**中间件：**

在文件管理中，中间件事务是指位于操作系统和文件系统之间的软件层，用于提供文件管理的功能和接口。以下是文件管理的中间件事务的一些示例：

* 1. **文件系统接口**：中间件事务可以提供文件系统的抽象接口，使应用程序可以使用统一的方式来操作不同类型的文件系统。它封装了底层文件系统的细节，提供了一致的文件访问接口，使开发人员能够以更高层次的抽象来处理文件操作。
  2. **文件缓存管理**：中间件事务可以实现文件缓存管理，将常用的文件数据缓存在内存中，提高文件访问的速度和效率。它可以根据文件的使用模式和访问频率，进行缓存策略的管理，如LRU（最近最少使用）算法或LFU（最不经常使用）算法。
  3. **文件索引和搜索：**中间件事务可以提供文件索引和搜索功能，加快文件的查找和检索过程。它可以建立索引结构，记录文件的元数据和位置信息，使得应用程序可以通过关键字或属性进行文件的快速搜索和定位。
  4. **文件传输和同步：**中间件事务可以实现文件传输和同步功能，将文件从一个位置或系统传输到另一个位置或系统，并确保文件的一致性和完整性。它可以处理文件传输的错误处理、重试机制和同步策略，以保证文件的安全传输和正确性。
  5. **文件权限和访问控制：**中间件事务可以提供文件权限和访问控制的管理功能，定义文件的访问权限和安全策略。它可以实现用户身份验证、权限验证和文件加密等机制，保护文件的机密性和完整性