

操作系统复习

第一章

1. 操作系统的目标

方便性、有效性、可扩充性和开放性。

2. 操作系统的作用

- 1) OS 作为用户与计算机硬件系统之间的接口
- 2) OS 作为计算机系统资源的管理者
- 3) OS 实现了对计算机资源的抽象

3. 操作系统的发展过程

- 1) 单批道处理系统
- 2) 多批道处理系统（无交互性）
- 3) 分时系统（及时响应，满足人机交互需求）
- 4) 实时系统（及时可靠）

4. 单批道处理系统的处理过程

首先由监督程序将磁带上的第一个作业装入内存，并把运行控制权交给该作业；当该作业处理完成时，又把控制权交还给监督程序，再由监督程序把磁带上的第二个作业调入内存。计算机系统就这样自动地一个作业紧接一个作业地进行处理，直至磁带上的所有作业全部完成，这样便形成了早期的批处理系统。

5. 单批道处理系统的优缺点

优点：解决人机矛盾和 CPU 与 IO 设备速度不匹配问题，提高系统资源的利用率和系统吞吐量。

缺点：单批道处理系统不能充分地利用系统资源。

6. 多批道处理系统的优缺点

优点：资源利用率高、系统吞吐量大。

缺点：平均周转时间长、无交互能力

7. 单批道处理系统和多批道处理系统的区别

单通道批处理系统在内存中只有一个作业，而多通道批处理系统可以在内存中同时托管多通道程序。

8. 操作系统的基本特征

并发、共享、虚拟、异步

9. 并行与并发的区别

并行性是指在两个或多个事件在同一时刻发生。并发性是指两个或多个事件在同一时间间隔内发生。并发性在一定时间内宏观上有多个程序在同时运行;在微观上程序分时的交替执行。

10. 操作系统的主要功能（简单了解）

处理机管理功能、存储器管理功能、文件管理功能、设备管理功能、用户接口

11. 操作系统的定义

操作系统是一组能有效地组织和管理计算机硬件和软件资源，合理地对各类作业进行调度，以及方便用户使用的程序的集合。

第二章

1. 进程和线程的区别

- (1) 调度：线程作为调度和分配的基本单位，进程作为拥有资源的基本单位
 - (2) 并发性：不仅进程之间可以并发执行，同一个进程的多个线程之间也可并发执行
 - (3) 拥有资源：进程是拥有资源的一个独立单位，线程不拥有系统资源，但可以访问隶属于进程的资源。
 - (4) 独立性：在同一进程中的不同线程之间的独立性比不同进程之间的独立性低得多
 - (5) 系统开销：在创建或撤消进程时，由于系统都要为之分配和回收资源，导致系统的开销明显大于创建或撤消线程时的开销。
 - (6) 支持多处理机系统：在单线程进程中，该进程只能运行在一个处理机上。在多线程进程中，可以将一个进程的多个线程分配到多个处理机上，使其并行执行。
- (引入进程的目的是为了使得多个程序能并发执行，以提高资源利用率和系统吞吐量。引入线程的目的是为了减少程序在并发执行时所付出的时空开销，使 OS 具有更好的并发性。)

2. 进程的基本状态及转换

1) 进程的三种基本状态

就绪状态：指进程已处于准备好运行的状态，即进程已分配到除 CPU 以外的所有必要资源或，只要只获得 CPU，便可立即执行。

执行状态：指进程已获得

阻塞状态：指正正在执行的进程由于发生某事件（如 I/O 请求、申请缓冲区失败等）暂时无法继续执行时的状态，即亦进程的执行受到阻塞。

2) 三种基本状态的转换

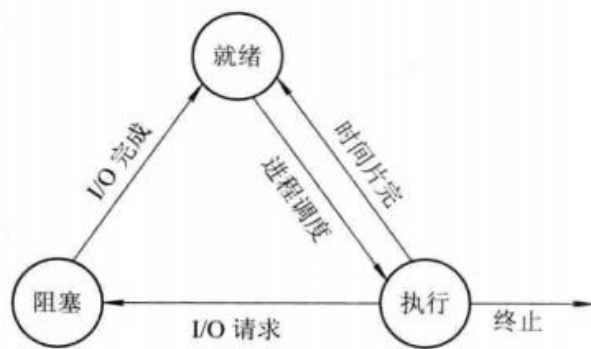


图 2-5 进程的三种基本状态及其转换

3. 进程控制（了解）

进程控制是进程管理中最基本的功能。进程的创建、撤销、唤醒，阻塞都属于进程控制。进程控制是通过原语来控制。

4. 进程同步

(一) 临界资源

一次只能被一个进程使用的叫临界资源。

(二) 临界区

每个进程中访问临界资源的那段代码称为临界区。

(三) 进程同步机制应遵循的规则

- 1) 空闲让进。当无进程处于临界区时，表明临界资源处于空闲状态，应允许一个请求进入临界区的进程立即进入自己的临界区，以有效地利用临界资源。
- 2) 忙则等待。当有进程进入临界区时，表明临界资源正在被访问，因而其它试图进入临界区的进程必须等待，以保证对临界资源的互斥访问。
- 3) 有限等待。对要求访问临界资源的进程，应保证在有限时间内能进入自己的临界区，以免陷入“死等”状态。
- 4) 让权等待。当进程不能进入自己的临界区时，应立即释放处理机，以免进程陷入“忙等”状态。

(四) 信号量机制（大题,pv 操作）

(五) 信号量的应用

5. 经典进程的同步问题（看书 P65）

生产者—消费者问题

读者—写者问题

第三章

1. 处理机调度的层次

- (1) 高级调度又称长程调度或作业调度。作用决定谁从外存进入内存
- (2) 低级调度又称进程调度或短程调度。作用决定谁获得处理机。
- (3) 中级调度又称内存调度。作用把那些暂时不能运行的进程，调至外存等待。把外存上那些已具备运行条件的就绪进程再重新调入内存，并修改其状态为就绪状态。是一个换进换出的过程。

2. CPU 的利用率

$$\text{CPU 的利用率} = \frac{\text{CPU 有效工作时间}}{\text{CPU 有效工作时间} + \text{CPU 空闲等待时间}}$$

3. 周转时间、带权周转时间、平均带权周转时间

周转时间：作业从提交到完成的时间=结束时间（完成时间）-到达时间=等待时间+要求服务时间

带权周转时间=周转时间/服务时间

平均带权周转时间=（周转时间/服务时间）/进程个数

4. 作业调度算法

先来先服务（对短作业不利）和短作业优先（对长作业不利）调度算法

5. 死锁的定义（举例子）

如果一组进程中的每一个进程都在等待仅由该组进程中的其他进程才能引发的事件，那么该组进程就是死锁的。

6. 死锁产生的原因：

竞争不可抢占性资源引发死锁

竞争可消耗资源引起死锁

进程推进顺序不当引起死锁

7. 产生死锁的必要条件

- (1) 互斥条件（固有条件，没办法破坏）：进程互斥使用临界资源
- (2) 请求和保持条件：进程在申请新资源的同时，保持对某些资源的占有
- (3) 不可抢占条件：资源只能由占有它的进程释放，不能被其它进程剥夺
- (4) 循环等待条件：存在循环等待链，在链中每个进程都在等待它的前一进程所持有的资源。

8. 处理死锁的办法（展开答）

(1) 预防死锁：通过设置某些限制条件,去破坏产生死锁的四个必要条件中的一个或几个来预防产生死锁。预防死锁是一种易实现的方法,已被广泛使用。

(2) 避免死锁：同样是属于事先预防策略,但它并不是事先采取各种限制措施,去破坏产生死锁的四个必要条件,而是在资源的动态分配过程中,用某种方法防止系统进入不安全状态,从而可以避免发生死锁★

(3) 检测死锁（了解）：无须事先采取限制性措施,允许进程在运行过程中发生死锁。但可通过检测机构及时地检测出死锁的发生,然后采取适当的措施,把进程从死锁中解脱出来。

(4) 解除死锁（了解）：当检测到系统中已发生死锁时,就采取相应的措施,把进程从死锁中解脱出来。常用的方法是撤消一些进程,回收它们的资源,将资源分配给已处于阻塞状态的进程,使其能继续运行。

9. 预防死锁的办法

（预防死锁的方法就是通过破坏产生的四个必要条件中的一个或几个。由于互斥条件是非共享设备所必须的，不仅不可以改变还应加以保证，所以主要是预防后三个条件。）

破坏“请求和保持”条件—预先静态分配（部分分配）

破坏“不可抢占”条件—（不剥夺）

破坏“循环等待”条件—有序资源分配（环路）

10. 利用银行家算法避免死锁（大题，work 表必画）

第四章 存储器管理

1. 基于顺序搜索的动态分区分配算法(应该不出大题)

(1) 首次适应算法（按照地址空间从小到大排序）

(2) 循环首次适应算法（从小到大排序，但是从上次已分配到的下一个开始寻找）

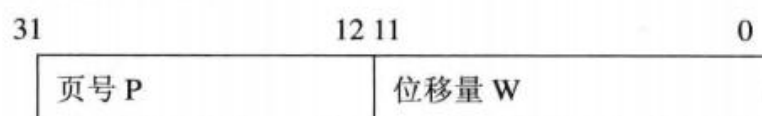
(3) 最佳适应算法（按照空闲区从小到大排序）

(4) 最坏适应算法（按照空闲区从大到小排序）

2. 分页存储管理的基本方法

(1) 地址结构

分页地址中的地址结构如下：



(2) 块表的作用

提高地址变换速度

(3) 具有块表的地址变换机构

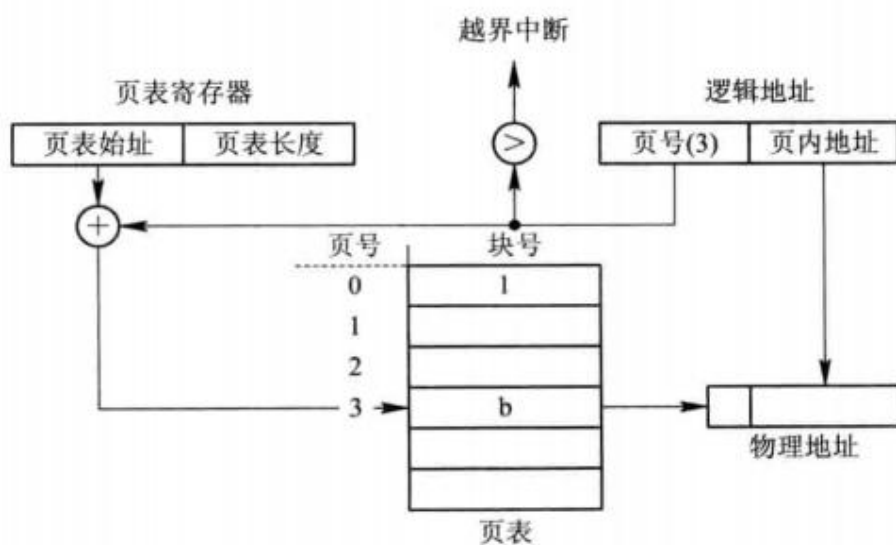


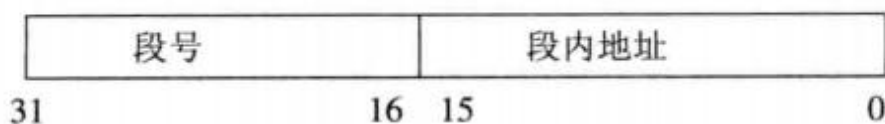
图 4-15 分页系统的地址变换机构

3. 分段存储管理方式

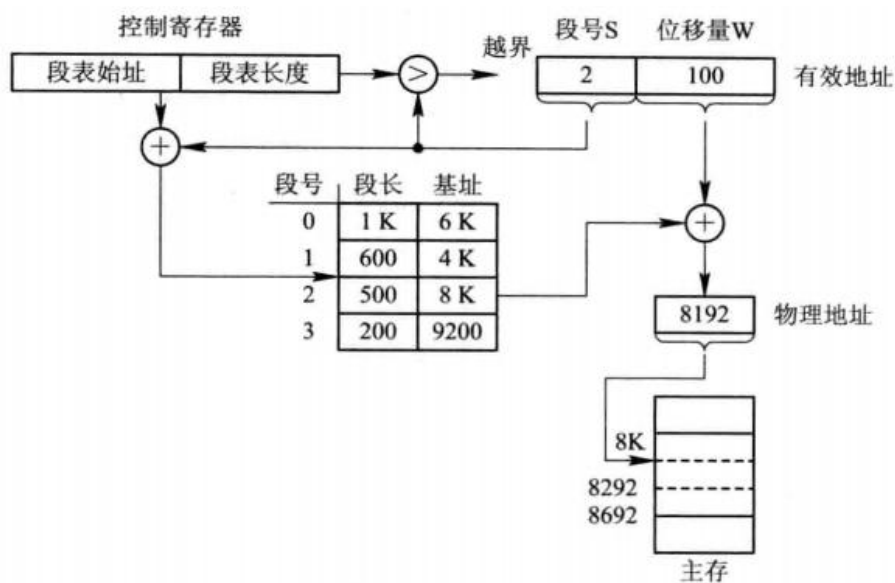
(1) 基本思路：段外离散，段内连续。

(2) 地址结构

分段地址中的地址具有如下结构：



(3) 分段系统的地址变换过程



4. 分段分页的主要区别

(1) 页是信息的物理单位。采用分页存储管理方式是实现离散分配方式，以消减内存的外零头，提高内存的利用率。或者说，分页仅仅只是系统管理上的需要，完全是系统的行为，对用户是不可见的。分段存储管理方式中的段则是信息的逻辑单位，它通常包含的是一组意义相对完整的信息。分段的目的主要在于能更好地满足用户的需要。

(2) 页的大小固定且由系统决定。在采用分页存储管理方式的系统中，在硬件结构上，就把用户程序的逻辑地址划分为页号和页内地址两部分，也就是说是由硬件实现的，因而在每个系统中只能有一种大小的页面。而段的长度却不固定，决定于用户所编写的程序，通常由编译程序在对源程序进行编译时，根据信息的性质来划分。

(3) 分页的用户程序地址空间是一维的。分页完全是系统的行为，故在分页系统中，用户程序的地址是属于单一的线性地址空间，程序员只需利用一个记忆符即可表示一个地址。而分段是用户的行为，故在分段系统中，用户程序的地址空间是二维的，程序员在标识一个地址时，既需给出段名，又需给出段内地址。

5. 段页式系统的基本原理

(1) 地址结构

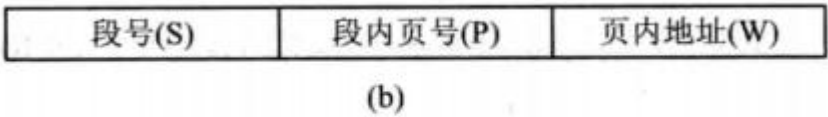


图 4-23 作业地址空间和地址结构

(2) 访问 3 次主存

6. 段页式存储管理方式提出的目的

便于管理，符合用户需求。

第五章 虚拟存储器

1. 虚拟存储器的概念

指具有请求调入功能和置换功能，能从逻辑上对内存容量加以扩充的一种存储器系统。

2. 虚拟存储器的实现方法

请求分页系统、请求分段系统、请求段页式系统

3. 请求分页中的地址变换过程

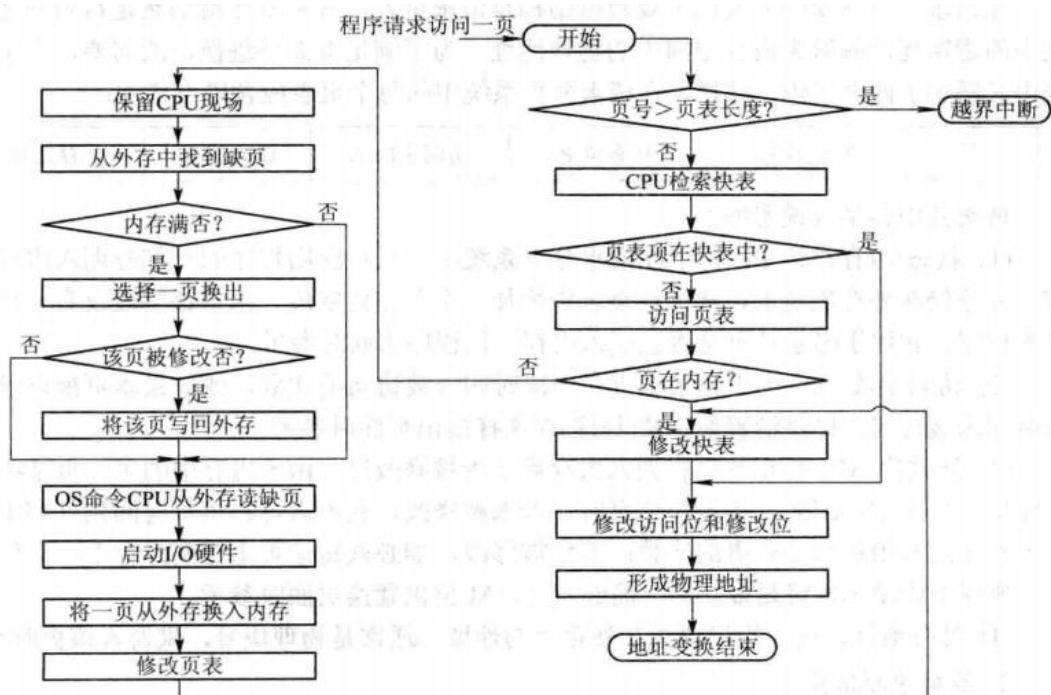


图 5-2 请求分页中的地址变换过程

4. 页面置换算法（大题）

最佳置换算法、先进先出置换算法、最近最久未使用置换算法、最少使用置换算法

5. 抖动的概念

所谓抖动现象，指如果分配给进程的存储块数量小于进程所需要的最小值，进程的运行将很频繁地产生缺页中断，这种频率非常高的页面置换现象称为抖动。

6. 产生“抖动”的原因

页面在内存和辅存间频繁交换

第六章

1. I/O 软件的层次结构及每层作用 ★

(1) 用户层 I/O 软件，实现与用户交互的接口，用户可直接调用该层所提供的、与 I/O 操作有关的库函数对设备进行操作。

(2) 设备独立性软件，用于实现用户程序与设备驱动器的统一接口、设备命名、设备的保护以及设备的分配与释放等，同时为设备管理和数据传送提供必要的存储空间。

(3) 设备驱动程序，与硬件直接相关，用于具体实现系统对设备发出的操作指令，驱动 I/O 设备工作的驱动程序。

(4) 中断处理程序，用于保存被中断进程的 CPU 环境，转入相应的中断处理程序进行处理，处理完毕再恢复被中断进程的现场后，返回到被中断的进程。

(箭头表示 I/O 的控制流)

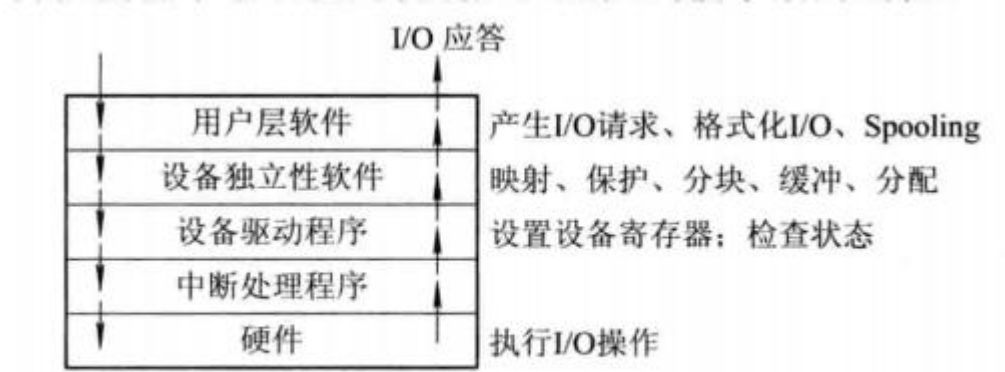


图 6-1 I/O 系统的层次结构

2. 设备控制器与 CPU 的关系

设备控制器是 CPU 与 I/O 设备之间的接口，接收从 CPU 发来的命令，去控制 I/O 设备工作，使处理机能够从繁杂的设备控制事务中解脱出来。

3. I/O 通道设备引入的目的

为了建立独立的 I/O 操作，不仅是数据的传送能独立于 CPU，而且也希望有关对 I/O 操作的组织、管理及其结束处理尽量独立，以保证 CPU 有更多的时间去进行数据处理；或者说，其目的是使一些原来由 CPU 处理的 I/O 任务转由通道来承担，从而把 CPU 从繁杂的 I/O 任务中解脱出来。

4. 对 I/O 设备的控制方式（通道对其干预最少）★

使用轮询的可编程 I/O 方式

使用中断的可编程 I/O 方式

直接存储器访问方式

5. 如何实现设备独立性

为了实现设备的独立性，应引入逻辑设备和物理设备两个概念。在应用程序中，使用逻辑设备名称来请求使用某类设备；而系统执行时，是使用物理设备名称。鉴于驱动程序是一个与硬件（或设备）紧密相关的软件，必须在驱动程序之上设置一层软件，称为设备独立性软件，以执行所有设备的公有操作、完成逻辑设备名到物理设备名的转换（为此应设置一张逻辑设备表）并向用户层（或文件层）软件提供统一接口，从而实现设备的独立性。

6.引入缓冲的目的

- (1) 缓和 CPU 与 I/O 设备间速度不匹配的矛盾
- (2) 减少对 CPU 的中断频率，放宽对 CPU 中断响应时间的限制
- (3) 解决数据粒度不匹配的问题

7. 引入 SPOOLing 系统目的

- (1) 提高 I/O 速度
- (2) 可以把独占设备改造为共享设备
- (3) 可以实现虚拟设备功能

8. 早期的磁盘调度算法（掌握）

先来先服务、最短寻道时间优先

9.基于扫描的磁盘调度算法（不知道考不考看看吧）

扫描算法、循环扫描算法、NStepSCAN 和 FSCAN 调度算法（?）

第七章 文件管理

1. 文件系统的目的

让用户以文件名来存取文件

2. 为何有文件目录

文件目录是一种数据结构，用于标识系统中的文件及其物理地址，供检索时使用。

3. 单级文件目录的目录项

目录项中含文件名、文件扩展名、文件长度、文件类型、文件物理地址以及其它的文件属性。

4. 两级文件目录的目的

- (1) 提高了检索目录的速度
- (2) 在不同的目录中，可以使用相同的文件名
- (3) 不同目录还可以使用不同的文件名访问系统中的同一个共享文件。

第八章 磁盘存储器的管理

1. 连续组织方式的优缺点

优点：

- (1) 顺序访问容易。
- (2) 顺序访问速度快。

缺点：

- (1) 要求为一个文件分配连续的存储空间，严重地降低了外存空间的利用率。如果是定期地利用紧凑方法来消除碎片，则又需花费大量的机器时间。
- (2) 必须事先知道文件的长度。
- (3) 不能灵活地删除和插入记录。
- (4) 对于那些动态增长的文件，由于事先很难知道文件的最终大小，因而很难为其分配空间，而即使事先知道文件的最终大小，在采用预分配存储空间的方法时，也会使大量的存储空间长期空闲。

2. 链接组织方式的优缺点

优点：

- (1) 消除了磁盘的外部碎片，提高了外存的利用率。
- (2) 对插入、删除和修改记录都非常容易。
- (3) 能适应文件的动态增长，无需事先知道文件的大小。
(链接方式又可分为隐式链接和显式链接两种形式。)

缺点：

- (1) 不能支持高效的直接存取。
- (2) FAT 需占用较大的内存空间。

3. 索引组织方式的优缺点

优点：

- (1) 支持直接访问
- (2) 不会产生外部碎片

缺点：

对于小文件采用索引分配方式时，其索引块的利用率是极低的。

4. 空闲表法、空闲链表法、成组链接法（不知道考的什么）