计算机操作系统复习资料

1. 操作系统的定义

操作系统 (Operating System，简称 OS)是管理计算机系统的全部硬件资源包括软件资源及数据资源；控制程序运行；改善人机界面；为其它应用软件提供支持等，使计算机系统所有资源最大限度地发挥作用，为用户提供方便的、有效的、友善的服务界面。

操作系统通常是最靠近硬件的一层系统软件，它把硬件裸机改造成为功能完善的一台虚拟机，使得计算机系统的使用和管理更加方便，计算机资源的利用效率更高，上层的应用程序可以获得比硬件提供的功能更多的支持。

操作系统是一个庞大的管理控制程序，大致包括 5 个方面的管理功能 ：进程与处理机管理、作业管理、存储管理、设备管理、文件管理。

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. 操作系统的基本特征

1）并发

2）共享

3）虚拟

4）异步

4. 分时系统的概念

把计算机的系统资源（尤其是 CPU时间）进行时间上的分割，每个时间段称为一个时间片，每个用户依次轮流使用时间片，实现多个用户分享同一台主机的操作系统。

5. 分时系统要解决的关键问题（ 2 个）

1）及时接收

2）及时处理

6. 并发性的概念

并发性是指两个或多个事件在同一事件间隔内发生。 在多道程序环境下，并发性是指在一段时间内宏观上有多个程序在同时运行，但在单处理机系统中，每一时刻却仅能有一道程序执行，故微观上这些程序只能是分时的交替执行。

7. 程序顺序执行的特征和并发执行的特征

顺序执行的特点 ： 顺序性 封闭性 可再现性

程序并发执行的特点：

1)、间断性（失去程序的封闭性）
2)、不可再现性
任何并发执行都是不可再现
3)、进程互斥（程序并发执行可以相互制约）

8. 进程的定义

进程是指在系统中能独立运行并作为资源分配的基本单位。

为了使参与并发执行的每个程序（含数据）都能独立的运行，在操作系统中必须为之配置一个专门的数据结构，称为进程控制块（ PCB）。系统利用PCB来描述进程的基本情况和活动过程，进而控制和管理进程。

9. 进程的组成部分

进程是由一组机器指令， 数据和堆栈组成的， 是一个能独立运行的活动实体。

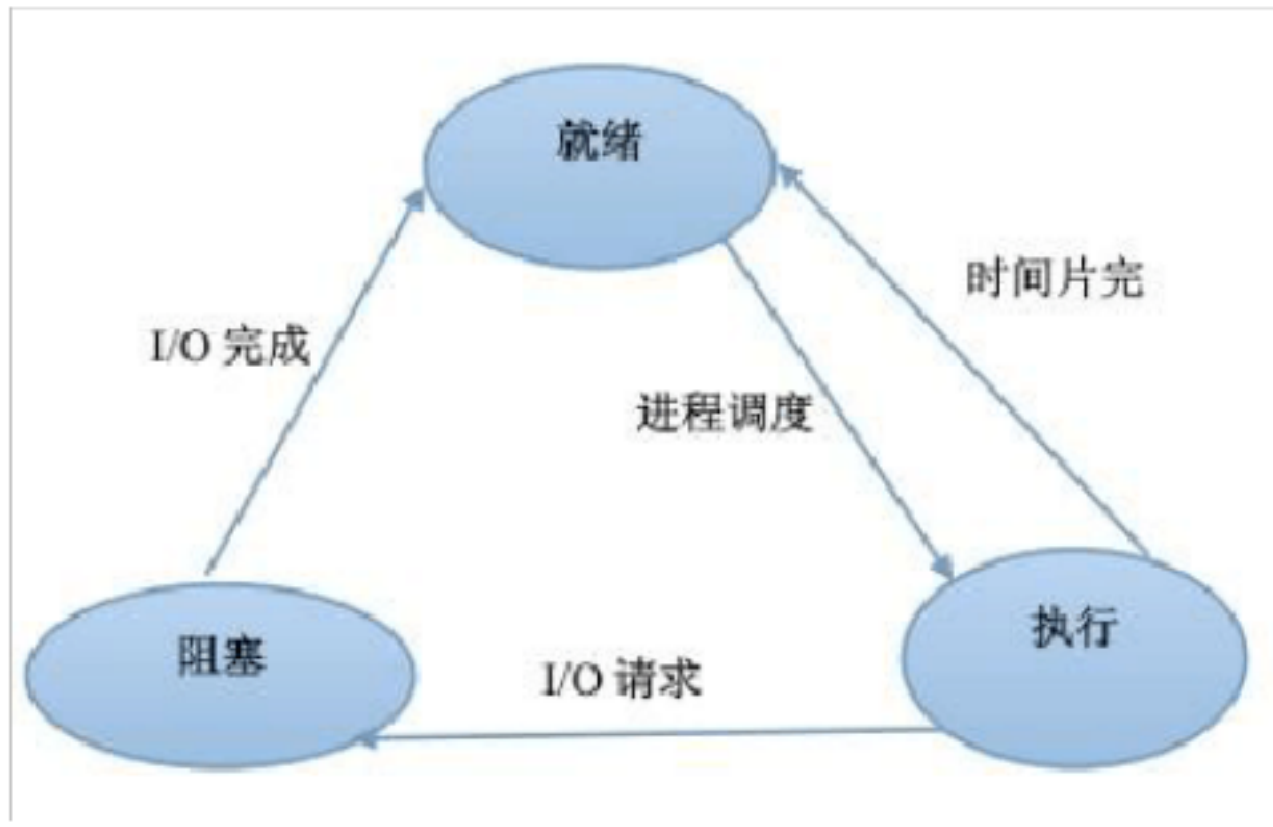
由程序段，相关的数据段和 PCB三部分便构成了进程实体（又称进程映像）。

10. 进程的状态（状态之间的变化）

就绪状态、执行状态、阻塞状态。

处于就绪状态的进程，在调度程序为之分配了处理机之后，该进程便可以执行，相应的，他就由就绪状态转变为执行状态。

正在执行的进程， 如果因为分配给它的时间片已经用完而被暂停执行时，该进程便由执行状态又回到就绪状态；如果因为发生某事件而使进程的执行受阻（如进程请求访问临界资源， 而该资源正在被其它进程访问） ，使之无法继续执行，该进程将有执行状态转变为阻塞状态。处于阻塞状态的进程，在获得了资源后，转变为就绪状态。



11. 进程同步的概念

进程同步是是并发执行的诸进程之间能有效地相互合作，从而使程序的执行具有可再现性，简单的说来就是：多个相关进程在执行次序上的协调。

12. PV原语的作用

PV原语通过操作信号量来处理进程间的同步与互斥的问题。　　其核心就是一段不可分割不可中断的程序。

13. 处理死锁的四种方法（有何不同）

1）预防死锁。这是一种简单和直观的事先预防方法。　该方法是通过设置某些限制条件，去破坏产生死锁的四个必要条件　（互斥条件，请求和保持条件，不可抢占条件，　循环等待条件）　中的一个或几个来预防产生死锁。　预防死锁是一种较易实现的方法，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。这种方法无须事先采取任何限制性措施，　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。当检测到系统中已发生死锁时，　就采取相应措施，　将进程从死锁状态中解脱出来。　常用的方法是撤销一些进程，　回收它们的资源，　将它们分配给已处于阻塞状态的进程，使其能继续运行。

上述的四种方法，从　1）到 4）对死锁的防范程度逐渐减弱，但对应的是资源利用率的提高，　以及进程因资源因素而阻塞的频度下降　（即并发程度提高）。

14. 解除死锁的方法

常采用解除死锁的两种方法是：

1）抢占资源。从一个或多个进程中抢占足够数量的资源，分配给死锁进程，以解除死锁状态。

2）终止（或撤销）进程。终止（或撤销）系统中的一个或多个死锁进程，直至打破循环环路，使系统从死锁状态解脱出来。

15. 死锁产生的必要条件

1）互斥条件

2）请求和保持条件

3）不可抢占条件

4）循环等待条件

16. 死锁的概念

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它进程才能引发的事件，那么该组进程是死锁的。

17. 银行家算法

银行家算法是一种最有代表性的避免死锁的算法。

要解释银行家算法，必须先解释操作系统安全状态和不安全状态。

安全状态：如果存在一个由系统中所有进程构成的安全序列　　P1，，，　Pn，则系统处于安全状态。安全状态一定是没有死锁发生。

不安全状态：不存在一个安全序列。不安全状态不一定导致死锁。

安全序列

一个进程序列 {P1，，， Pn}是安全的，如果对于每一个进程 Pi(1 ≤i ≤n)，它以后尚需要的资源量不超过系统当前剩余资源量与所有进程 Pj (j < i) 当前占有资源量之和。

18. 进程调度的功能

1) 记录系统中所有进程的执行情况

作为进程调度的准备，进程管理模块必须将系统中各进程的执行情况和状态特征记录在各进程的 PCB 表中。并且，根据各进程的状态特征和资源需求等、 进程管理模块还将各进程的 PCB 表排成相应的队列并进行动态队列转接。进程调度模块通过 PCB 变化来掌握系统中存在的所有进程的执行情况和状态特征， 并在适当的时机从就绪队列中选择出一个进程占据处理机。

2) 选择占有处理机的进程

进程调度的主要功能是按照一定的策略选择 一个处于就绪状态的进程，使其获得处理机执行。根据不同的系统设计目的，有各种各样的选择策略，例如 系统开销较少的静态优先数调度法，适合于分时系统的轮转法 (Round RoLin) 和多级互馈轮转法 (Round Robin with Multip1e feedback) 等。这些选择策略决定了调度算法的 性能。

3) 进行进程上下文切换

一个进程的上下文 (context) 包括进程的状态、有关变量和数据结构的值、机器寄存器的值和 PCB 以及有关程序、数据等。一个进程的执行是在进程的上下文中执行。当正在执行的进程由于某种原因要让出处理机时，系统要做进程上下文切换，以使另一个进程得以执行。当进行上下文切换时点统要首先检查是否允许做上下文切换(在有些情况下，上下文切换是不允许的，例如系统正在执行某个不允许中断的原语时)。然后，系统要保留有关被切换进程的足够信息，以便以后切换回该进程时，顺利恢复该进程的执行。 在系统保留了 CPU 现场之后， 调度程序选择一个新的处于就绪状态的进程、并装配该进程的上下文，使 CPU 的控制权掌握在被选中进程手中。

19. 作业调度算法（ FCFS和短作业优先）

先来先服务（ FCFS）调度算法

FCFS是最简单的调度算法，该算法既可用于作业调度，也可用于进程调度。当在作业调度中采用该算法时，系统将按照作业到达的先后次序来进行调度，或者说它是优先考虑在系统中等待时间最长的作业，而不管该作业所需执行时间的长短从后备作业队列中选择几个最先进入该队列的作业，将它们调入内存，为它们分配资源和创建进程。然后把它们放入就绪队列。

当在进程调度中采用 FCFS算法时，每次调度是从就绪的进程队列中选择一个最先进入该队列的进程，为之分配处理机，使之投入运行。该进程一直运行到完成或发生某事件而阻塞后，进程调度程序才将处理机分配给其它进程。

优点：

1）简单可靠

2）容易理解，实现方便

3）非抢占式的

缺点：

1）有利于长的作业和进程，不利于短的

2）有利于 CPU繁忙型的作业或进程，不利于 I/O 繁忙型的

短作业优先（ SJF）的调度算法

SJF算法是以作业的长短来计算优先级，作业越短，其优先级越高。作业的长短是以作业所要求的运行时间来衡量的。 SJF 算法可以分别用于作业调度和进程调度。班在短作业优先调度算法用于作业调度时， 它将从外存的作业后备队列中选择若干个估计运行时间最短的作业，优先将它们调入内存运行。

缺点：

1）必须预知作业的运行时间

2）对长作业非常不利，长作业的周转时间会明显地增长

3）在采用 SJF 算法时，人 - 机无法实现交互

4）该调度算法完全未考虑作业的紧迫程度，故不能保证紧迫性作业能得到及时处理

20. 存储管理的对象

存储管理的对象是主存储器 (简称内存或主存) 。

分区管理的分配算法

存储管理基本技术

三种基本的存储管理技术：分区法、可重定位分区法和对换技术

1. 分区法：把内存划分成若干分区，每个分区里容纳一个作业。

1) 固定分区：分区的个数、分区的大小固定不变； 每个分区只能放一道作业。

优点：管理方式简单。

缺点：内存空间利用率低。

2) 动态分区法：分区大小和个数依作业情况而定；作业进入内存时才建分区。

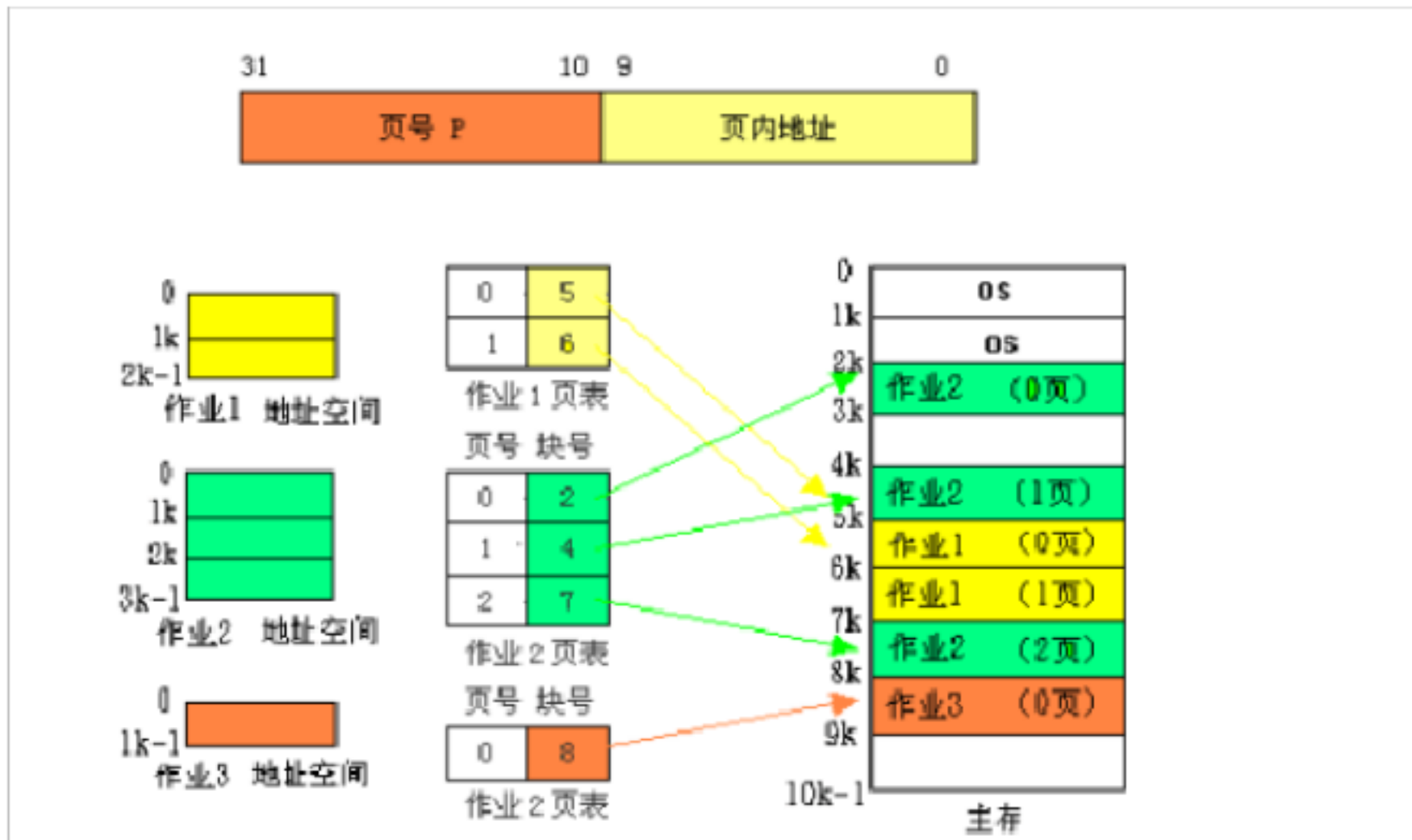
优点：按需分配内存

缺点：产生大量碎片。

2. 可重定位分区分配：通过紧缩可解决碎片问题；作业在内存中可以移动。

优点：解决了碎片的问题，提高了主存利用率；

缺点：增加了开销。，但须消耗大量的 CPU 时间。



3. 对换技术：作业（或进程）在内存和磁盘之间交换，换出暂时不能运行的作业（或进程）；换入具备运行条件的作业（或进程）。

基于顺序搜索的的动态分配算法：

1 首次适应算法　　2　　循环首次适应算法　　3　最佳适应算法　　4　最坏适应算法

基于索引搜索的动态分配算法

1 快速适应算法　　2　　伙伴系统　　3　哈希算法

21. 分页存储管理的基本思想和页表的作用

将用户程序的地址空间分为若干个固定大小的区域，称为“ 页 ”或“ 页面 ”　　。典型的页面大小为　1KB。相应的，也将内存空间分为若干个物理块或页框，页和块的大小相同。这样可将用户的程序的任一页放入任一物理块中，实现了离散分配。

页表的作用：能在内存中找到每一个页面所对应的物理块。

22. 段式存储的基本思想

把用户程序的地址空间分为若干个大小不同的段，每段可定义一组相对完整的信息。在存储器分配时，以段为单位，这些段在内存中可以不相邻接，所以也同样实现了离散分配。

23. 分页与分段的区别

分页和分段系统有许多相似之处，但在概念上两者完全不同，主要表现在：　　1、页是信息的物理单位，分页是为实现离散分配方式，以消减内存的外零头，提高内存的利用率；或者说，分页仅仅是由于系统管理的需要，而不是用户的需要。段是信息的逻辑单位，它含有一组其意义相对完整的信息。　　分段的目的是为了能更好的满足用户的需要。

2、页的大小固定且由系统确定，把　　逻辑地址　划分为页号和页内地址两部分，是由机器

硬件实现的，因而一个系统只能有一种大小的页面。段的长度却不固定，决定于用户所编写的程序，通常由　编辑程序 在对源程序进行编辑时，根据信息的性质来划分。　　3、分页的作业 地址空间 是维一的，即单一的　线性空间 ，程序员 只须利用一个记忆符，即可表示一地址。分段的作业地址空间是二维的，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。

24. 页面置换的算法，　FIFO,LRU

FIFO：

1. 先进先出法（FIFO）：将最先进入内存的页换出内存。

例如 内存块数量为 3 时，采用 FIFO 页面置换算法，下面页面走向情况下，缺页次数是多少？

9

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0
7	7	2	2		2	2	4	4	4	0			0	0	
	0	0	0		3	3	3	2	2	2			1	1	
		1	1		1	0	0	0	3	3			3	2	

∴ 缺页次数=15 次

LRU:

3. 最近最少使用置换法（LRU）：将最近一段时间里最久没有使用过的页面换出内存。

例如 内存块数量为 3 时，采用 LRU 页面置换算法，下面页面走向情况下，缺页次数是多少？

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	2	2		2		4	4	4	0			1		1		1		
	0	0	0		0		0	0	3	3			3		0		0		
		1	1		3		3	2	2	2			2		2		7		

∴ 缺页次数=12 次

25. 输入输出四种控制方式

1）程序直接控制方式

2）中断驱动方式

3）DMA(直接存储器存取）方式

4）通道控制方式

26. 输入输出 I/O 重定向的概念

I/O 重定向是一个过程，这个过程捕捉一个文件、或命令、或程序、或脚本、甚至代码块（ code block ）的输出，然后把捕捉到的输出，作为输入发送给另外一个文件、或命令、或程序、或脚本。 I/O 重定向最常用的方法是管道（管道符 "|" ）。

27. 设备独立性

设备独立性，即应用程序独立于具体使用的物理设备。为了实现设备独立性而引入了逻辑设备和物理设备这两个概念。在应用程序中，使用逻辑设备名称来请求使用某类设备 ;而系统在实际执行时， 还必须使用物理设备名称。 因此， 系统须具有将逻辑设备名称转换为某物理设备名称的功能， 这非常类似于存储器管理中所介绍的逻辑地址和物理地址的概念。

28. 中断的定义

中断是指 CPU对 I/O 发来的中断信号的一中响应。 CPU暂停正在执行的程序，保留 CPU环境后，自动地转去执行该 I/O 设备的中断处理程序。执行完后，在回到断点，继续执行原来的程序。 I/O 设备可以是字符设备，也可以是块设备， 通信设备等。由于中断是由外部设备引起的， 故又称为外中断。

29. 中断处理程序的处理过程

1）测定是否有未响应的中断信号

2）保护被中断进程的 CPU环境

3）转入相应的设备处理程序

4）中断处理

5）恢复 CPU的现场并退出中断

30. 假脱机（ Spooling ）系统的作用

1) 提高了 I/O 速度. 从对低速 I/O 设备进行的 I/O 操作变为对输入井或输出井的操作 , 如同脱机操作一样 , 提高了 I/O 速度 , 缓和了 CPU与低速 I/O 设备速度不匹配的矛盾 .

2) 设备并没有分配给任何进程 . 在输入井或输出井中 , 分配给进程的是一存储区和建立一张 I/O 请求表 .

3) 实现了虚拟设备功能 . 多个进程同时使用一独享设备 , 而对每一进程而言 , 都认为自己独占这一设备 , 不过 , 该设备是逻辑上的设备 .

31. 磁盘调度的算法

早期的磁盘调度算法

1）先来先服务（ FCFS）

2）最短寻道时间优先（ SSTF）

基于扫描的磁盘调度算法

1）扫描（ SCAN）算法

2）循环扫描（ CSCAN）算法

3）NStepSCAN 和 FSCAN 调度算法

32. 文件逻辑结构，物理结构的定义

1）文件的逻辑结构。 这是从用户观点出发所观察到的文件组织形式， 即文件是由一系列的逻辑记录组成的， 是用户可以直接处理的数据及其结构， 它独立于文件的物理特性，又称为文件组织。

2）文件的物理结构， 又称为文件的存储结构。 这是指系统将文件存储在外存上所形成的一种存储组织形式， 是用户不能看见的。 文件的物理结构不仅与存储介质的存储性能有关， 而且与所采用的外存分配方式有关。 无论是文件的逻辑结构，还是其物理结构，都会影响对文件的检索速度。

33. 目录文件存放的信息

该目录中所有子目录文件和数据文件的目录

34. 文件多级目录结构的特点

多级（树形）目录结构的优点是便于文件分类，可为每类文件建立一个子目录；查找速度快，因为每个目录下的文件数目较少；可以实现 文件共享 ；缺点是比较复杂。

35. 保护域的定义

为了对系统中的资源进行保护而引入了保护域的概念， 保护域简称“ 域 ”。“ 域 ”是进程对一组对象访问权的集合，进程只能在制定域内执行操作。这样，“ 域 ”也就规定了进程所能访问的对象和能执行的操作。

36. 访问权的定义

为了对系统中的对象加以保护，应由系统来控制进程对对象的访问。对象可以是硬件对象，如磁盘驱动器，打印机；也可以是软件对象，如文件，程序。对对象所施加的操作也有所不同，如对文件可以是读，也可以是写或执行操作。我们把一个进程能对某对象执行操作的权力，称为访问权。每个访问权可以用一个有序对（对象名，权集）来表示。例如，某进程有对文件 F1 执行读和写操作的权力，则可将该进程的访问权表示成（ F1, {R/W}）。

37. 位示图进行盘块，回收的过程

位示图是利用二进制的一位来表示磁盘中的一个盘块的使用情况。当其值为 "0" 时，表示对应的盘块空闲 ；为 "1" 时，表示已经分配。有的系统把 "0" 作为盘块已分配的标记，把 "1" 作为空闲标志。 (它们的本质上是相同的，都是用一位的两种状态标志空闲和已分配两种情况。) 磁盘上的所有盘块都有一个二进制位与之对应，这样，由所有盘块所对应的位构成一个集合，称为位

示图。通常可用 m*n 个位数来构成位示图，并使 m*n 等于磁盘的总块数。位示图也可描述为一个二维数组 map[m,n] 。

盘块的分配

根据位示图进行盘块分配时，可分三步进行：

1） 顺序扫描位示图，从中找出一个或一组其值为“ 0 ”的二进制位（“ 0 ”表示空闲时）。

2） 将所找到的一个或一组二进制位转换为与之对应的盘块号。假定找到的其值为“ 0 ”的二进制位位于位示图的第 i 行，第 j 列，则其相应的盘块号应按下式计算： b=n(i-1)+j ，式中， n 代表每行的位数。

3） 修改位示图，令 map[I,j]=1 。

盘块的回收

1） 将回收盘块的盘块号转换为位示图中的行号和列号。转换公式为：

I=(b-1)DIV n+1

J=(b-1)MOD n+1

2） 修改位示图。令 map[I,j]=0. 。

38. CPU指令的分类

常见的特权指令有以下几种 ：

（1）有关对 I/O 设备使用的指令 如启动 I/O 设备指令、测试 I/O 设备工作状态和控制 I/O 设备动作的指令等。

（2）有关访问程序状态的指令 如对程序状态字（ PSW 的指令等。

（3）存取特殊寄存器指令 如存取中断寄存器、时钟寄存器等指令。

（4）其他指令

39.CPU运行状态的分类

40. 网络操作系统的互操作功能

为了了实现多个网络之间的通信和资源共享，不仅需要将他们从物理上连接在一起，而且还应使不同网络的计算机系统之间能进行通信（信息互通）和实现资源共享（信息互用） 。为此网络 OS中必须提供应用互操作功能，以实现“ 信息互通性 ”及“ 信息互用性 ” 。

（1） 信息互通性。为了避免在不同网络中，因采用了不同的协议而不能识别和通信，在互连网络的每一个网络中都应配置同一类型的传输协议，以实现各个网络之间的通信。

（2） 信息的互用性。所谓信息的互用性，是指在不同的网络中的站点之间能实现信息的互用，亦即一个网络中的用户能够访问另一个网络文件系统（或数据库系统）中的文件（数据） 。不能实现信息的互用性的原因是在不同网络中所配置的网络文件系统 （或数据库系统），通常使用了各不相同的结构、各不相同的文件命名方式和存取文件的命令，于是便发生了有一个源网络中的用户发往一个目标网络的文件访问命令不能被目标网络的节点所识别的情况。对此，一个当前相对比较流行的解决方案是由 SUN公司推出的网络文件系统协议 NFS

41. 网络协议的三要素

1）语义

2）语法

3）时序

42. 加密算法的分类

对称加密算法

非对称加密算法

43. 身份认证的依据

当前身份验证主要依据下述三个方面的信息来确认：

1) 所知，即基于用户所知道的信息，如系统的登录名，口令等。
2) 所有，指用户所具有的东西，如身份证，信用卡等。
3) 用户特征，指用户所具有的特征，特别是生理特征，如指纹，声纹， DN 等。

1. 某杂技团进行走钢丝表演。在钢丝的 A、B 两端各有 n 名演员（n>1）在等待表演。只要钢丝上无人时便允许一名演员从钢丝的一端走到另一端。现要求两端的演员交替地走钢丝，且从 A 端的一名演员先开始。请问，把一名演员看作一个进程时，怎样用 WAIT,SIGNAL操作来进行控制？请写出能进行正确管理的程序。

```
begin S1,S2: semaphore;
    S1: =1;        (定义两个信号量)
    S2: =0;
    Cobegin    (正确给出信号量初值，)
    processAtoBi (3 分)
    (I=1, 2, …, n)
    begin
    P(S1);
    {表演};
    V(S2);
    end;
    coend
    processBtoAj (3 分)
    (j=1, 2, …, n)
    begin
    P(S2);
   {表演};
   V(S1);
   end;
```

2. 有一阅览室，读者进入时必须先在一张登记表中进行登记，该表为每一座位列一表目，包括座号和读者姓名，读者离开时要消掉登记信息，阅览室中共

有 100 个座位，试问：试用信号量和 wait，signal 原语写出这些进程间的同步算法。

```
begin
    S1:=100 (有 100 个座位)
    S2:=0 (有没阅读者)
    mutex: =1
cobegin
P1: repeat
    P(S1);
    P(mutex);
    登记信息;
    V(muetx);
    V(S2)就座，阅读;
    until false
P2: repeat
        P(S2)
        P(mutex);
        消掉信息;
        V(muetx);
        V(S1);
        离开阅览室;
        until false
coend
end
```

3. 请用信号量解决以下的“过独木桥”问题：同一方向的行人可连续过桥，当某一方向有人过桥时，另一方向的行人必须等待；当某一方向无人过桥时，另一方向的行人可以过桥。

解：将独木桥的两个方向分别标记为 A 和 B；并用整形变量 countA 和 countB 分别表示 A、B 方向上已在独木桥上的行人数，初值为 0；再设置三个初值都 1 的互斥信号量：SA 用来实现对 countA 的互斥访问，SB 用来实现对 countB 的互斥访问，mutex 用来实现两个方向的行人对独木桥的互斥使用。则具体描述如下：

```
Var SA,SB,mutex:semaphore:=1,1,1;
    CountA,countB:integer:=0,0:
    begin
    parbegin
     process A: begin
                  wait(SA);
                    if(countA=0) then wait(mutex);
                    countA:=countA+1;
                  signal(SA);
                  过独木桥;
                  wait(SA);
                    countA:=countA-1;
                    if (countA=0) then signal(mutex);
                  signa(SA);
                end
process B: begin
                  wait(SB);
                    if(countB=0) then wait(mutex);
                    countB:=countB+1;
                  signal(SB);
                  过独木桥;
                  wait(SB);
                    countB:=countB-1;
                    if (countB=0) then signal(mutex);
                  signa(SB);
                end
     parend
end
```

4. 假定系统有三个并发进程 read，move和 print 共享缓冲器 B1和 B2。进程 read 负责从输入设备上读信息， 每读出一个记录后把它存放到缓冲器 B1中。进程 move 从缓冲器 B1 中取出一记录，加工后存入缓冲器 B2。进程 print 将 B2 中的记录取出打印输出。 缓冲器 B1 和 B2 每次只能存放一个记录。 要求三个进程协调完成任务，使打印出来的与读入的记录的个数，次序完全一样。

请用 WAIT()和 SIGNAL()原语操作，写出它们的并发程序。

```
begin  SR,SM1,SM2,SP:semaphore;
       B1,B2:record;
       SR:=1;SM1:=0;SM2:=1;SP:=0
```

```
cobegin
process read
          X:record;
begin  R: (接收来自输入设备上一个记录)
             X:=接收的一个记录;
             WAIT(SR);
             B1:=X;
             SIGNAL(SM1);
             Goto   R;
end;
Process  move
   Y:record;
   begin
         M:WAIT(SM1);
          Y:=B1;
          SIGNAL(SR)
          加工 Y
          WAIT(SM2);
          B2:=Y;
          SIGNAL(SP);
          goto  M;
   end;
Process  print
   Z:record;
   begin
     P:WAIT(SP);
       Z:=B2;
       SIGNAL(SM2)
       打印 Z
       goto   P;
       end;
  coend;
end;
```

1、某虚拟存储器的用户编程空间共 32 个页面，每页为 1KB，内存为 16KB。假定某时刻一用户页表中已调入内存的页面的页号和物理块号的对照表如下：

页号	物理块号
0	5
1	10
2	4
3	7

则逻辑地址 0A5C（H）所对应的物理地址是什么？

解：程序空间的大小为 32KB，因此逻辑地址的有效位数是 15 位。内存储空间的大小是 16KB，因此物理地址至少需要 14 位。

当页面为 1KB时，虚地址 0A5C表示页号为 00010，页内地址是 1001011100. 该页在内存的第 4 块，即块号为 0100，因此 0A5C的物理地址是 01001001011100，即 123CH。

2、某段表内容如下：

段号　　段首地址　　段长度
0　120K　40K
1　760K　30K
2　480K　20K
3　370K　20K

一逻辑地址为（ 2，154）的实际物理地址为多少？

答：逻辑地址（ 2,154 ）表示段号为 2，即段手地址为 480K，154 为单元号，则实际物理地址为 480K+154。

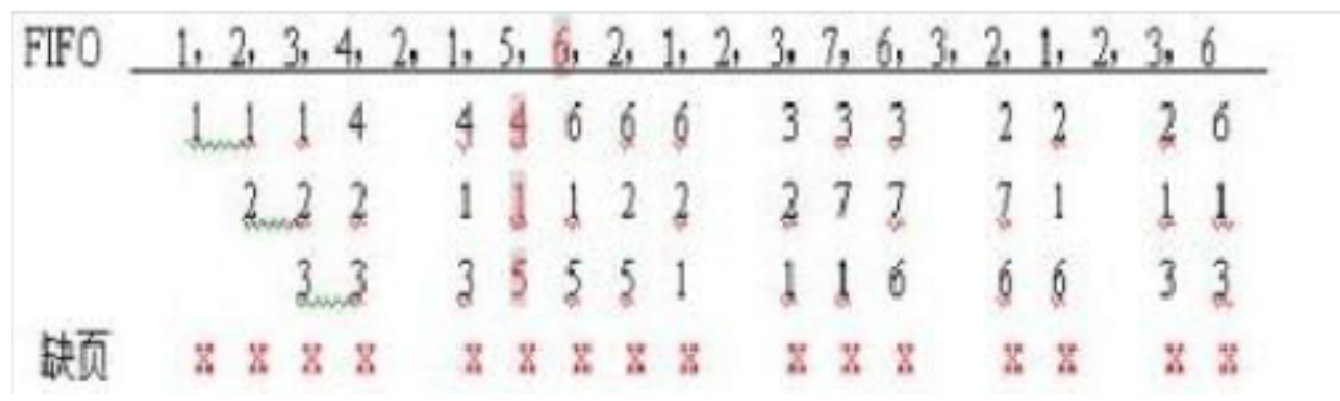
3、考虑下述页面走向：

1，2，3，4，2，1，5，6，2，1，2，3，7，6，3，2，1，2，3，6

当内存块数量分别为 3 时，试问 FIFO、LRU、OPT这三种置换算法的缺页次数各是多少？

答：缺页定义为所有内存块最初都是空的，　所以第一次用到的页面都产生一次缺页。

当内存块数量为 3 时：

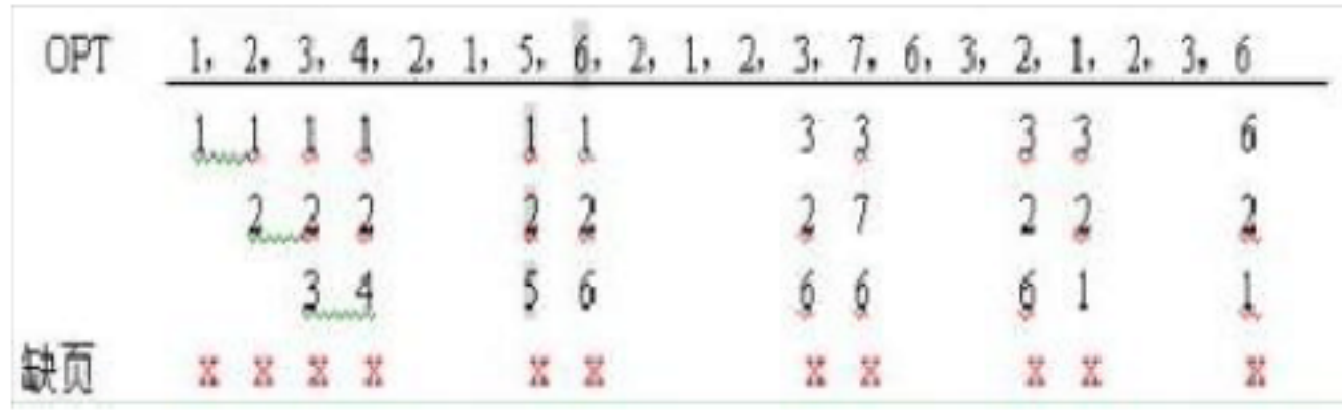


发生缺页中断的次数为 16。

在 FIFO 算法中，先进入内存的页面被先换出。当页　6 要调入时，内存的状态为 4、1、5，考查页 6 之前调入的页面，分别为 5、1、2、4，可见 4 为最先进入内存的，本次应换出，然后把页　6 调入内存。

发生缺页中断的次数为 15。

在 LRU算法中，最近最少使用的页面被先换出。　当页 6 要调入时，内存的状态为 5、2、1，考查页 6 之前调入的页面，分别为 5、1、2，可见 2 为最近一段时间内使用最少的，本次应换出，然后把页　6 调入内存。



发生缺页中断的次数为 11。

在 OPT算法中，在最远的将来才被访问的页面被先换出。当页 6 要调入时，内存的状态为 1、2、5，考查页 6 后面要调入的页面，分别为 2、1、2、，，可见 5 为最近一段时间内使用最少的，本次应换出，然后把页 6 调入内存。

OPT算法因为要知道后面请求的页框， 因此我觉得这个算法有个小小的 bug，如果在某个请求中， 若在该请求的页框之后的页框序列中至少存在一个和当前内存块中不匹配的页框，则按照内存块的顺序（从上往下）替换没有出现的页框。比如上面那个 OPT例子。对于最后一个页框请求， 因为 6 未命中， 且 6 之后没有请求的序列，因此应该替换 3，所以替换后的序列为 6 ， 2 ， 1 。

4. . 假设一个磁盘有 200 个磁道，编号从 O～199。当前磁头正在 143 道上服务，并且刚刚完成了 125 道的请求。如果寻道请求队列的顺序是： 86，147，9l，177， 94， 150， 102， 175， 130 问：为完成上述请求，使用最短寻道时间优先磁盘调度算法 SSTF时，磁头移动的总量是多少 ?(要求写出分析过程)

（1）采用先来先服务磁盘调度算法 FCFS，进行调度的情况为：从 143 道开始

下一磁道	移动磁道数
86	57
147	61
91	56
177	86
94	83
150	56
102	48
175	73
130	45

磁头移动总量为 565。

（2）采用最短寻道时间优先磁盘调度算法 SSTF，进行调度的情况为：从 143 道开始

下一磁道	移动磁道数
147	4
150	3
130	20
102	28
94	8
91	3
86	5
175	89
177	2

磁头移动总量为 162。

(147-143)+(150-147)+(150-130)+(130-102)+(102-94)+(94-91)+(91-86)=162

5. 银行家算法 (操作系统）

在银行家算法中，某 T0 时刻的资源分配情况如下：（有三类资源 A、B、C，五个进程 P0、P1、P2、P3、P4）

Max Allocation Need Available

A B C A B C A B C A B C

P0 7 5 3 0 1 0 7 4 3 3 3 2

P1 3 2 2 2 0 0 1 2 2

P2 9 0 2 3 0 2 6 0 0

P3 2 2 2 2 1 1 0 1 1
P4 4 3 3 0 0 2 4 3 1
试问：
1. 该状态是否安全？

2. 在 T0 时刻， P1发出请求 Request(1 ， 1， 2) ， 系统能否满足？为什么？
答案：
1、这是安全状态：
P1 的需求小于可用资源数，先满足 P1 的请求，然后回收 P1 资源：可用资源变为 （3,3,2 ）＋（ 2,0,0 ）＝（ 5,3,2 ）;
这时 P3 可分配，P3结束后回收资源， 可用资源为（5,3,2 ）＋(2,1,1 ）＝(7,4,3 ）
这时 P0 可分配，P0结束后回收资源， 可用资源为（7,4,3 ）＋(0,1,0 ）＋(7,5,3 ）
接下来是 P2，结束后可用资源为（ 7,5,3 ）＋（ 3,0,2 ）＝（ 10,5,5 ）
最后分配 P4，结束后可用资源为（ 10,5,5 ）＋（ 0,0,2 ）＝（ 10,5,7 ）
这样得到一个安全序列： P1—P3—P0—P2—P4，所以 T0 状态是安全的。

2、T0 时刻 P1 请求（ 1,1,2 ）<可用资源数（ 3,3,2 ），可以直接满足。