《操作系统》期末考试复习资料

●考试时间：1 月 7 日 下午 14:00 **●考试地点：**创新楼 103

●考试范围：《操作系统》第 1—4 章

●考试题型：主客观题各占 50%

◆选择题 20 分（每题 1 分） ◆填空题 20 分（每题 2 分）

◆判断题 10 分（每题 1 分） ◆名词解释 10 分（每题 2 分）

◆简答题 20 分（每题 5 分） ◆综合题 20 分（每题 10 分）

●复习要点：

◆选择题（参考《计算机操作系统》习题指导与题解选择题部分）

◆填空题（此部分为《计算机操作系统》习题指导与题解填空题题部分）

第一章：操作系统引论

1. 设计现代 OS 的主要目标：①和②
   ①充分利用资源 ②方便用户
2. 单道批处理系统是在解决①和②的矛盾中发展起来的。
   ①人机匹配 ②CPU 与 I/O 设备速度不匹配
3. 在单处理机环境下的多道程序设计具有①和②的特点。
   ①宏观上同时运行 ②微观上交替运行
4. 现在操作系统的两个最基本的特征是①和②，除此之外，它还具有③和④的特征。
   ①并发②资源共享 ③虚拟性 ④异步性
5. 从资源管理的角度看，操作系统具有四大功能 ：①，②，③，④，为了方便用户，操作系统还必须提供⑤。
   ①处理机管理 ②存储管理 ③文件管理 ④设备管理 ⑤友好的用户接口
6. 操作系统的基本类型主要有①和②和③。
   ①批处理系统 ②分时系统 ③实时系统
7. 批处理系统的主要优点是①和②，主要缺点③和④。
   ①资源利用率 ②系统的吞吐量 ③无交互作用能力 ④作业平均周转时间长
8. 实现分时系统的关键问题是①，为此必须引入②的概念，并采用③调度算法 。
   ①人机交互 ②时间片 ③时间片轮转
9. 分时系统的基本特征，①，②，③和④。
   ①多路性 ②独立性 ③交互性 ④及时性
10. 若干事件在同一时间间隔内发生称为①；若干事件在同一时刻发生称为②。
    ①并发 ②并行
11. 实时系统可分为①和②两种类型，民航售票系统属于③，而导弹飞行控制系统则属于④。
    ①实时信息 ②实时控制 ③实时信息 ④实时控制
12. 为了使实时系统高度可靠和安全，通常不强求①。

①资源利用率

13. 当前比较流行的微内核的操作系统结构，是建立在层次化结构的基础上的，而且还采用了①模式和②技术。

①客户机/服务器 ②面向对象程序设计。

第二章：进程管理

1. 在单用户任务环境下，用户独占全机，此时机内资源的状态，只能由运行程序的操作加以改变，此时的程序执行具有①性和②性特征。

①封闭 ②可再现性

2. 并发进程之间的相互制约，是由于他们的①和②而产生的，因而导致程序在并发执行时具有③特征。

①共享资源 ②相互合作 ③间断性或异步性

3. 程序并发执行与顺序执行时相比产生了一些新特征，分别是①，②和③。

①间断性 ②失去封闭性 ③不可再现性

4. 引入进程的目的是①，而引入线程的目的是②。

① 使程序能并发的执行，提高系统利用率和系统吞吐量；

② 减少并发执行的开销，提高程序执行的并发程度。

5. 进程由①，②和③组成，其中④是进程存在的唯一标志。

①PCB ②程序段 ③数据段 ④PCB

6. 进程最基本的特征是①和②，除此之外，它还有③，④和⑤特征。

①动态性 ②并发性 ③独立特性 ④异步性 ⑤结构

7. 由于进程的实质是程序的一次执行，故进程由①的基本特征，该特征还表现在进程由②而产生，由③而执行，由④而消亡，即进程具有一定的生命期。

①动态性 ②创建 ③调度 ④撤销

8. 引入进程带来的好处是①和②

①提高资源的利用率 ②增加系统吞吐量

9. 当前正在执行的进程由于时间片用完而暂时执行时，该进程应转变为①状态；若因发生某种事件而不能继续执行时，应转为②状态；若应终端用户的请求而暂停执行时，它应转为③状态。

①就绪 ②阻塞 ③静止就绪

10. 用户为阻止进程继续运行，应利用①原语，若进程正在执行，应转变为②状态，以后，若用户要恢复其运行，应利用③原语此时进程应转变为④状态。

①挂起 ②静止就绪 ③激活 ④活动就绪

11. 系统中共有 5 个用户进程，且当前 CPU 在用户态下执行，则最多可有①个用户进程处于就绪状态，最多可有②个用户进程处于阻塞状态，若当前在核心态下执行，则最多可有③个用户进程处于就绪状态，最多可有④个用户进程处于阻塞状态。

①4 ②4 ③5 ④5

12. 同步机制应遵循的准则有是①，②，③和④。

①空闲让进 ②忙则等待 ③有限等待 ④让权等待

13. 在记录型信号量机制中，S.value>0 时的值表示①；每次 wait 操作意味着②，因此应将 S.value③，当 S.value④时，进程应阻塞。

①可用临界资源数量 ②申请一个临界资源 ③减 1 ④小于 0

14. 在记录型信号量机制中，每次 signal 操作意味着①，因此应将 S.value②，当 S.value<=0 时，表示③，此时应④。

①释放一个临界资源 ②加 1 ③仍有请求该资源的进程被阻塞 ④唤醒相应阻塞队列中的首进程

15. 在利用信号量实现进程互斥时，应将①置于②和③之间。

①临界区 ②wait 操作 ③signal 操作

16. 在每个进程中访问①的那段代码称为临界区。为实现对它的共享，应保证进程②进入自己的临界区，为此，在每个进程的临界区前应设置③，临界区后应设置④。

①临界资源 ②互斥 ③进入区 ④退出区。

17. 进程通信的类型有①，②，③三类，其中④利用共享文件进程通信。

①共享存储器 ②消息系统 ③管道通信 ④管道通信

18. 为实现消息缓冲独队列通信，应在 PCB 中增加①，②，③三个数据项。

①消息队列首指针 mq ②消息到胡此信号量 mutex ③消息队列资源量 sm

19. 在采用用户线程的系统中，OS 进程 CPU 调度的对象是①，在采用内核支持线程的系统中，CPU 调度的对象是②。

①进程 ②线程

20. 线程之所以能减少并发执行的开销是因为①。

①线程基本不占用资源。

第三章：处理机调度和死锁

1. 高级调度又称作①调度，其主要功能是②；低级调度又称作③调度，其主要功能是④。

①作业 ②按照一定的算法从外存的后备队列中选若干作业进入内存，并为他们创建进程

③进程 ④按一定的算法从就绪队列中选一个进程投入执行

2. 进程调度有①和②两种方式。

①抢占调度 ②非抢占调度

3. 在抢占调度方式中，抢占的原则有①，②和③。

①时间片原则 ②短作业有限 ③优先权原则

4. 在设计进程调度程序时，应考虑①，②和③三个问题。

①引起调度的因素 ②调度算法的选择 ③就绪队列的组织

5. 为了使作业的平均周转时间最短，应该选择①调度算法；为了使当前执行的进程总是优先权最高的进程，则应选择②调度算法；而分时系统则常采用③调度算法。

①短作业优先 ②立即抢占的高优先权优先 ③时间片轮转

6. 高响应比优先的主要原因是①和②。

①运行时间 ②等待时间

7. 死锁产生的主要原因是①和②。

①竞争资源 ②进程推进顺序非法

8. 死锁产生的必要条件是①、②、③和④。

①互斥条件 ②请求与保持条件 ③不剥夺条件 ④环路等待条件

9. 通过破坏死锁产生的四个必要条件可进行死锁的预防，其中①条件一般是不允许破坏的，一次性分配所有资源破坏的是其中的②条件，资源的有序分配破坏的是其中的③条件。

①互斥 ②请求与保持 ③环路等待

10. 在避免死锁的方法中，允许进程动态地申请资源，但系统在进行分配时应先计算资源分配的①。若此次分配不会导致系统进入②，便将资源分配给它，否则进程③。

①安全性 ②不安全状态 ③等待

11. 解决死锁问题的方法有预防、避免、检测并解除等，一次性分配所有的资源采用的是其中的①方法，银行家算法采用的是其中的②方法。

①预防死锁 ②避免死锁

12. ①和②是解除死锁的两种常用方法。

①撤销进程 ②剥夺资源

第四章：存储器管理

1. 使每道程序能在内存中“各得其所”是通过①功能实现的；保证每道程序在不受干扰的环境下运行，是通过②功能实的；为缓和内存紧张的情况而将内存中暂时不能运行的进程调至外存，这是通过③功能实现的；能让较大的用户程序在较小的内存空间中运行，是通过④功能实现的。

①内存分配 ②内存保护 ③对换 ④内存扩充（或虚拟存储器）

2. 把作业装入内存中随即进行地址变换的方式称为①，而在作业执行期间，当访问到指令或数据时才进行地址变换的方式称为②。

①静态重定位 ②动态重定位

3. 地址变换机构的基本任务是将①中的②变换为③中的④。

①地址空间 ②逻辑地址 ③内存空间 ④物理地址

4. 通常，用户程序使用①地址，处理机执行程序时则必须用②地址。

①逻辑 ②物理

5. 在首次适应算法中，空闲区应以①的次序拉链；在最佳适应算法中，空闲区应以②的次序拉链。

①地址递增 ②空闲区大小递增

6. 在连续分配方式中可通过①来减少内存零头，它必须得到②技术的支持。

①紧凑 ②动态重定位

7. 在伙伴系统中，令 $buddy_k(x)$ 表示大小为 2^k、地址为 x 的块的伙伴的地址，则 $buddy_k(x)$ 的通用表达式为①。

① $x+2^k-\left[(x/2^k)\%2\right]*2^{k+1}$（其中“%2”表示除以 2 然后取余数）

8. 实现进程对换应具备①、②和③三方面的功能。

①对换空间的管理 ②进程换入 ③进程换出

9. 在请求调页系统中，地址变换过程可能会因为①、②和③等原因而产生中断。

①逻辑地址越界 ②缺页 ③访问权限错误

10. 分页系统中若页面较小，虽有利于①，但会引起②；若页面较大，虽可减少③，但会引起④。

①减少块内碎片 ②页表太长 ③页表长度 ④块内碎片增大

11. 分页系统中，页表的作用是实现①到②的转换。

①页号 ②物理块号

12. 在分页系统中为实现地址转换而设置了页表寄存器，其中存放了①和②；在进程未运行时，这些信息保存在③中。

①页表始址 ②页表长度 ③进程的 PCB

13. 引入分段主要是满足用户的需要，具体包括①、②、③、④等方面。

①便于编程 ②分段共享 ③分段保护 ④动态链接

14. 在页表中最基本的数据项是①，而在段表中则是②和③。

①物理块号 ②段的内存基址 ③段长

15. 把逻辑地址分为页号和页内地址是由①进行的，故分页系统的作业地址空间是②维的；把逻辑地址分成段号和段内地址是由③进行的，故分段系统的作业地址空间是④维的。

①机器硬件 ②一 ③程序员 ④二

16. 在段页式系统中（无快表），为获得一条指令或数据，都需三次访问内存。第一次从内存中取得①；第二次从内存中取得②；第三次从内存中取得③。

①页表起始地址 ②块号 ③指令或数据

17. 虚拟存储器的基本特征是①和②，因而决定了实现虚拟存储器的关键技术是③和④。

①多次性 ②对换性 ③请求调页（段） ④页（段）置换

18. 实现虚拟存储器除了需要有一定容量的内存和相当容量的内存外，还需要有①、②和③的硬件支持。

①页表机制 ②地址变换机构 ③缺页中断机构

19. 为实现请求分页管理，应在纯分页的页表基础上增加①、②、③和④等数据项。

①状态位 ②访问字段 ③修改位 ④外存地址

20. 在请求调页系统中，调页的策略有①和②两种方式。

①预调页 ②请求调页

21. 在请求调页系统中，反复进行页面换进和换出的现象称为①，它产生的原因主要是②。

①抖动 ②置换算法选用不当

22. 分页系统的内存保护通常有①和②两种措施。

①越界检查 ②存取控制

23. 分段系统中的越界检查是通过①中存放的②和段表中的③来实现的。

①段表寄存器 ②段表长度 ③段长

24. 在分段系统中常用的存储保护措施有①、②、③三种方式。

①越界检查 ②存取控制权限检查 ③环保护机构

◆判断题（此部分为网上摘录习题）

1、 早期的批处理系统中，用户可以用交互式方式方便地使用计算机。（ × ）

2、 分时系统中，时间片越小越好。（ × ）

当时间片过小时，进程调度时间所占比重加大。

3、 作业控制语言是供用户编写程序以实现某项计算任务。（ × ）

作业控制语言是供书写作业说明书的，以控制作业的执行(不同于编程语言)。

4、 对批处理作业，运行时不须提供相应的作业控制信息。……（ × ）

5、 联机用户接口是指用户与操作系统之间的接口，它不是命令接口。……（ × ）

6、 操作系统是系统软件中的一种，在进行系统安装时可以先安装其它软件，然后再装操作系统。（ × ）

7、 SPOOLing 系统实现设备管理的虚拟技术，即：将独占设备改造为共享设备，它由专门负责 I/O 的常驻内存的进程以及输入、输出井组成。（ √ ）

8、 批处理系统的（主要优点）是系统的吞吐量大、资源利用率高、系统的开销较小。(√)

9、 （Windows 98 操作系统）是支持多任务的操作系统。(√)

10、 原语是一种不可分割的操作。(√)

11、 交互式作业也称为脱机用户作业。(×)

改正:“脱机”改为“联机”

12、 用户程序有时也可以在核心态下运行. (×)

13、 实时系统中的作业周转时间有严格的限制. (×)

14、 执行系统调用时可以被中断. (√)

15、 原语和系统调用的主要区别在于两者的实现方法不同. (×)

16、 在大型多道程序设计系统中，为充分利用外部设备，应使运行的若干程序都是 I/O 型的. (√)

17、 设置中断屏蔽指令可以在目态下执行. (×)

18、 ＭＳ－ＤＯＳ是一种多用户操作系统。(×)

19、 在 UNIX 系统中，因为有 SPOOLING 技术存在，用户无法做真正的联机打印输出。(×)

20、 系统调用返回时由用户态变为核心态，执行核心程序。(×)

21、 系统调用返回时，由核心态变为用户态执行用户程序。(√)

22、 中断返回时，核心便执行与机器相关的特定指令序列，恢复中断时寄存器内容和执行核心栈退栈，进程回到用户态。(√)

23、 中断返回时便执行与机器相关的特定指令序列，恢复寄存器内容进程回到核态.(×)

24、 中断的处理是由硬件和软件协同完成的，各中断处理程序是操作系统的重要组成部分。所以，对中断的处理是在核心态下进行的。(√)

25、 各中断处理程序是操作系统的核心，所以，对中断的处理是在用户状态下进行的。(×)

26、 集中式保存是将中断的现场都统一保存在中断现场保存栈中，进栈和退栈操作由系统严格按照后进先出原则实施。(√)

27、 UNIX 系统中当处理机执行到 TRAP 指令时就进入陷入机构。 陷入处理子程序对用户态下的 TRAP 指令引起的陷入事件进行处理，先进行参数传递，然后执行相应的系统调用。(√)

28、 资源共享是现代操作系统的一个基本特征。(√)

29、 系统刚启动时，用户提交的作业称为前台作业。(×)

30、 若无进程处于运行状态，则就绪队列和等待队列均为空。(×)

就绪队列为空，等待队列可能不空。

31、 在动态优先级调度中，随着进程执行时间的增加，其优先级降低。……(√)

32、 程序在运行时需要很多系统资源，如内存、文件、设备等，因此操作系统以程序为单位分配系统资源。(×)

改正：程序（或者进程）在运行时需要很多系统资源，如内存、文件、设备等，因此操作系统以进程为单位分配系统资源。

33、 进程获得处理机而运行是通过申请而得到的. (×)

34、 进程控制块中的所有信息必须常驻内存. (×)

35、 优先数是进程调度的重要依据，一旦确定不能改变. (×)

36、 进程在运行中，可以自行修改自己的进程控制块. (×)

37、 进程申请 CPU 得不到满足时，其状态变为等待态. (×)

38、 当一个进程从等待态变成就绪态，则一定有一个进程从就绪态变成运行态. (×)

39、 在作业调度时，采用最高响应比优先的作业调度算法可以得到最短的作业平均周转时间. (×)

40、 进程优先数是进程调度的重要依据，必须根据进程运行情况动态改变. (×)

41、 进程状态的转换是由操作系统完成的，对用户是透明的. （ √ ）
42、 优先数是进程调度的重要依据，优先数大的进程首先被调度运行. （ × ）
43、 不可抢占式动态优先数法一定会引起进程长时间得不到运行. （ × ）
44、 进程从运行状态变为等待状态是由于时间片中断发生. （ × ）
45、 计算机中所有的共享资源都是临界资源。（ × ）
46、 进程是独立的，能够并发执行，程序也一样。（ × ）
47、 当条件满足时，进程可以由阻塞状态直接转换为运行状态。（ × ）
48、 当条件满足时，进程可以由阻塞状态转换为就绪状态。（ √ ）
49、 当条件满足时，进程可以由就绪状态转换为阻塞状态。（ × ）
50、 当某个条件满足时，进程可以由运行状态转换为就绪状态。（ √ ）
51、 进程的动态、并发特征是通过程序表现出来的。（ × ）
52、 在计算机系统中必须互斥地使用的资源都是硬件资源。（ × ）
53、 当进程间需要交换大量数据时，P，V 操作也能满足进程通信的要求。（ × ）
54、 UNIX 系统进程状态分为 10 种，在一定条件下进行变迁。（ √ ）
55、 处理机管理的实现策略决定了操作系统的类型,其算法的优劣不会影响整个系统的性能.（ × ）
56、 作业一经提交则立即进入内存并运行。（ × ）
57、 在操作系统中，核心进程具有较高的权力，可以随意进行进程的调度。（ × ）
58、 通常，在一定的 I/O 等待时间的百分比下，运行程序的道数越多，CPU 空闲时间的百分比越高。（ × ）
59、 先来先服务（FCFS）算法是一种简单的调度算法，但其效率比较高。（ × ）
60、 FCFS 调度算法对短作业有利。（ × ）
61、 时间片的大小对轮转法（RR）的性能有很大的影响，时间片太短，会导致系统开销大大增加。（ √ ）
62、 在分时系统中，进程调度都采用优先级调度算法为主，短进程优先调度算法为辅。（ × ）
63、 Windows98 中由于可以同时运行几个作业，所以可以被认为也是一种分时系统。（ × ）
64、 并发性是指若干事件在同一时刻发生。……（ √ ）
65、 对临界资源，应采用互斥访问方式来实现共享。……（ √ ）
66、 临界段是指进程中用于实现进程互斥的那段代码。……（ × ）
67、 对（临界资源）应采取互斥访问方式来实现共享。（ √ ）
68、 （并发性）是指若干个事件在不同时刻发生。（ × ）
改正： “不同时刻”改为“同一时间间隔内”
69、 通过任何手段都无法实现计算机系统资源之间的互换. （ × ）
70、 由于 P、V 操作描述同步、互斥等问题的能力不足，所以有必要引入其它的通讯原语或机制，如 send，receive 或 Monitor 等. （ × ）
71、 如果信号量 S 的当前值为-5，则表示系统中共有 5 个等待进程. （ × ）
72、 进程间的互斥是一种特殊的同步关系. （ √ ）
73、 即使在多道程序环境下，用户也能设计用内存物理地址直接访问内存的程序。…（ × ）
74、 在页式虚存系统中，为了提高内存利用率，允许用户使用不同大小的页面。……（ × ）
75、 虚拟存储器是利用操作系统产生的一个假想的特大存储器，是逻辑上扩充了内存容量，而物理内存的容量并未增加。（ √ ）
76、 分页式存储管理中，(页的大小）是可以不相等的。（ × ）
77、 操作系统的所有程序都必须常驻内存. （ × ）
78、 虚拟存储系统可以在每一台计算机上实现. （ × ）
79、 在虚存系统中，只要磁盘空间无限大，作业就能拥有任意大的编址空间. （ × ）

80、 在内存为 M 的分时系统中，当注册的用户有 N 个时,每个用户拥有 M/N 的内存空间. （ × ）

81、 大多数虚拟系统采用 OPT(优化)淘汰算法是因为它确实可以得到最小的缺页率. （ × ）

82、 在请求页式存储管理中，页面淘汰所花费的时间不属于系统开销. （ × ）

83、 页式存储管理中，用户应将自己的程序划分成若干相等的页. （ × ）

84、 存储保护的目的是限制内存的分配. （ × ）

85、 在页式虚拟存储系统中，页面长度是根据程序长度动态地分配的. （ × ）

86、 在页式虚拟存储系统中，页面长度固定并且是硬件的设计特性. （ √ ）

87、 存储保护的功能是限制内存存取. （ √ ）

88、 在页式存储管理方案中，为了提高内存的利用效率，允许同时使用不同大小的页面.（ × ）

89、 在虚拟存储方式下，程序员编制程序时不必考虑主存的容量，但系统的吞吐量在很大程度上依赖于主存储器的容量.（ √ ）

90、 固定分区式管理是针对单道系统的内存管理方案.（ × ）

91、 可重定位分区管理可以对作业分配不连续的内存单元.（ × ）

92、 利用交换技术扩充内存时，设计时必须考虑的问题是：如何减少信息交换量、降低交换所用的时间.（ √ ）

93、 在现代操作系统中，不允许用户干预内存的分配.（ √ ）

94、 采用动态重定位技术的系统，目标程序可以不经任何改动，而装入物理内存.（ √ ）

95、 页式存储管理中,一个作业可以占用不连续的内存空间,而段式存储管理,一个作业则是占用连续的内存空间.（ × ）

96、 所谓最近最少使用（LRU）页面调度算法是指将驻留在内存中使用次数最少的页面淘汰掉. （ × ）

97、 CPU 可以直接存取外存上的信息。（ × ）

98、 内存中的碎片，可以直接通过拼凑合并成为一个连续区。（ √ ）

99、 动态重定位技术使得作业在内存中可以移动。（ √ ）

100、虚拟存储器向用户提供了容量无限大的存储空间。（ × ）

101、虚拟存储器是利用操作系统产生的一个假想的特大存储器,在逻辑上扩充了内存容量,而物理内存容量并未增加。（ √ ）

102、作业一定要全部装入内存方可执行。（ × ）

103、紧缩法是唯一的解决碎片问题的方法。（ × ）

104、分页、请求分页存储管理技术的逻辑地址由页号 p 和页内地址 d 组成，因此是一个二维地址空间。（ × ）

◆名词解释

第一章：操作系统引论

1 操作系统：操作系统是管理和控制计算机系统内各种硬件和软件资源，有效地组织多道程序运行的系统软件(或程序集合)，是用户与计算机之间的接口。

2 管态：当执行操作系统程序时，处理机所处的状态

3 目态：当执行普通用户程序时，处理机所处的状态。

4 多道程序设计：在这种设计技术下，内存中能同时存放多道程序，在管理程序的控制下交替的执行。这些作业共享 CPU 和系统中的其他资源。

5 并发：是指两个或多个活动在同一给定的时间间隔中进行。它是宏观上的概念。

6 并行：是指两个或多个活动在同一时刻同时执行的情况。

7 吞吐量：在一段给定的时间内，计算机所能完成的总工作量。

8 分时：就是对时间的共享。在分时系统中，分时主要是指若干并发程序对 CPU 时间的共享。

9 实时：表示“及时”或“既时”。

10 系统调用：是用户在程序中能以“函数调用”形式调用的、由操作系统提供的子功能的集合。每一个子功能称作一条系统调用命令。它是操作系统对外的接口，是用户级程序取得操作系统服务的唯一途径。

11 特权指令：指指令系统中这样一些指令，如启动设备指令、设置时钟指令、中断屏蔽指令和清内存指令，这些指令只能由操作系统使用。

12 命令解释程序：其主要功能是接收用户输入的命令，然后予以解释并且执行。

13 脱机 I/0：是指输入/输出工作不受主机直接控制，而由卫星机专门负责完成 I/0，主机专门完成快速计算任务，从而二者可以并行操作。

14 联机 I/0：是指作业的输入、调入内存及结果输出都在 cpu 直接控制下进行。

15 资源共享：是指计算机系统中的资源被多个进程所功用。例如，多个进程同时占用内存，从而对内存共享；它们并发执行时对 cpu 进行共享；各个进程在执行过程中提出对文件的读写请求，从而对磁盘进行共享等等。

第二章：进程管理

1 顺序性：是指顺序程序所规定的每个动作都在上个动作结束后才开始的特性。

2 封闭性：是指只有程序本身的动作才能改变程序的运行环境。

3 可再现性：是指程序的执行结果与程序运行的速度无关。

4 进程：程序在并发环境中的执行过程。

5 互斥：在逻辑上本来完全独立的进程，由于竞争同一个资源而产生的相互制约的关系。

6 同步：是指进程间共同完成一项任务时直接发生相互作用的关系。也就是说，这些具有伙伴关系的进程在执行次序上必须遵循确定的规律。

7 异步：是指程序在执行某一个操作时，只是发出开始的指令；由另外的并行程序执行这段代码，当完成时再通知调用者 。

8 临界资源：一次仅允许一个进程使用的资源。

9 临界区：在每个进程中访问临界资源的那段程序。

10 线程：线程是进程中实施调度和分派的基本单位。

11 管程：管程是一种高级同步机制，一个管程定义一个数据结构和能为并发进程在其上执行的一组操作，这组操作能使进程同步和改变管程中的数据。

12 进程控制块：进程控制块是进程存在的唯一标识，它保存了系统管理和控制进程所必须的信息，是进程动态特性的集中表现。

13 原语：指操作系统中实现一些具有特定功能的程序段，这些程序段的执行过程是不可分割的，即其执行过程不允许被中断。

14 就绪态：进程已经获得了除 cpu 之外的全部资源，等待系统分配 cpu，一旦获得 cpu，进程就可以变为运行态。

15 运行态：正在 cpu 上执行的进程所处的状态。在单 cpu 系统中，任何时候最多只能有一个进程处于运行状态。

16 阻塞态：又称等待态，指正在运行的进程因等待某个条件发生而不能运行时所处的状态。处于阻塞态的进程在逻辑上是不能运行的，即使 cpu 空闲，它也不能占用 cpu。

17 进程通信：是指进程间的信息交换。

18 同步机制：同步机构是负责处理进程之间制约关系的机制，即操作系统中负责解决进程之间协调工作的同步关系（直接制约关系），以及共享临界资源的互斥关系（间接制约关系）的执行机构。

第三章：处理机调度和死锁

1 死锁：是指在一个进程集合中的每个进程都在等待仅由该集合中的另一个进程才能引发的事件而无限期地僵持下去的局

面。

2 饥饿：在系统中，每个资源占有者都在有限时间内释放它所占有的资源，但资源中存在某些申请者由于某种原因却永远得不到资源的一种错误现象。

3 死锁防止：要求进程申请资源时遵循某种协议，从而打破产生死锁的四个必要条件中的一个或几个，保证系统不会进入死锁状态。

4 死锁避免：对进程所发出的每一个申请资源命令加以动态地检查，并根据检查结果决定是否进行资源分配。就是说，在资源分配过程中若预测有发生死锁的可能性，则加以避免。这种方法的关键是确定资源分配的安全性。

5 安全序列：针对当前分配状态来说，系统至少能够按照某种次序为每个进程分配资源（直至最大需求），并且使他们依次成功地运行完毕，这种进程序列{p1, p2, …, pn}就是安全序列。

6 作业：用户在一次上机过程中要求计算机系统所做工作的集合。

7 作业步：一般情况下，一个作业可划分成若干个部分，每个部分称为一个作业步。

8 周转时间：是指从作业进入系统开始，到作业退出系统所经历的时间。

9 响应时间：是分时系统的一个技术指标，指从用户输入命令到系统对命令开始执行和显示所需要的时间。

10 作业调度：作业调度的主要任务是完成作业从后备状态到执行状态和从执行状态到完成状态的转换。

11 进程调度：也称低级调度程序，它完成进程从就绪状态到运行状态的转化。实际上，进程调度完成一台物理的 cpu 转变成多台虚拟（或逻辑）的 cpu 的工作。

12 交换调度：是基于系统确定的某个策略，将主存中处于等待状态或就绪状态的某个或某些进程交换到外存交换区中，以便将外存交换区上具备运行条件的进程换入主存，准备执行。引入交换调度的目的是为了解决主存紧张和提高主存的利用效率。

13 剥夺式调度：当一个进程正在执行时，系统基于某种策略强行将处理机从占有者进程剥夺而分配给另一个进程的调度。这种调度方式系统开销大，但系统能及时响应请求。

14 非剥夺式调度：系统一旦把处理机分配给某个进程之后，该进程一直运行下去，直到该进程完成或因等待某个事件发生时，才将处理机分配给其他进程。这种调度方式实现简单，系统开销小，但系统性能不够好。

第四章：存储器管理

1 物理地址：内存中各存储单元的地址由统一的基地址顺序编址，这种地址称为物理地址。

2 逻辑地址：用户程序经编译之后的每个目标模块都以 0 为基地址顺序编址，这种地址称为逻辑地址。

3 逻辑地址空间：由程序中逻辑地址组成的地址范围叫做逻辑地址空间。

4 物理地址空间：由内存中的一系列存储单元所限定的地址范围称作内存空间。

5 重定位：把逻辑地址转变为内存物理地址的过程叫做重定位。

6 静态重定位：在目标程序装入内存时所进行的重定位。

7 动态重定位：在程序执行期间，每次访问内存之前进行的重定位。

8 内部碎片：在一个分区内部出现的碎片（即被浪费的空间）称作内部碎片。如固定分区法会产生内部碎片。

9 外部碎片：在所有分区之外新产生的碎片称作外部碎片，如在动态分区法实施过程中出现的越来越多的小空闲块，由于它们太小，无法装入一个小进程，因而被浪费掉。

10 碎片：在分区法中，内存出现许多容量太小、无法被利用的小分区称作“碎片”。

11 紧缩：移动某些已分区的内容，使所有作业的分区紧挨在一起，而把空闲区留在另一端，这种技术称为紧缩。

12 可重定位地址：当含有它的程序被重定位时，将随之被调整的一种地址。

13 固定分区法：内存中分区的个数固定不变，各个分区的大小也固定不变，但不同分区的大小可以不同，每个分区只可装入一道作业。

14 动态分区法：各个分区是在相应作业要求进入内存时才建立的，使其大小恰好适应作业的大小。

15 可再入代码：也称纯代码，是指那些在其执行过程本身不做任何修改的代码，通常由指令和常数组成。

16 虚拟存储器：虚拟存储器是用户能作为可编程内存对待的虚拟存储空间，在这种计算机系统中实现了用户逻辑存储器与物理存储器的分离，它是操作系统给用户提供的一个比真实内存空间大得多的地址空间。

17 抖动：页面抖动是系统中频繁进行页面置换的现象。即如果一个进程没有一定数量的内存块，它很快就发生缺页。此时，它必须淘汰某页。由于所有这些页面都正在使用，所以刚被淘汰出去的页很快又被访问，因而要把它重新调入。可是调入不久又再被淘汰出去，这样再访问，再调入，如此反复，使得整个系统的页面替换非常频繁，以致大部分机器时间都用在来回进行的页面调度上，只有一小部分时间用于进程的实际运算方面。

18 工作集：工作集是一个进程在某一小段时间内访问页面的集合。利用工作集模型可防止抖动，也可以进行页面置换。

19 程序局部性原理：在相对短的一段时间内，进程集中在一组子程序或循环中之行，导致所有的存储器访问局限于进程地址空间的一个固定子集。这种现象就叫做程序局部性原理。

20 快表：又叫“联想存储器”。在分页系统中，由于页表是存放在主存中的，因此 cpu 存取一个数据时要访问两次主存。这样使计算机的处理速度降低约一倍。为了提高地址变换速度，在地址变换机构中增设一个具有并行查找能力的高速缓冲存储器，用以存放当前访问的页表项。这样的高速缓冲存储器就是快表。

21 交换：交换系统指系统根据需要把主存中暂时不运行的某个（或某些）作业部分或全部移到外存。而把外存中的某个（或某些）作业移到相应的主存区，并使其投入运行。

22 换页：指系统根据某种策略选择某页出主存，将某页调入主存的过程。

23 实存：实存是指计算机配置的物理存储器，它直接向 cpu 提供程序和数据。

24 虚存：虚存是指系统向用户程序提供的编程空间，其大小由 cpu 的地址长度决定。

◆简答题

❖ 画图——多道程序

【例】有三个程序 A、B、C，它们使用同一个设备进行 I/O 操作，并按 A、B、C 的优先次序执行。这三个程序的计算和 I/O 操作时间如表所示。假设调度的时间可忽略不计，请分别画出单道程序环境和多道程序环境下（假设内存中可同时装入这三道程序），它们运行的时间关系图，并比较它们的总运行时间。

表 程序运行的时间表 （单位：ms）

程序 / 操作	A	B	C
计算	30	60	20
I/O	40	30	40
计算	10	10	20

【解】单道程序环境下，它们运行的时间关系如图 1 所示，总的运行时间为 260ms。

多道程序环境下，如果 CPU 不能被抢占，则它们运行的时间关系如图 2 所示，总的运行时间为 180ms；如果 CPU 可被抢占，则它们运行的时间关系如图 3 所示，总的运行时间为 190ms。

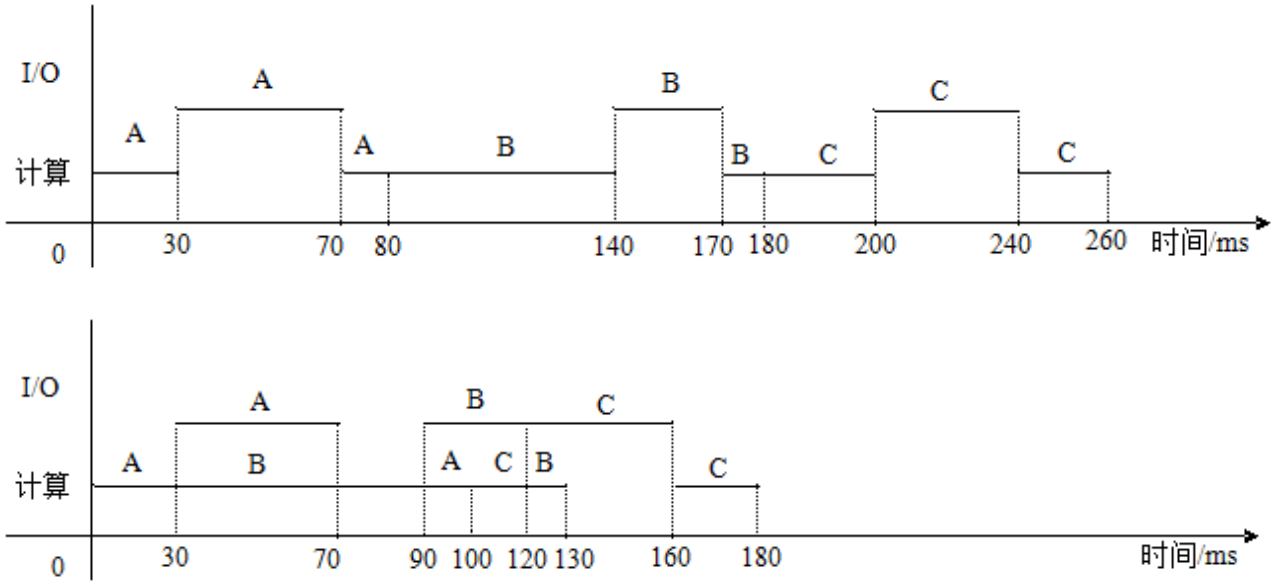


图 2 多道、非抢占式运行的时间关系图

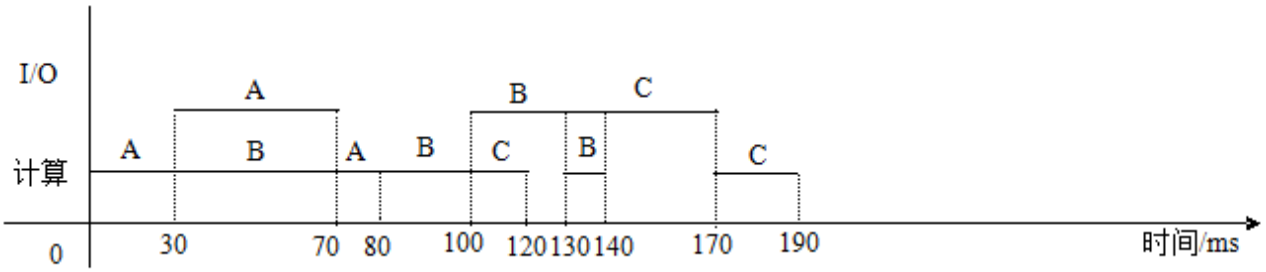


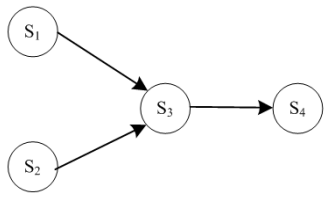
图 3 多道、抢占式运行的时间关系图

❖ 画图——根据代码，画出前趋图

【例 1】试画出下面 4 条语句的前驱图：

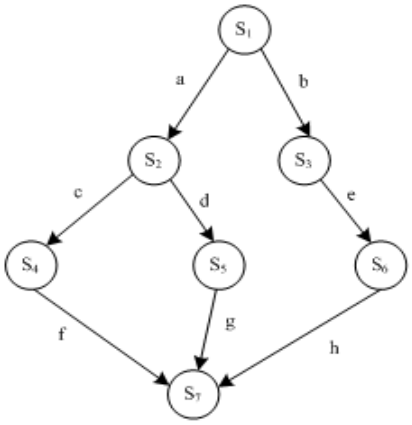
S1:a:=x+y;
S2:b:=z+1;
S3:c=a-b;
S4:w:=c+1;

【解】:

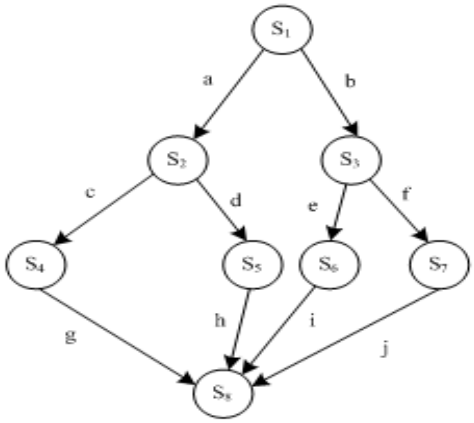


【例 2】试写出相应的程序来描述图 2-17 所示的前驱图。 （或者根据代码画出前驱图）

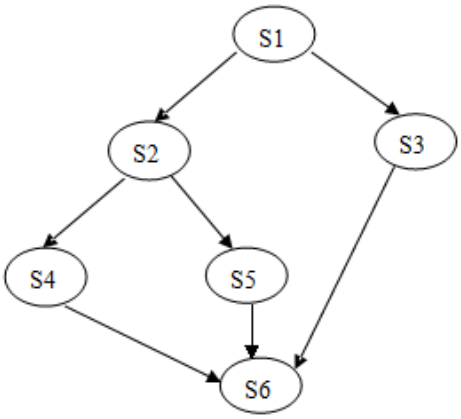
（1）Var a,b,c,d,e,f,g,h:semaphore:=0,0,0,0,0,0,0,0;
Begin
begin S_1; Signal(a); Signal(b);end;
begin wait(a); S_2;Signal(c); Signal(d);end;
begin wait(b); S_3;Signal(e); end;
begin wait(c); S_4;Signal(f);end;
begin wait(d); S_5;Signal(g); end;
begin wait(e); S_6;Signal(h);end;
begin wait(f); wait(g); wait(h); S_7; end;
end



（2）Var a,b,c,d,e,f,g,h,i,j:semaphore:=0,0,0,0,0,0,0,0,0,0;
Begin
begin S_1; Signal(a); Signal(b);end;
begin wait(a); S_2;Signal(c); Signal(d);end;
begin wait(b); S_3;Signal(e); Signal(f);end;
begin wait(c); S_4;Signal(g);end;
begin wait(d); S_5;Signal(h); end;
begin wait(e); S_6;Signal(i);end;
begin wait(f); S_7;Signal(j);end;
begin wait(g); wait(h); wait(i); wait(i); S_8; end;
End



（3）var a,b,c,d,e,f,g:semaphore:=0,0,0,0,0,0,0;
begin
parbegin
begin S1; signal(a);signal(b);end;
begin wait(a);S2;signal(c);signal(d);end;
begin wait(b);S3;signal(e);end;
begin wait(c);S4;signal(f);end;
begin wait(d);S5;signal(g);end;
begin wait(e);wait(f);wait(g);S6;end;
parend
end



❖ 算法——页面置换

（1）先进先出页面置换算法

（2）最近最久未使用置换算法

【例】对于如下的页面访问序列：

1， 2， 3， 4， 1， 2， 5， 1， 2， 3， 4， 5

当内存块数量为 3 时，试问：使用 FIFO、LRU 置换算法产生的缺页中断是多少？写出依次产生缺页中断后应淘汰的页。（所有内存开始时都是空的，凡第一次用到的页面都产生一次缺页中断。要求写出计算步骤。）

【解】采用先进先出（FIFO）调度算法，页面调度过程如下：

页面次序	1	2	3	4	1	2	5	1	2	3	4	5
主存	1	1	1	4	4	4	5			5	5	
页面		2	2	2	1	1	1			3	3	
情况			3	3	3	2	2			2	4	

共产生缺页中断 9 次。依次淘汰的页是 1、2、3、4、1、2。

采用最近最少使用（LRU）调度算法，页面调度过程如下：

页面次序	1	2	3	4	1	2	5	1	2	3	4	5
主存	1	1	1	4	4	4	5			3	3	3
页面		2	2	2	1	1	1			1	4	4
情况			3	3	3	2	2			2	2	5

共产生缺页中断 10 次。依次淘汰的页是 1、2、3、4、5、1、2。

【例 2】假定在一个请求页式存储管理系统中，某作业 J 所涉及的页面依次为：

3,2，1,4，4,5，3,4，3,2，1,5

并已知主存中有 3 个可供作业 J 使用的空白存储块(块的大小与页面大小相同)，试说明采用 FIFO 和 LRU 两种算法进行页面置换时，缺页中断的次数各是多少。

【解】先进先出算法的实质是：总是选择作业中在主存驻留时间最长(即最老)的一页淘汰，即先进入主存的页，先退出主存。在本例中，给出了页面踪迹，只需要按页面使用的顺序去进行页面的替换，记录缺页次数即可。若在内存中为每一作业进程开辟 3 页，对于题中的页面访问过程，采用先进先出(FIFO)淘汰算法，其页面调度过程见表 1 。

表 1　采用先进先出(FIFO)淘汰算法，其页面访问过程

页面 1		3	3	3	2	2	1	4	4	4	5	3	2
页面 2			2	2	1	1	4	5	5	5	3	2	1
页面 3				1	4	4	5	3	3	3	2	1	5
页面	3	2	1	4	4	5	3	4	3	2	1	5	
缺页	缺	缺	缺	缺	缺	缺	缺		缺	缺			

最近最少使用（LRU）算法的基本思想是根据一个作业在执行过程中过去的页面踪迹来推测未来的行为。它认为过去一段时间里不曾被访问过的页，在最近的将来可能也不会再被访问。这种算法的实质是：当需要置换一页时，选择在最近一段时间内最久未用的页予以淘汰。采用最近最少使用（LRU）淘汰算法，其页面调度过程见表 2

表 2　采用最近最少使用(LRU)淘汰算法，其页面访问过程

内存中页面 1	3	3	3	2	2	1	4	5	5	4	3	2
内存中页面 2		2	2	1	1	4	5	3	4	3	2	1
内存中页面 3				1	4	4	5	3	4	3	2	1	5
页面	3	2	1	4	4	5	3	4	3	2	1	5	
缺页	缺	缺	缺	缺		缺	缺			缺	缺	缺	

由此可知：假设系统初始时在内存中没有页面。

① 采用先进先出（FIFO）淘汰算法，缺页中断次数为 9 次。

② 采用最近最少使用（LRU）淘汰算法，缺页中断次数为 9 次。

【例 3】考虑下面的页访问串：1，2，3，4，2，1，5，6，2，1，2，3，7，6，3。假定物理块数为 3，若应用下面的页面替换算法，分别会出现多少次缺页？

（1）LRU 替换法算法　　（2）FIFO 替换算法　　（3）Optimal 替换算法

答：LRU 算法的情况如下表：

页面走向	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3
物理页 0	1	1	1	4		4	5	5	5	1		1	7	7	
物理页 1		2	2	2		2	2	6	6	6		3	3	3	
物理页 2			3	3		1	1	1	2	2		2	2	6	
缺页否	Y	Y	Y	Y		Y	Y	Y	Y	Y		Y	Y	Y	

缺页次数为 12。

FIFO 算法的情况如下表：

页面走向	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3
物理页 0	1	1	1	4		4	4	6	6	6		3	3	3	
物理页 1		2	2	2		1	1	1	2	2		2	7	7	
物理页 2			3	3		3	5	5	5	1		1	1	6	
缺页否	Y	Y	Y	Y		Y	Y	Y	Y	Y		Y	Y	Y	

缺页次数为 12。

Optimal 算法的情况如下表：

页面走向	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3
物理页 0	1	1	1	1			1	1				3	3		
物理页 1		2	2	2			2	2				2	7		
物理页 2			3	4			5	6				6	6		
缺页否	Y	Y	Y	Y			Y	Y				Y	Y		

缺页次数为 8。

❖ 计算——分页存储

【例 1】对于一个将页表存放在内存中的分页系统：

如果访问内存需要 0.2us，有效访问时间为多少？

如果加一快表，且假定在快表中找到页表项的几率高达 90%，则有效访问时间又是多少（假定查快表需发的时间为 0）？

【分析】每次访问数据时，若不使用快表，则需两次访问内存，即先从内存的页表中读出页对应的块号，然后再根据形成的物理地址去存取数据；使用快表时，若能从快表中直接找出对应的页表项，则可立即形成物理地址去访问相应的数据，否则，仍需两次访问内存。

【解】（1）有效访问时间为 $2\times0.2=0.4\mu s$

（2）有效访问时间为 $0.9\times0.2+(1-0.9)\times2\times0.2=0.22\mu s$

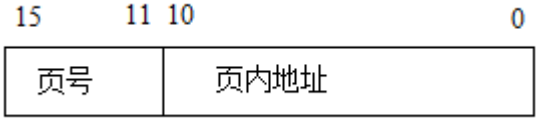
【例 2】某系统采用页式存储管理策略，拥有逻辑空间 32 页，每页为 2KB，拥有物理空间 1MB。

（1）写出逻辑地址的各省市。

（2）若不考虑访问权限等，进程的页表有多少项？每项至少有多少位？

（3）如果物理空间减少一半，页表结构应相应作怎样的改变？

【解】（1）该系统拥有逻辑空间 32 页，故逻辑地址中页号必须用 5 位来描述，而每页为 2KB，因此，页内地址必须用 11 位来描述。这样，可得到它的逻辑地址格式如下：



（2）每个进程最多有 32 个页面，因此，进程的页表项最多为 32 项；若不考虑访问权限等，则页表项中只需给出页所对应

的物理块块号。1MB 的物理空间可分成 2^9 个内存块，故每个页表项至少有 9 位。

（3）如果物理空间减少一半，则页表中页表项数仍不变，但每项的长度可减少 1 位。

【例 3】已知某分页系统，主存容量为 64KB，页面大小为 1KB。对于一个 4 页大的作业，其 0、1、2、3 页分别被分配到主存的 2、4、6、7 块中。

（1）将十进制的逻辑地址 1023、2500、3500、4500 转换成物理地址；

（2）以十进制的逻辑地址 1023 为例画出地址变换过程图。

【分析】在分页系统中进行地址转换时，地址变换机构将自动把逻辑地址转化为页号和页内地址，如果页号不小于页表长度，则产生越界中断；否则便以页号为索引去检索页表，从中得到对应的块号，并把块号和页内地址分别送入物理地址寄存器的块号和块内地址字段中，形成物理地址。

【解】（1）对于上述逻辑地址，可先计算出它们的页号和页内地址（逻辑地址除以页面大小得到的商为页号，余数为页内地址），然后通过页表转换成对应的物理地址：

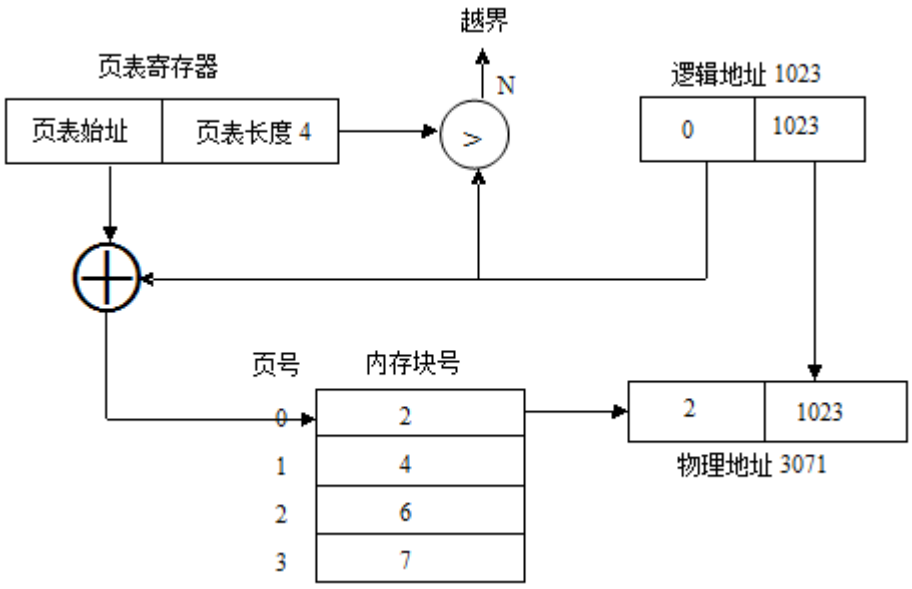
①逻辑地址 1023。1023/1K，得到页号为 0，页内地址为 1023，查页表找到对应的物理块号为 2，故物理地址为 $2\times1K+1023=3071$。

②逻辑地址 2500。2500/1K，得到页号为 2，页内地址为 452，查页表找到对应的物理块号为 6，故物理地址为 $6\times1K+452=6596$。

③逻辑地址 3500。3500/1K，得到页号为 3，页内地址为 428，查页表找到对应的物理块号为 7，故物理地址为 $7\times1K+428=7596$。

④逻辑地址 4500。4500/1K，得到页号为 4，页内地址为 404，因页号不小于页表长度，故产生越界中断。

（2）逻辑地址 1023 的地址变换过程如下图所示，其中的页表项中没有考虑每页的访问权限。



【例 4】已知某系统页面长度为 4KB，每个页表项为 4B，采用多层分页策略映射 64 位的用户地址空间。若限定最高层页表只占 1 页，则它可采用几层分页策略？

【解】由题意可知，该系统的用户地址空间为 $2^{64}B$，而页的大小为 4KB，故一作业最多可有 $2^{64}/2^{12}$（即2^{52}）个页，其页表的大小则为 $2^{52}\times4$(即2^{54})B。因此，又可将页表分成 2^{42} 个页表页，并为它建立两级页表，两级页表的大小为 $2^{44}B$。依此类推，可知道它的 3、4、5、6 级页表的长度分别是 $2^{34}B$、$2^{24}B$、$2^{14}B$、$2^{4}B$，故必须采用 6 层分页策略。

❖ 计算——分段存储

【例】对于下表所示的段表，请将逻辑地址（0,137），（1,4000），（2,3600），（5,230）转换成物理地址。

表 段表

段号	内存地址	段长
0	50K	10KB
1	60K	3KB
2	70K	5KB
3	120K	8KB
4	150K	4KB

【分析】在分段系统中进行地址转换时，地址变换机构首先将逻辑地址中的段号与段表长度作比较，如果段号超长，则产生越界中断；否则便以段号为索引去检索段表，从中得到段在内存的始址和段长；然后再将逻辑地址的段内地址与段表项中的段长做比较，若不越界，则由段的始址与段内地址相加，形成物理地址。

【解】（1）段号 0 小于段表长 5，故段号合法；由段表的第 0 项可获得段的内存始址为 50K，段长为 10KB；由于段内地址为 137，小于段长 10KB，故段内地址也是合法的，因此可得出对应的物理地址为 50KB+137=51337。

（2）段号 1 小于段表长 5，故段号合法；由段表的第 1 项可获得段的内存始址为 60K，段长为 3KB；经检查，段内地址 4000 超过段长 3KB，因此产生越界中断。

（3）段号 2 小于段表长 5，故段号合法；由段表的第 2 项可获得段的内存始址为 70K，段长为 5KB；由于段内地址为 3600 页合法。因此可得出对应的物理地址为 70KB+3600=75280。

（4）段号 5 等于段表长 5，故段号不合法，产生越界中断。

【例 2】现有一个作业，在段式存储管理的系统中已为其主存分配，建立的段表内容如下：

段号	主存起始地址	段长度
0	120	40
1	760	30
2	480	20
3	370	20

计算逻辑地址（2，15），（0，60），（3，18）的绝对地址是多少？

注：括号中第一个元素为段号，第二个元素为段内地址。

【解】段式存储管理的地址转换过程为：（1）根据逻辑地址中的段号查段表的相应栏目；（2）根据段内地址<段长度，检查地址是否越界；（3）若不越界，则绝对地址=该段的主存起始地址+段内地址。

逻辑地址（2，15）查段表得段长度为 20，段内地址 15<20，地址不越界，段号 2 查表得段首地址为 480，于是绝对地址为 480+15=495。

逻辑地址（0，60）查段表得段长度为 40，段内地址 60>40，地址越界，系统发出“地址越界”中断。

逻辑地址（3，18）查段表得段长度为 20，段内地址 18<20，地址不越界，段号 3 查表得段首地址为 370，于是绝对地址=370+18=388。

◆综合题

❖ 算法——银行家

【例】在银行家算法中，若出现下面的资源分配情况：

Process	Allocation	Need	Available
P0	0 0 3 2	0 0 1 2	1 6 2 2
P1	1 0 0 0	1 6 5 0	
P2	1 3 5 4	2 3 5 6	
P3	0 0 3 2	0 6 5 2	
P4	0 0 1 4	0 6 5 6	

试问：

（1）该状态是否安全？

（2）当进程 P2 提出请求 Request（1,2,2,2）后，系统能否将资源分配给它？

（3）如果系统立即满足 P2 的上述请求，则系统是否立即进入死锁状态？

【解】（1）利用安全性算法对上面的状态进行分析（见下表），找到一个安全序列{P0，P3，P4，P1，P2}，故系统是安全的。

表 安全性检查过程

资源情况 / 进程	Work	Need	Allocation	Work+allocation	Finish
	A B C D	A B C D	A B C D	A B C D	
P0	1 6 2 2	0 0 1 2	0 0 3 2	1 6 5 4	True
P3	1 6 5 4	0 6 5 2	0 0 3 2	1 6 8 6	True
P4	1 6 8 6	0 6 5 6	0 0 1 4	1 6 9 10	True
P1	1 6 9 10	1 6 5 0	1 0 0 0	2 6 9 10	True
P2	2 6 9 10	2 3 5 6	1 3 5 4	3 9 14 14	True

（2）P2 发出请求向量 Request（1,2,2,2）后，系统按银行家算法进行检查：

① $\mathrm{Re}\,quest_2(1,2,2,2) \le \mathrm{Need}_2(2,3,5,6)$

② $\mathrm{Re}\,quest_2(1,2,2,2) \le \mathrm{Available}(1,6,2,2)$

③系统先假定可为 P2 分配资源，并修改 Available，$\mathrm{Allocation}_2$ 和 $Need_2$ 向量：

Available=（0,4,0,0）

$\mathrm{Allocation}_2$ =(2,5,7,6)

$Need_2$ =(1,1,3,4)

④进行安全性检查：此时对于所有的进程，条件 $Need_2 \le Available(0,4,0,0)$ 都不成立，即 Available 不能满足任何进程的请求，故系统进入不安全状态。

因此，当进程 P2 提出请求 Request（1,2,2,2）后，系统不能将资源分配给它。

（3）系统立即满足进程 P2 的请求（1,2,2,2）后，并没有马上进入死锁状态。这是因为，此时上述进程并没有申请新的资源，并因得不到资源而进入阻塞状态；只有当上述进程提出新的请求，导致所有没有执行完的多个进程因得不到资源而阻塞并形成循环等待链时，系统才进入死锁状态。

❖ 编程——进程同步

【例 1】读者和写者问题。有一数据区可为多个进程共享。假设一些进程只能对数据区进行读操作（读者），而另一些进程只能进行写操作（写者），读者和写者需要遵守如下规则：

(1) 多个读者可以同时从数据区取数据；

(2) 读、写操作不能同时进行；

(3) 写操作不能同时进行。

【解】这是一个经典的问题，该问题首先在 1971 年由 Courtois 等人解决。根据规则可知，写操作与写操作是互斥的，写操作与所有的读操作是互斥的，读操作与读操作之间是可以同时进行的。

（1）读写进程操作的流程如下：

〈写进程〉

是否有读进程读数据？

有，则等待；

是否有写进程写数据？

有，则等待；

写数据；

设立可写标志；

设立可读标志。

〈读进程〉

是否有写进程写数据？

有，则等待；

读数据；

设立可写标志

（2）信号量及其含义

根据上述的流程，可以得出需要 3 个信号量：

WR-M 是读写互斥，用于读进程与写进程之间的互斥，表示是否没有读写操作，可否进行读写操作；

W-M 是用于写进程与写进程之间的互斥，表示是否没有写操作，是否可以进行写操作；

count-M 是用于读进程互斥操作读进程个数变变量 readcount，其含义为 readcount 是否可操作。

这里关键问题是即要实现读写互斥，又要实现并发的读操作。为此引入一个变量 readcount，来记录当前需要进行读操作进程的个数，当 readcount>0 时，表示有读操作进程，readcount=0 时表示没有读操作进程。因此，当 readcount 开始大于 0 时，即 readcount=1 时，开始读进程开始读写互斥，readcount=0 时，读进程撤消读写互斥。这样就把每个读进程与写进程的互斥转化为整个读操作与写操作的互斥。

（3）P、V 操作的确定

〈写进程〉

P(WR-M)

P(W-M)

写数据

V(WR-M)

V(W-M)

〈读进程〉

P(count-M)

Readcount= Readcount+1

If Readcount=1 then P(WR-M);

V(count-M)

读数据

P(count-M)

Readcount= Readcount-1

If Readcount=0 then V(WR-M);

V(count-M)

以上的解法是读进程优先的，也就是说一旦一个读进程开始读操作，只要还有需要进行读操作的进程，那么，就要保持读操作的独占，直到没有读进程操作的时候，即读进程记数 Readcount=0 时，才能允许写进程进行操作。这里可以将 WR-M 和 W-M 合并成一个信号量，现在这样做可以更容易理解些。

（4）信号量的初值

WR-M=1；W-M=1；count-M=0。

【例 2】多个进程共享一个文件，其中只读文件的称为读者，只写文件的称为写者。读者可以同时读，但写者只能独立写。请：（1）说明进程间的相互制约关系，应设置哪些信号量？

（2）用 P,V 操作写出其同步算法。

（3）修改上述的同步算法，使得它对写者优先，即一旦有写者到达，后续的读者必须等待。而无论是否有读者在读文件。

【解】本题前两问是经典读者写者问题，第三问对读者写者问题加了一些限制，即使算法对写者优先。

（1）进程间的相互制约关系有三类：一是读者之间允许同时读；二是读者与写者之间须互斥；三是写者之间须互斥。

为了解读者，写者之间的同步，应设置两个信号量和一个共享变量：读互斥信号量 rmutex，用于使读者互斥地访问共享变量 count,其初值为 1；写互斥信号量 wmutex,用于实现写者与读者的互斥及写者与写者的互斥，其初值为 1；共享变量 count，用于记录当前正在读文件的读者数目，初值为 0。

（2）进程间的控制算法如下所示：

semaphore rmutex=1;

semaphore wmutex=1;

```
 main()
{
  cobegin
      reader ();
      writer ();
  coend
}
reader()
{
     while (1)
      {
          p (rmutex);
          if  (count==0)  p(wmutex);  /*当最后一个读者读完文件时，阻止写者写*/
          count++;
          v(rmutex);
          读文件；
          p(rmutex);
          count-- ;
          if  (count==0)  v(wmutex);  /*当最后一个读者读完文件时，允许写者写*/
          v（rmutex）;
      }
  }
writer()
{
     while(1)
      {
          p(wmutex);
          写文件；
          v(wmutex);
      }
  }
```

（3）为了提高写者的优先级，增加一个信号量 s，用于在写进程到达后封锁后续的读者。其控制流程如下：

```
semaphore rmutex=1;
semaphore wmutex=1;
semaphore count=0;
semaphore s=1;
main()
{
  cobegin
     reader();
     writer();
  coend
}
reader()
{
  while (1)
   {
     p(s);
     p(rmutex);
     if(count==0)  p(wmutex);  /*当前最后一个读者读完文件时，允许写者写*/
     v(rmutex);
    }
  }
writer()
{
     while()
      {
        p(s);
        p(wmutex);
        写文件；
        v(wmutex);
        v(s);
      }
}
```