**操作系统硕士研究生入学考试模拟试题参考答案**

1. **填空题** 
   1. 中断优先级是由硬件规定的，若要调整中断响应次序可通过中断屏蔽实现。(北京大学1997)
   2. 多道程序设计的特点是多道、宏观上并行和微观上串行。(西安电子科大2001)
   3. 多道运行的特征之一是宏观上并行，它的含义是同时进入主存的程序都处于运行过程中，但都未运行完毕。(华中理工2000)
   4. 操作系统的基本类型主要有批处理操作系统、分时操作系统和实时操作系统。(哈工大2002)
   5. 批处理系统主要解决吞吐量问题，分时系统主要解决交互性问题。(华中理工2001)
   6. 操作系统具备处理同时性活动的能力，其最重要的硬件支持是中断机构。(华中科技2001)
   7. 为了实现处理器与外部设备的并行工作，系统引入了中断硬件机制。(北大1997)
   8. 操作系统向用户提供了两类接口，一类是命令级接口，另一类是程序级接口。(西安电子科大 2001)(中科大1998)
   9. 现代操作系统的两个最基本的特征是 (程序执行的) 并发性和 (资源的) 共享性。(中科院1997) (北交大2000)
   10. 多道程序设计的引入给存储管理提出了新的课题，应考虑的三个问题是存储分配、存储保护和存储扩充。(西安电子科大2000)
   11. 在操作系统中，一种用空间换取时间的资源转换技术是SPOOLing。(北京大学1997)
   12. 如果系统中有 n 个进程，则在等待队列中进程的个数最多为 n 个。(北京大学1997)
   13. 若使当前运行进程总是优先级最高的进程，应选择剥夺式进程调度算法。(北京大学1997)
   14. 实现一个管程时必须考虑的 3 个主要问题包括互斥、同步、条件变量。(西安电子科大1996)
   15. 在采用请求分页式存储管理的系统中，地址变换过程可能会因为缺页、地址越界

和访问权限错等原因而产生中断。(中科大1998)

* 1. 存储管理应实现的功能是：主存空间的分配和保护、地址重定位、主存空间的共享和扩充。 (西安电子科大2002)
  2. 分区存储管理方案不能实现虚存的原因是作业地址空间大小不能超过存储空间大小。(西交大1999)
  3. 设有 8 页的逻辑空间，每页 1024 字下，它们被映射到 32 块的物理存储区中。那么，逻辑地址的有效位是 13 位，物理地址至少是 15 位。(西北工大2000)
  4. 采用交换技术获得的好处是以牺牲处理器时间为代价的。(西交大1999)
  5. 通道技术的引入，实现了处理器与设备的并行、设备与设备的并行、进程

与进程的并行。(中科院1998)

* 1. 设备管理中引入缓冲机制的主要原因是为了改善处理器与输入／输出设备之间速度不匹配的矛盾、实现输入／输出设备之间的并行操作和有效减少输入／输出操作次数。(中科院1999)
  2. 设备驱动程序是一种低级的系统例程，它通常分为驱动输入／输出设备工作程序 和设备中断处理程序两部分。(中科院1998)
  3. 在通道进行输入／输出操作期间，要访问两个内存的固定单元CAW和 CSW 。(西安理工2000)
  4. 利用符号链接实现文件共享时，对文件主删除了共享文件后造成的指针悬空问题，解决的方法是文件的拥有者才具有指向索引结点的指针，共享文件的其他用户中只有该文件的路径名而无指向索引结点指针。(中科院1999)
  5. 使用位示图 (20 行、 30 列) 表示空闲盘块状态。如当分配一个盘块号为 132 时，其在位示图中的行、列数为 4 、 11 。当释放一个盘块号为 318 时，其所在位示图中的行、列数为 10 、 17 。(注：行为 0~19 、列为 0~29 ，首盘块号为 1)(北京邮电大学1997)
  6. 每个索引文件都至少有一张索引表，其中，每一个表项应包括能标识该记录的记录键和物理地址。(西安电子科大2002)
  7. 文件系统中若文件的物理结构采用连续结构，则文件控制块中关于文件的物理位置应包括起始块号和总块数。(西北工大2000)
  8. 采用直接存取法存取文件时，对索引文件效率最高，对串联文件效率最低。
  9. 在 UNIX 文件管理系统中，为了对磁盘空间的空闲块进行有效管理，采用的方法是成组链接法。(西安电子科大2000)
  10. 在 UNIX 系统Ⅴ中，如果一个盘块的大小为 1KB，每个盘号占 4 个字节，那么，一个进程要访问偏移量为 263168B处的数据时，需要经过 1 次间接。(中科大1998)
  11. 在请求分页系统中，引用位标识 该页最近是否被访问过，它的用途是 为淘汰算法选页提供参考。(华中理工2001)
  12. 进程的目标代码在虚拟存储器中不考虑 物理空间大小和 信息存放的物理位置，只规定 进程中互相关联的信息的相对位置。(南开2001)
  13. 多用户环境中为了实现多用户之间的隔离，必须采取 存储保护 措施。(华中科技2001)
  14. 请求分页式虚存系统必须至少具有三种硬件支持，即 一定量内存和较大外存 、

地址转换机构及缺页中断机构。(中科院2000)

* 1. 在页面替换算法中，目前有多种方法被提出以防止抖动现象，它们的共同点是通过

程序的局部性原理 耒实现的。(中科院2000)

* 1. 虚拟设备是操作系统利用某种I/O技术，将某个 独占 设备改造为多个用户可以同时共享的设备。(北京理2002)
  2. I/O设备处理进程平时处于 睡眠 状态，当 I/O请求 和 I/O操作结束出现时，被唤醒。(西安电子科大2000)
  3. 文件目录中用 盘块号记录文件的一维地址，而实际读写磁盘需用 柱面号 ， 磁头号及 扇区号 三维地址。(南开大学2000)
  4. UNIX的文件目录项由两部分组成，即文件名和 索引节点 ；后者在文件被打开后又有两种形式，即 外存索引节点和 内存索引节点 。(中科院2000)
  5. UNIX能与其他操作系统组网是由于它支持 TCP/IP 协议。(中科院2000)
  6. 在一个请求分页存储管理系统中，某程序的页面走向为：7，0，1，2，0，3，0，4，2，3，0，3，2，1，2，0，1，7，0，1。假设分得的页框数是3，并且开始时页框中是空的，则分别采用最佳转换算法和LRU页面转换算法，在访问过程中发生缺页中断的次数分别是 9 和 13 。(南京航空2001)
  7. 一台计算机有十台磁带机被m个进程竞争，每个进程最多需要三台磁带机，那么m为 4 时，系统没有死锁的危险。(南京航空2001)
  8. 某系统中有11台打印机，N个进程共享打印机资源，每个进程要求3台。当N的取值不超过 5 时，系统不会发生死锁。(上海交大2002)
  9. 在分页虚拟存储管理中，“二次机会”调度策略和“时钟”调度策略在决定淘汰哪一页时，都用到了引用位 。 (上海交大2002)
  10. 虚拟存储器实现的基础是程序局部性原理理论。实现虚拟存储器是现代计算机操作系统 扩充主存 的主要方法。操作系统实现虚拟存储器需要 地址转换机构、 存储保护机构 和 中断机构 等软硬件支持。 (四川大学2000)
  11. 在微型计算机中，当I/O控制方式采用DMA时，数据传输的基本单位是 数据块 ，而且在外设和 内存 之间直接建立数据通路，在I/O过程中，不需要 CPU 的干预，完全由DMA控制器控制。(四川大学2000)
  12. 进程映像的三个组成部分是 程序块 、 数据块 、 PCB ，其中后者是进程存在的惟一标志。(合肥工大2000)
  13. 进程的基本状态有执行、就绪和等待。批处理作业的基本状态有运行、输入 后备 和完成。其中，运行 状态包含了进程的等待状态。作业调度程序实现了作业从 后备 状态到执行状态的转换，而进程调度 程序实现了进程从就绪状态到执行状态的转换。处于等待状态的进程是不会获得CPU的。(大连理工2002)
  14. SPOOLing技术是在共享设备上模拟独占设备的技术 。由预输入程序将作业执行中需要访问的数据预先读入输入井中。缓冲输出程序则负责将输出井中的信息在输出设备上输出。(大连理工2002)
  15. 文件在文件存储空间的组织方式，称为文件的 物理 结构。 (电子科大2001)
  16. 所谓操作系统虚拟机的概念，是指 OS屏蔽硬件细节，为用户提供使用方便、功能扩展、效率提高的计算机 。(华中科技2002)
  17. 程序并发执行时失去程序的封闭性和可再现性的主要原因是 运行程序相互制约 。(华中理工2001)
  18. 最佳适应算法是将作业放置到能满足作业长度空闲区中最小的一个空闲区中。(华中理工2000)
  19. 分区存储管理方案不能实现虚存的原因是 存储空间连续性要求 。(西交大1999)
  20. UNIX系统的核心结构由 进程控制 子系统和文件子系统两个部分组成的。(华中科技2001)

1. **术语/名词解释**

**1. 西北大学2000**

并行与并发、对换与切换、管道与通信、直接通信与间接通信、死锁与饥饿

(1)并发与并行；

若干个事件在同一时刻发生称为并行；若干个事件在同一时间间隔内发生称为并发。并行是并发的特例，并发是并行的拓展。

(2)对换与切换；

对换是指把内存中暂时不能运行的进程或暂时不用的程序和数据，换出到外存上，以腾出足够的内存空间，把已具备运行条件的进程或进程所需的程序和数据换入内存。切换是指将CPU的使用权从一个进程转到另一个进程。在某些系统中，进程切换往往伴随着信息的对换。

(3)管道与通道；

管道（Pipe）是连接两个进程的一个共享文件，进程通过对该文件的读、写实现进程间的通信。管道文件实际上是一个临时文件，它以磁盘为中介实现进程间的通信，与内存相比，其通信速度较慢。通道（I/O处理机）是实现I/O操作的硬件装置。通道对管道的实现提供子硬件支持。

(4)直接通信与间接通信；

消息系统有直接通信和间接通信之分。

1）直接通信。直接通信方式有一个基本原则：进程在发送和接收消息时，必须指明接收者或发送者的名字。这种通信方式中Send和Receive原语定义如下：

Send（P，message），将消息发送给进程P；

Receive（Q，message），接收来自进程Q的消息。

这种通信方式中通信链路具有如下特征：每一对欲通信的进程间自动建立了一条双向通信链，只需知道对方的标识信息便可进行通信；每条通信链路严格地对应两个进程；相互通信的一对进程之间存在一条通信链路。

2）间接通信。进程间通过信箱进行消息传递的通信方式称为间接通信，又称为“信箱通信”；信箱（Mailbox）可以抽象地看成是一个虚设备，进程可以把消息（也称信件）放入信箱，也可以从中取出一条消息。信箱必须有唯一的标识符。在这种通信方式中，某个进程可以通过一组不同的信箱同时与其他多个进程通信。两个进程之间只有当它们有一个可共享的信箱时才可进行通信。

间接通信方式中的通信链路具有如下特征：只有当两个进程有了一个可共享的信箱时，通信链路才在两者之间建立；一条通信链路可以连接两个以上的进程；每一对通信进程之间可以有多条不同的通信链路，每一条链路对应一个信箱；通信链路可以是单向的，也可以是双向的。

(5) 死锁与“饿死”。

死锁是因竞争资源而引起的一种具有普遍性的现象，在多道程序系统中，由于多个并发进程共享系统的资源，如使用不当有可能造成一种僵局，即系统中两个或多个进程无限期地等待永远不会发生的条件，在无外力的干预下，这些进程都不能向前推进，我们称之为死锁。死锁不仅在两个进程之间发生，也可能在多个进程之间，甚至在系统全部进程之间发生。当死锁发生时，一定有一个资源被无限期地占用而得不到释放。

“饿死”是指系统中的每个资源占用者都在有限的时间内释放它所占用的资源，但是仍然存在申请者永远得不到资源的现象。因此，在操作系统中，不仅要考虑如：何防止“死锁”，还要考虑如何避免“饿死”。

**2. 西安理工大学2000**

系统调用、多道程序设计、周转时间、碎片、系统抖动

(1)系统调用

系统调用是用户在程序中能用“访管指令”调用的由操作系统提供的子功能的集合。每一个子功能称为一条系统调用命令（或广义指令）。系统调用是操作系统在程序级给用户提供的接口。

(2)多道程序设计

多道程序设计是指在主存中同时存放多道用户作业，它们都处于执行的开始点和结束点之间。

(3)周转时间

所谓周转时间是指作业从进入系统到处理完成所经历的时间。

(4)碎片

所谓碎片是指存储器上不能利用的空闲区。

(5)系统抖动

在分页存储系统中，将某一页从实存移到辅存为“出页”，从辅存调入主存为“入页”。刚“出页”的页又要“入页”，或刚“入页”的页又要“出页”。这种反复出入页的现象称为“抖动现象”或者“系统颠簸”。

**3. 西安交通大学1999/2000**

线程、分时系统、系统调用、地址再定位、多道程序设计、地址空间和存储空间、通道

(1) 线程

所谓线程（thread），从操作系统管理角度看线程是指“进程的一个可调度实体”，是处理机调度的基本单位；从编程逻辑看线程是指“程序内部的一个单一的顺序控制流”。线程是进程的一个组成部分。

(2) 分时系统

所谓分时系统就是在一台计算机上，连接多个终端，用户通过各自的终端和终端命令把作业送人计算机，计算机又通过终端向各用户报告其作业的运行情况，这种计算机能分时轮流地为各终端用户服务并能及时对用户服务请求予以响应，这就构成了分时系统。分时系统设计的主要目标是使用户能与系统交互作用，对用户的请求及时响应，并在可能条件下尽量提高系统资源的利用率。分时系统的主要特征是：

1）同时性：若干个终端用户按照系统提供的各种服务，在各自终端进行操作，同时使用一台计算机资源。宏观上看是各用户在并行工作，微观上看是各用户轮流使用计算机。

2）独立性：用户间可以相互独立操作，互不干涉，系统保证各用户程序运行的完整性，不会发生相互混淆或破坏现象。

3）及时性：系统可对用户的输入及时作出响应。分时系统性能的主要指标之一是响应时间，它是指从终端发出命令到系统予以应答所需的时间。

4）交互性：用户可根据系统对请求的响应结果，进一步向系统提出新的请求，即能使用户和系统进行人 机对话的工作方式，所以分时系统也称之为交互式系统。

(3) 系统调用

系统调用是用户在程序中能用“访管指令”调用的由操作系统提供的子功能的集合。每一个子功能称为一条系统调用命令（或广义指令）。系统调用是操作系统在程序级给用户提供的接口。

(4) 地址再定位

所谓地址再定位，就是当一个程序装入到与其地址空间不一致的存储空间而进行的地址变换过程，即将地址空间给出的逻辑地址映射到内存的物理地址。地址重定位有静态重定位和动态重定位两种方式。

(5)多道程度设计

多道程序设计是指在主存中同时存放多道用户作业，它们都处于执行的开始点和结束点之间。多道程序设计的特点如下：

1）多道。主存中有多道程序，它们在任一时刻必须处于就绪、运行、阻塞三种状态之一。

2）宏观上并行。从宏观上看，它们在同时执行。

3）微观上串行。从微观上看，它们在交替、穿插地执行。

采用多道程序设计后，减少了CPU时间的浪费。尤其对计算题的作业，由于I/O操作较少，CPU浪费的时间很少。

(6) .地址空间和存储空间

地址空间指用户编程使用的全部地址，其中，每个地址单元又称逻辑地址，由于它常相对于程序的起始地址编号，故又叫相对地址。存储空间指内存中可直接寻址的物理单元的集合，物理单元又称物理地址或绝对地址。

(7) 通道

又称I/O处理器，它在接受CPU的I/O命令后，能相对独立地完成主存储器和外围设备之间的信息传送，并与中央处理器并行地执行操作，I/O结束后又能发点I/O中断请求CPU干予，具有上述一系到逻辑功能的部件就是通道。采用通道技术主要解决输入输出操作的独立性和各部件工作的并行性，减少外围设备和中央处理器的逻辑联系，实现外围设备和中央处理器之间的并行操作；通道和通道之间的并行操作；各通道上的外围设备之间的并行操作，以达到提高整个系统效率的目的。

**4. 清华大学2001**

多道程序、多重处理、进程、线程、虚存存储器

（1）多道程序 多个用户程序(作业)同时进入主存，并启动它们同时运行的程序设计技术。在单CPU上这些程序在宏观上是同时运行的，而微观上看它们交替执行。

（2）多重处理 系统配置多个CPU，能真正同时执行多道程序。要有效使用多重处理，必须采用多道程序设计技术，而多道程序设计并不—定要求多重处理系统支持。

（3） 进程 进程是一个可并发执行的具有独立功能的程序关于某个数据集合的一次执行过程，也是操作系统进行资源分配和保护的基本单位。

（4）线程 多线程环境中进程是操作系统中进行保护和资源分配的基本单位。线程是操作系统进程中能够独立执行的实体（控制流），是处理器调度和分派的基本单位。线程是进程的组成部分，每个进程内允许包含多个并发执行的控制流，这就是多线程。

（5）虚拟存储器 在具有层次结构存储器的计算机系统中，采用自动实现部分装入和部分对换功能，为用户提供一个比物理主存容量大得多的，可寻址的一种“主存储器”。

**5. 南京大学2001**

线程、目录、 死锁、窗口

(1)线程：线程又称轻量级进程，是指进程中的一条执行路径。

(2)目录：文件系统中的所有文件控制的有序集合称为目录。

(3)死锁：若一个进程集合中的每一个进程都在等待只能由本集合中的另一个进程才能引发的事件，则称这组进程处于死锁。

(4)窗口：把用户的工作显示在计算机屏幕的一个矩形区域中，允许用户在矩形区域中操作应用软件和文件，该矩形区域就称为“窗口”。

**6. 南京大学2003**

**写出操作系统中常用缩写字的英文全称：**

(1) RAID (2) LWP (3) IPC (4)IPT

（1）Reundant Array of Independent Disks

（2）Light Weight Process

（3）Interprocess Communication

（4）Inverted Page Table

**解释下列概念及说明关系或区别：**

(1)计算机系统的安全性、计算机系统的可靠性

(2)死锁、饥饿

(3)同步、互斥

(4)文件目录、目录文件

（1）计算机系统的安全性和可靠性是两个概念，可靠性指硬件系统正常持续运行的程度，目标为反故障；安全性是指不因人为疏漏和蓄谋作案而导致信息资源被泄露、篡改和破坏，目标是反泄露。

关系：可靠性是安全性的基础，安全性比可靠性更为复杂。

（2）死锁是因进程竞争资源，但系统拥有资源的数量有限，或并发进程推进的顺序不当而造成的一种永远等待资源的僵局。而饥饿是指每个资源占用者都在有限时间内释放占用的资源，但申请进程仍然长时间得不到资源的现象。

关系：产生饥饿现象并不意味着一定产生死锁。

（3）并发进程的执行会产生相互制约关系：一种是进程之间竞争使用独占型资源，只能让它们逐个使用，这种现象称互斥，是一种竞争关系，也称间接的制的关系。另一种是进程之间协同完成任务，在关键点上等待另一进程发来的消息，以便协同一致，是一种协作关系，也称直接的制的关系。

关系：本质上，互斥是一种特殊的同步，因为它也是进程之间的执行次序上的一种协调。

（4）文件目录记录文件的管理和说明信息，如文件名字、长度、属性、外存位置等信息，用于对单个文件的控制。把同一个卷上的若干文件的文件目录组成一个单独的文件，由于它是由文件目录组成的，故称目录文件，它用于对整个文件系统的管理。

**7. 南京航空航天大学2001**

多道程序设计、计算机操作系统、用户态与核心态、进程控制块、SPOOLing、逻辑文件和物理文件、进程映象、临界资源和临界区

(1)多道程序设计 见清华大学2001(1)/西安交大2000(5)

(2)计算机操作系统 操作系统是管理系统资源、控制程序执行、改善人机界面、提供各种服务，合理组织计算机工作流程和为用户有效使用计算机提供良好运行环境的一种系统软件。

(3) 用户态与核心态 硬件处理器的不同状态(或模式)，用于加强系统的安全性。多数系统把处理器状态简单的划分为核心态（又称特权状态、系统模式、特态或管态）和用户态（又称目标状态、用户模式、常态或目态）。当处理器处于管理状态时，程序可以执行全部机器指令，访问所有资源，并具有改变处理器状态的能力；当处理器处于用户状态时，程序只能执行非特权指令。

(4)进程控制块 标识进程存在和记录、刻画进程状态及有关信息的数据结构。它是操作系统掌握进程的唯一资料结构，是操作系统控制和管理进程的主要依据。它包括了进程执行时的情况，以及进程让出处理器后所处的状态、断点等的标识信息、现埸信息和控制信息。

(5)SPOOLing、 是外围设备同时联机操作的简称假脱机系统。其思路是：利用多道程序设计技术，在运行用户作业的同时，将大批新的作业信息从输入设备上预先输入到辅助存储器磁盘的输入缓冲区域中暂时保存，称为“预输入”。此后，由作业调度程序调出作业执行。作业使用数据时不必再启动输入设备，而只要从磁盘的输入缓冲区域中读入。类似地，作业执行中不必直接启动输出设备输出数据，而只要将作业的输出数据暂时保存到磁盘的输出缓冲区域中，在作业执行完毕后，由操作系统组织信息成批输出。称为“缓输出”。这样能带耒缩短作业执行时间、增加多道程序道数、加强诈作业调度灵活性的优点。Spooling技术是用一类物理设备模拟另一类物理设备技术，是使独占使用的设备变成可共享设备的技术，也是一种速度匹配技术。

(6)逻辑文件和物理文件 逻辑文件—是从用户观点出发，从方便使用的角度考虑文件信息的组织及配置方式，这种文件叫逻辑文件，它分为流式文件和记录式文件。物理文件---从系统观点出发，考虑文件在物理介质上的组织和存放方式，这种文件叫物理文件，它分串连文件、连续文件、索引文件和哈希文件。

(7)进程映象 UNIX SVR4中，进程由三部分组成：proc结构、数据段和正文段，它们合称为进程映像，而把进程定义为映像的执行。

(8) 临界资源和临界区 进程中涉及共享变量的程序段称临界区。临界区中共享变量代表的资源称临界资谅，这些资源一次仅能被一个进程使用。

**8. 复旦大学2000**

用户级线程和系统级线程

(1) 用户级线程ULT线程管理的全部工作都由应用程序来做，在用户空间内实现，内核是不知道线程的存在的。用户级多线程由用户空间运行的线程库来实现，任何应用程序均需通过线程库进行程序设计，再与线程库连接后运行来实现多线程。线程库是一个ULT管理的例行程序包，在这种情况下，线程库是线程的运行支撑环境。

当一个应用程序提交给系统后，系统为它建立一个由内核管理的进程，该进程在线程库环境下开始运行时，只有一个由线程库为进程建立的线程。首先，运行这个线程，当应用进程处于运行状态时，线程通过调用线程库中的“孵化”过程，可以孵化出运行在同一进程中的新线程。上述活动均发生在用户空间，且在单个进程中，内核并不知道这些活动。内核按进程为单位调度，并赋予一个进程状态（就绪、运行、阻塞…）。

(2) 系统级线程

内核级KLT线程设施中，线程管理的所有工作由操作系统内核来做。内核专门提供了一个KLT应用程序设计接口（API），供开发者使用，应用程序区不需要有线程管理的代码。Windows 2000/XP和OS/2都是采用这种方法的例子。

任何应用都可以被程序设计成多个线程，当提交给操作系统执行时，内核为它创建一个进程和一个线程，线程在执行中可以通过内核创建线程原语来创建其他线程，这个应用的所有线程均在一个进程中获得支持。内核要为整个进程及进程中的单个线程维护现场信息，所以，应在内核空间中建立和维护进程控制块PCB及线程控制块TCB，内核的调度是在线程的基础上进行的。

**9. 北京邮电大学1998**

进程、线程、管程

1. 进程和线程 见清华大学2001

(2) 管程 管程是管理进程同步的一种同步机制，它保证进程互斥地访问共享变置(代表共享资源)，并提供了一个阻塞和唤醒进程的设施----条件变量。它有三部分组成：(1)局部于管程的数据结构----共享变量，该共享变量表示了共享资源的状态。(2)局部于管程对上述数据结构进行规定操作的若干过程。(3)数据结构的初始化操作。

* + 1. **北京理工1999 /2000**

临界区、并发、不可剥夺资源、地址空间、工作集、抖动、程序局部性原理、设备独立性、虚拟设备

(1) 临界区 进程中涉及代表共享资源的共享变量的程序段称临界区。

(2) 并发 是指两个或两个以上的事件或活动在同一时间间隔内发生。操作系统是一个并发系统，操作系统的并发性指计算机系统中同时存在若干个运行着的程序，因此，它应该具有处理和调度多个程序同时执行的能力。

(3)不可剥夺资源 除非拥有该资源的进程主动放弃，系统不得强行取回的一类资源。

(4) 地址空间 地址空间指用户编程使用的全部地址，其中，每个地址单元又称逻辑地址，由于它常相对于程序的起始地址编号，故又叫相对地址。存储空间指内存中可直接寻址的物理单元的集合，物理单元又称物理地址或绝对地址。

(5) 工作集 Denning提出的工作集概念是：“为确保每个进程每一时刻能够执行下去，在物理存储器中必须有的最少页面数”。但也有的文献所用工作集概念稍有不同是指：“在未来的时间间隔内，一个进程运行时所需访问的页面集”。

(6) 抖动 页面淘汰中的一种现象，刚被淘汰的页面又立即要用，因而，又要把它调入，而调入不久再被淘汰，淘汰不久再被调入。如此反复，使得整个系统的页面调度非常频繁以至于大部时间都花在来回调度页面上。处理器花费大量时间用于对换页面而不是执行计算任务的现象叫做“抖动”（Thrashing），又称“颠簸”。

(7) 程序局部性原理 发现的程序和数据的访问都有聚集成群的倾向，在一个时间段内，仅使用其中一小部分（称空间局部性），或者最近访问过的程序代码和数据，很快又被访问（称时间局部性）。

(8) 设备独立性、虚拟设备 见相关词条。

**11. 武汉大学1999**

原语、特权指令、顺序进程与并发进程、死锁与饥饿

(1)原语 是在管态下执行、完成操作系统特定功能的过程。原语和机器指令类似，其特点是执行过程中不允许被中断，是一个不可分割的基本单位，原语的执行是顺序的而不可能是并发的。

(2)特权指令 仅当处理器运行在核心态时，程序中才能正常执行的机器指令。

(3)顺序进程与并发进程 前者指在处理器上独占全部资源，按严格顺序执行指令的进程，其程序执行的结果与它的执行速度无关，且初始环境一定时，执行结果可再现。后者指的执行在时间上是可重迭的一组进程，由于打破了封闭性和可再现性，并发执行中进程的执行具有间断性。

(4)锁与饥饿 见问答题1。

1. **大连理工1999**

CPU利用率、临界区、重定位、抖动、死锁、虚拟设备

见有关条目。

**13. 北京工业大2000**

死锁、管态、目态、PCB、低级调度、响应比、进程、临界区

(1) 死锁 见问答题1。

(2) 管态、目态 、PCB 见南京航空2001。

(3) 低级调度 负责动态地按一定算法把处理器分配给进程或内核级线程的过程称低级调度。操作系统中实现低级调度的程序称为低级调度程序，或分派程序（dispatcher）。

(4) 响应比 作业进入系统后的等待时间与估计计算时间之和称为作业的响应时间，作业的响应时间除以作业估计计算时间称作响应比，定义为：响应比＝作业响应时间/作业估计计算时间=1+作业等待时间/作业估计计算时间

(5) 进程、临界区 见有关词条。

**14. 哈工大2000**

进程、线程、临界区、同步机构

1. 进程、线程、临界区 见有关词条
2. 同步机构 操作系统实现进程同步的机构称同步机构或同步机制，它由同步原语组成。不同的同步机制采用不同的同步方法，迄今己设计出许多种同步机制，最常用的同步机制有：信号量及PV操作，管程和消息传递。

**15. 南开大学1999 /2000**

抖动、进程调度、虚拟设备、SPOOLing、动态地址变换、磁盘缓冲区、中断及异常

(1) 抖动、进程调度、虚拟设备、SPOOLing 见有关词条。

(2) 动态地址变换 为保证程序的正确运行，必须把程序和数据的逻辑地址转换为物理地址，这一工作称为地址变换或重定位。如果是在程序执行过程中，CPU访问程序和数据之前实现地址转换，称为动态重定位，动态重定位必须借助于硬件的地址转换机构实现。

(3) 磁盘缓冲区 开劈的用于磁盘I/O暂存信息的内存缓冲区，使用它的优点是匹配CPU和磁盘速度的差异，减少内外交换次数。

(4) 中断及异常 中断是指来自处理器和主存储器之外的中断信号引起的中断，又叫外中断。包括：电源故障中断、时钟中断、控制台中断、它机中断和I/O中断等。每个不同的中断具有不同的中断优先级，在处理高一级中断时，往往会屏蔽部分或全部低级中断。异常是指来自处理器和主存内部的中断信号引起的中断，又叫内中断。包括：通路校验错、主存奇偶错、非法操作码、地址越界、页面失效、调试指令、访管中断、算术操作溢出等各种程序性中断。其中访管中断是由机器指令提供的特殊指令，该指令执行时将会引起内中断。异常是不能被屏蔽的，一旦出现应立即响应并加以处理。

**16. 四川大学2000**

临界资源 、原语 、重定位 、驱动调度 、 FCB

(1) 临界资源 、原语 、重定位 见有关词条。

(2) 驱动调度 作为操作系统的辅助存储器，用来存放文件的磁盘一类高速大容量旋转型存储设备，在繁重的输入输出负载之下，同时会有若干个输入输出请求来到并等待处理。系统必须采用一种调度策略，使能按最佳次序执行要求访问的诸请求，这就叫驱动调度，使用的算法叫驱动调度算法。驱动调度能减少为若干个输入输出请求服务所需的总时间，从而，提高系统效率。

(3) FCB 文件控制块--文件系统依靠文件目录实现文件的“按名存取”。每一个文件在文件目录中登记一项，实质上文件目录是文件系统建立和维护的它所包含的文件的清单，每个文件的文件目录项又称文件控制块FCB(File Control Block)，一般应该包括以下内容：

1) 有关文件存取控制的信息：如文件名、用户名、文件主存取权限、授权者存取权限：文件类型和文件属性，如读写文件、执行文件、只读文件等。

2) 有关文件结构的信息：文件的逻辑结构，如记录类型、记录个数、记录长度、成组因子数等。文件的物理结构，如文件所在设备名，文件物理结构类型，记录存放在外存的相对位置或文件第一块的物理块号，也可指出文件索引的所在位置等。

3) 有关文件使用的信息：已打开该文件的进程数，文件被修改的情况，文件最大和当前大小等。

4) 有关文件管理的信息：如文件建立日期、文件最近修改日期、文件访问日期、文件保留期限、记帐信息等。

**17. 中山大学2001**

临界区、挂起、快表

(1) 临界区 见有关词条。

(2) 挂起 为达到平滑系统操作负荷，或满足用户程序调试等目的，而新引入的一种进程状态称”挂起”态。被挂起的进程，对换到磁盘镜像区中，释放它所占有的某些资源，不难看出，可以把一个挂起进程等同于不在主存的进程，因此，挂起的进程将不参与低级调度直到系统资源充裕厉它们被对换进主存。

(3) 快表 存分页式存储管理中，为了提高运算速度，通常都在MMU中设置一个专用的高速缓冲存储器，用来存放最近访问的部分页表，这种高速存储器称为相联存储器，也称TLB（Translation Lookaside Buffer），它成为分页式存储管理的一个重要组成部分。存放在相联存储器中的页表称快表。

1. **问答题**

1. 简述死锁与’饥饿’。(西北工大2000)

答：如果在一个进程集合中的每个进程都在等待只能由该集合中的其他一个进程才能引发的事件，则称一组进程或系统此时发生了死锁。产生死锁的原因是因诸进程竞争资源或进程推进顺序不当而造成’。饥饿’也是进程间竞争资源产生的一种状况，一个进程由于其他进程总是优先于它而被无限期拖延获得资源或接受服务，陷入了’饥饿’状态，但此时系统并未发生死锁。

2.剥夺式与非剥夺式优先级调度算法各有什么优缺点?设计一种调度算法，使该调度算法能保留以上两种算法的各自优点并克服它们的缺点。(西北工大2000)

答：非剥夺式优先级调度算法优点是实现简单、系统开销小、貌似公正，但可能使系统性能恶化。如后到的紧急任务不能立即投入远行，使紧急任务的响应时间延长；后到的多个短进程必须等长进程运行完毕，使得周转时间增加而系统吞吐率降低。剥夺式剥夺式优先级调度算法能保证高优先级进程立即投入运行，减少进程平均周转时间，提高系统吞吐率，但系统开销较大，系统的复杂性也随之增加。

采用多级反馈队列调度算法能结合两种算法的各自优点。算法的基本思路如下：设置多个就绪队列，并赋予各队列不同的优先权。就绪队列的设置按时间片大小划分，优先权越高的队列，其进程运行时获得的时间片越小。系统总是把新创建的进程首先排入优先级最高的队列，若它执行一个时间片后尚未完成，系统便把它放入下一级队伍的末尾、即进程的优先级动态地逐步降低。非最低优先级队列均采用时间片轮转的FCFS调度算法，而最低优先级队列可采用轮转法或其他调度算法。

3.一个分层结构的操作系统由裸机、用户、CPU调度和PV操作、文件管理、作业管理、内存管理、设备管理、命令管理等部分组成。试按层次结构原则从内到外将各部分重新排列。(中科院1997)

答：试按层次结构原则从内到外依次为：裸机、CPU调度和PV操作、内存管理、设备管理、文件管理、作业管理、命令管理、用户。

1. 计算机系统采用通道部件后，己能实现CPU与外设的并行工作，为什么进要引入多道程序设计?(北大1996)

答：—是机器部件能并行工作仅仅有了提高效率的可能性，而机器部件并行工作的实现还需要软件技术去利用和发挥，这就是要引入多道程序设计技术的原因之一。二是多个程序同时进入主存工作时，它们将共享系统的资源，会发生直接或间接的制约关系，即程序不再在封闭环境中运行，会出现程序执行顺序、完成时间及运行结果的不确定性，必须要引入多道程序设计技术来解决这个问题。

5. 关于处理机调度，试问：(1)什么是处理机三级调度?(2)处理机三级调度分别在什么情况下发生?(3)各级调度分别完成什么工作?

答：处理机三级调度发生的情况是：（1）高级调度。高级调度是根据系统内所有资源的使用情况，一旦可能便从后备作业中选择一道作业进入系统，并创建相应的进程，分配必要的系统资源，然后，将进程“就绪”。（2）低级调度。低级调度即为CPU调度，它是根据CPU资源的使用情况及时分配CPU。即从“就绪”的进程中选择一个进程在CPU上“运行”。

（3）中级调度。在内存中常常有许多进程处于某种等待状态，这些进程在“等待”期间无谓地占用着内存资源。如将它们暂时换至外存，则所节省出来的内存空间可用以接纳新的进程，一旦换出外存的进程，具备运行条件时再将其重新换入内存。为此，在逻辑上将主存延伸，用一部分外存空间（称为交换区）替代主存，并且实施交换调度（中级调度）。

高级调度完成作业调度，使“后备”状态的作业变为“执行”状态；中级调度完成内存和外存信息的交换调度；低级调度完成进程调度，使“就绪”的进程在CPU上“运行”。

6. 系统调用是操作系统和用户程序的接口，库函数也是操作系统和用户程序的接口，这句话对么?(清华1995)

答：不对。系统调用是操作系统为了扩充机器功能、增强系统能力、方便用户使用而在内核中建立的系统程序(过程)，通常运行在系统态。用户程序或其他系统程序通过系统调用就可以访问系统资源，调用操作系统功能，它是用户程序或其他系统程序获得操作系统服务的唯一途径。库函数是程序设计语言中，把常用的功能编写成函数。放在函数库中供用户编程时使用。库函数仍属用户程序而非系统程序，运行时仍在用户态而非系统态。

7. 现有两道作业同时执行，一道以计算为主，另一道以输入输出为主，你将怎样赋予作业进程占有处理器的优先级?为什么?(南大2000)

答：优先级调度的基本原则是：照顾时间紧迫的作业使其得到及时处理、照顾会话型作业使其快速响应、照顾I/O繁忙型作业，使外设充分保持忙碌。故I/O为主的作业对应的进程优先级高，而以计算为主的作业对应的进程优先级低。

1. 操作系统中为什么要引入进程概念?为了实现并发进程间的合作和协调工作，以及保证系统的安全性，系统在进程管理方面应做哪些工作?(南大1997)

答：在多道程序环境下，程序可以并发执行，一个程序的任意两条指令之间都可能发生随机事件而引发程序切换。因而，每个程序的执行都可能不是连续的。此外，程序的并发执行又引起了资源共享和竞争的问题，造成了各并发执行的程序间可能存在制约关系，程序和计算不再一一对应。系统需要一个既能描述程序动态执行过程，又能用来共享资源的一个单位，操作系统引入的这个单位就是进程。

操作系统在进程管理方面应做：进程控制、进程调度、进程同步与互斥、进程通信等工作。

9.试比较进程和程序的区别。(哈工大2000)

答：进程是程序在数据集合上的一次执行过程，是动态概念，它有生命周期，由创建而产生，由调度而执行，由撤销而消亡；而程序是一组有序指令序列，是静态概念，程序作为一种系统资源是永久存在的。进程和程序之间无一一对应关系，不同进程可对应同一程序，同一程序运行中也可产生不同进程。进程是系统进行资源分配与运行调度的独立单位，程序无此概念。

10. 进程和线程的主要区别是什么?(西北工大1999)

答：(1)线程是进程的组成部分，一个进程可包含有多个线程 ； (2)进程是资源分配与保护的基本单位，拥有自己的地址空间和各种资源。线程是处理器调度的基本单位，自己不具有资源，只能和其他线程共享进程的资源；(3)以进程为单位进行处理器调度和切换和进程通信时，涉及改变地址空间、资源转移及现埸保护等问题，时空开销大，系统效率低；但进程内的多线程均在同一地址空间活动，因而，线程切换、通信效率高，共享数据也很方便；(4)进程的调度和控制大多由操作系统的内核完成，而线程控制既可由操作系统的内核完成，也可由用户程序控制完成。

11.进程之间存在哪几种相互制约关系?各是什么原因引起的?下列活动分别属于哪种制约关系?(1)若干同学去图书馆借书；(2)两队举行篮球赛；(3)流水线生产的各道工序；(4)商品生产和社会消费。(北理工1996)

答：进程之间相互制约关系分同步(直接制约关系)和互斥(间接制约关系)。前者是因合作进程之间协调彼此工作而控制自己的执行速度，即因进程间相互合作而等待产生的制约关系。后者是因进程间竞争临界资源而禁止两个以上进程同时进入临界区所发生的制约关系。(1)互斥；(2) 互斥；(2) 同步；(4) 同步。

12.Dijkstra1965年提出的银行家算法其主要思想是什么?它能够用来解决实际中的死锁问题吗?为什么?(中科院1996)

答：银行家算法是解决死锁的一种策略，其主要思想是：在每次实施资源分配前，先进行试探性分配，以确保系统始终处于安全状态，即查找试探性分配中是否存在某种进程执行顺序，按此顺序进行资源分配和回收使全部进程能正常运行结束，如果是则正常分配，否则拒绝本次资源分配。银行家行算法虽有很好理论意义，但该算法要预先获得有关信息，如进程所需的最大资源量，系统中进程的总数等，这是难以做到的，所以，银行家算法缺乏实用价值。

13. 按序分配是防止死锁的一种策略。什么是按序分配?为什么按序分配可以防止死锁?(南大2000)

答：按序分配策略把系统的所有资源安排一个顺序，按顺序给每个资源一个编号，规定每个进程申请两个以上资源时，总是先申请编号小的再申请编号大的资源。这样，在进程集合中总存在某个进程，它占有了己申请资源中编号最大的资源，因而，它不再能申请其他资源，当它运行结束，就可以释放占用的全部资源。剩余的进程集合中又会有一个进程此时占有己申请资源中编号最大的资源，那么，它也能运行结束。以此类推，最终所有进程都能运行结束，故系统不会发生死锁。实质上，按序分配通过破坏死锁的循环等待条件而防止死锁。

14. 什么是进程控制块?试从进程管理、进程通信、中断处理、文件管理、存储管理、设备管理的角度设计进程控制块应包含的项目。(北大1999)

答：进程控制块PCB是操作系统为每个进程建立的，用于记录和刻画进程状态及有关信息的数据结构，也是操作系统掌握进程的唯一资料结构，是操作系统控制和管理进程的主要依据。PCB是进程存在的唯—标志，系统根据PCB而感知进程的存在，创建进程时为其分配PCB，撤销进程时回收PUB。

从进程管理角度，应有：进程标识、进程状态、进程优先级、队列指针等。

从进程通信角度，应有：消息队列首指针、访问消息队列互斥信号量、消息计数等。

从中断处理角度，应有：现埸信息(上下文)、中断源及类型等。

从文件管理角度，应有：保存进程使用文件的文件控制块(FCB)等。

从存储管理角度，应有：保存进程使用的程序和数据的内外存地址或页表位置等。

从设备管理角度，应有：保存进程分配到的资源及所需资源情况等。

1. 有一OS以如下方法定义P、V掠操作：

P(S)： S:=S-1;

If S<0 then 本进程进入相应队列尾等待；

V(S): S:=S+1;

If S=0 then 唤醒相应队列尾进程，并进就绪队列;

这与传统的P、V操作的定义有什么缺点?

答：与传统的P、V操作的定义相比缺点为：

(1)V操作中，仅当S=0、即队列中只有一个进程阻塞时才唤醒，否则不预唤醒，这是错误的。(2)V操作唤醒的是队尾进程，不是先进入等待的进程，会造成饥饿现象。(3)传统的V操作在唤醒阻塞进程后，执行进程将继续运行。而此处却转入就绪队列，在执行进程能够继续执行的情况下无故使其进入就绪队列的做法只会增加CPU调度的开销，降低系统效率。

1. 分页存储管理有效地解决了什么问题?试叙述其实现原理。(中科院1995)

答：分页存储管理有效地解决了存储器碎片(零头)问题。其实现原理如下：(1)以同样大小的块为单位划分存储器，块(页框)是内存分配的单位，其大小为2的幂次。(2) 以同样大小的页(页面)划分作业地址空间，页与块的大小相等。(3)作业的页面在存储器中可存放在不连续的页框中，并建立页表记录每个页面所对应的页框。(4)逻辑地址由页面号与页内位移组成，通过地址变换机构动态地实现作业的逻辑地址到内存的物理地址的转换。

1. 什么是虚拟存储器?举—例说明操作系统是如何实现虚拟内存的。(西交大1999)

答：为了充分利用主存空间，也为用户大作业提供方便(作业地址空间可超过主存可用空间)，由操作系统把主存与辅存统一管理和使用，自动实现部分装入和部分对换功能，即用户大作业在执行时，—部分用户信息放在主存，而其他部分信息放在辅存，当访问信息不在主存时，由系统将其从辅存调入主存。从效果上看，为用户提供了一个比物理主存容量大得多的，可寻址的一种“主存储器”，称为虚拟存储器。

例如，请求分页存储管理系统实现了一种虚拟存储器。其实现要点是：为每个作业创建一张页表，仅装入作业当前需要的部分页面到主存，其他存放在辅存上，并在页表中登记所有页面的情况。当作业运行访问到不在主存的页面时，通过产生的“缺页”中断，由系统把所需页面调入主存。在处理缺页中断过程中，如果主存中己没有或很少有空闲页框，则还涉及到页页的淘汰问题。从逻辑地址到物理地址的变换，可借助于页表，由动态地址重定位机构完成。

1. 在虚拟页式存储系统中引入了缺页中断：(1)试说明为什么引入缺页中断?(2) 缺页中断的实现由哪几部分组成?并分别给出其实现方法。(西交大2000)

答：(1)由于虚拟页式存储系统中，作业在执行时并不把全部页面放入主存，那么当访问页面不在主存时，可产生中断并由系统响应中断将该缺页由辅存调入主存，然后恢复中断程序并继续执行。故对虚拟页式存储系统必须引入缺页中断。

(2) 缺页中断由硬件与软件两部分合作完成：1)硬件 CPU执行一条指令时首先获得逻辑地址，然后计算页号，查看页表该页是否在主存中，如果该页不在主存中，则引发缺页中断并转入缺页中断处理程序。2)软件 首先查看主存分块表，是否有空闲页框?如果没有则根据淘汰算法选择主存中一页淘汰，并修改主存分块表，这时有了空闲页框。在有空闲页框的情况下，系统根据页表中提供的该页辅存地址把页面调入主存的一个空闲块中，修改主存分块表和页表，再重新执行刚才被中断的指令。

1. LRU算法的基本思想是什么?有什么特点?给出该算法的流程图。(中科院1996)(西北大学1998)

答：LRU算法的基本思想：根据程序的局部性原理，依照一个作业在执行过程中己执行过的页面访问踪迹来推测来的页面走向、即认为过去一段时间里不曾被访问过的页面，在最近的将来也不会再被访问。故LRU算法选择在最近一段时间最久不用的页面予以淘汰。

LRU算法适应性强，但实现困难，因为要不断对前面访问过的页面的历史加以记录和更新，由硬件实现则成本加大，由软件实现则开销可观。常用的LRU近似算法如图，退化为“最近不用”算法NRU。

NRU算法

查找指针指向页表中下一个表项

页面访问位=0

置页面访问位为0

选择该页面淘汰

选择该页面淘汰

返回

y

N

20. 在请求分页虚存管理系统中，若驻留集为m个页框，页框初始为空，在长为p的引用串中具有n个不同页面(n>m)，对于FIFO、LRU两种页面替换算法，试给出缺页中断的上限和下限，并举例说明。(国防科大2000)

**答：**对于FIFO、LRU两种页面替换算法，缺页中断的上限和下限：为p和n。因为有n个不同页面，无论怎样安排，不同页面进入内存至少要产生一次缺页中断，故下限为n次。由于m<n，引用串中有些页可能进入内存后又被调出，而多次发生缺页中断。极端情况，访问的页都不在内存，这样共发生了p次缺页中断。例如，当m=3，p=12，n=4时，有如下访问中：1，1，1，2，2，3，3，3，4，4，4，4。缺页中断为下限4次。而访问串：2，3，4，1，2，3，4，1，2，3，4，1。缺页中断为上限12次。

1. 有两台互联的计算机，现需要把这两台计算机的主存储器以一个统一的逻辑地址空间呈现在应用程序面前。那么，在通信系统支持下，需要什么样的硬件支持?操作系统应如何实现逻辑地址到物理地址的变换?(图示并配必要说明)(电子科大1996)

答：在段表中增加一个标设位以表示是否存在于本机主存，同时，硬件应能产生缺段中断。在通信系统支持下，操作系统实现逻辑地址到物理地址的变换过程如图。

控制寄存器

段表长度 段表始址

段号 位移

逻辑地址

标识 在/不在主存 段长 主存始址

在本机 在

在它机 指针

作业段表

+

物理地址

本机主存

通过通信网络指向它机作业段表

1. 某系统采用页式存储管理，采用LRU算法淘汰页面，(1)写出LRU的基本实现方法和原理。(2)如果系统发生“抖动”应如何解决?

答：(1)LRU淘汰最近最少使用的页面，方法之一是可在页表项中为每页增加一个计数器，每当页面被访问时，页对应的页表项中的计数器加1，当发生缺页中断淘汰计数器值最小的那一只，同时将所有计数器清0。(2) 系统发生“抖动”时应采用增加工作集的方法，可把缺页进程锁住不让其换出，而调入的页总是占据那些暂时不执行进程所占用的内存页面，从而扩大缺页进程的工作集。

1. 简述LRU、NRU和LFU这3种页面置换算法的思想，并各给出一种可能的实现方案。(中科大1998)

答：LRU算法利用“最近的过去”作为“最近的将来”的一种近似，选择最近最久未使用过的页面予以淘汰。实现方法：为每个页表的每个页面保留一个访问字段，用来记录该页面自上次被访问以来所经历的时间T，当需要淘汰一个页面时，总是选择现有页面面中T值最大的页面淘汰。

NRU算法总是选择在最近一个时期内未被访问过的页面予以淘汰。实现方法：为页表的每个页面保留一个访问位，当某页被访问时其访问位置1，系统周期性地对所有访问位清0。当需要淘汰一个页面时，总是从访问位为0的页面中选择一个予以淘汰。

LFU算法总是选择在最近时期使用最少的页面予以淘汰。实现方法：为每个页面增设一个访问计数器，每当页面被访问时其访问计数器加1。当需要淘汰一个页面时，总是淘汰计数器值最小的页面，同时，所有计数器清0。

1. 解决大作业和小内存矛盾有哪些途径?简述其实现思想。(上交大1997)

答：覆盖技术：由用户把一个程序划分为若干个功能相对独立的程序段，并根据程序的逻辑结构让不会同时执行的程序段共享同一块内存区(称覆盖区)。程序执行中再依需要把程序段调入覆盖区，由此达到小内存运行大作业的目的。

虚存技术：操作系统把主存与辅存统一管理和使用，自动实现部分装入和部分对换功能，即用户大作业在执行时，—部分用户信息放在主存，而其他部分信息放在辅存，当访问信息不在主存时，由系统将其从辅存调入主存。由此，能达到小内存运行大作业的目的，而且可以还可以在小内存中运行多道大作业。

1. 为什么要在设备管理中引入缓冲技术?操作系统如何实现缓冲技术?(国防科大2001)

答：(1)调节CPU和I/O设备之间速度不匹配的矛盾 例如，如果不设缓冲，则程序输出时由于打印机速度跟不上而使CPU停下来等待，而在CPU计算时，打印机又因无数据输出而闲置。有了缓冲区，则程序可把输出数据预先输到缓冲区后继续运行，而打印机可从缓冲区取数慢慢打印，从而，CPU和I/O设备之间速度不匹配的矛盾得到缓和。

(2)实现I/O设备之间的并行操作 类似地，可以开出多缓冲，每个对应于一个设备，实现I/O设备和I/O设备之间的并行操作

(3)减少内外(I/O)交换次数 开设缓冲区后可以实现成组和分解操作，既减少了内外(I/O)交换次数，又充分利用了外存空间。同时，减少内外(I/O)交换次数，也减少了CPU处理I/O中断的次数，提高了系统效率。

缓冲区是临界资源，OS要管理缓冲区的申请、释放和互斥问题。例如，可设缓冲池，并分成空闲缓冲区、输入缓冲区、输出缓冲区。当输入设备需要输入数据时，从空闲缓冲队列取一个空缓冲区，待装满数据后，将其插入输入队列。当CPU处理输入数据时，就从输入队列取下一个数据缓冲区进行处理，处理完该缓冲区数据后将其插入空闲缓冲区队列。当CPU进行数据输出时，也作类似处理。

1. 什么是DMA方式?它与中断方式的主要区别是什么?()

答：DMA又称直接内存存取方式，其特点是外设在硬件支持下直接与内存交换成批数据而不需CPU干预。DMA方式下，地址总线、数据总线及相关控制信号线均与CPU共用。平时被CPU管理使用，当DMA时，乐用偷窃总线控制权的办法，CPU管理的总线由DMA控制器接管，并由DMA控制器控制外设与内存间成批交换数据。每当DMA传送的数据完成后发出一个中断，由CPU响应中断并回收控制权。

DMA中断方式的主要区别是：(1)中断方式在每个数据传送完成后中断CPU，而DMA是在所要求的一批数据全部传送完毕才中断CPU。(2) 中断方式的数据传送是在中断处理时由CPU控制完成，而DMA方式则是在DMA控制器控制下完成。

1. 文件系统必须完成哪些工作?(厦门大学1999)

答：(1)文件目录管理

(2)文件的组织

(3)文件的存取

(4)文件存储空间管理

(5)文件的共享、保护、保密

(6)提供各种文件操作

28. 文件存储器的管理与内存管理有何异同?(西北大学1998)

答：文件存储器的管理与内存管理异同点：

1. 主要任务 内存管理的主要任务为多道程序设计提供支撑；文件存储管理的主要任务为文件系统提供和管理文件空间。
2. 基本功能 内存管理的基本功能为内存的分配、回收、共享、保护和扩充；文件存储管理的基本功能包括文件存储空间的分配和回收。
3. 分配方式 都可采用连续分配，但以离散分配方式为主。前者常用算法有：最先适应、最佳适应和最坏适应算法等；两者采用离散分配方式时，采用机制和数据结构不尽相同，内存管理常采用页表、段表；而外存管理采用文件分配表、位示图等。
4. 分配单位 内存以字节、页面(若干字节) 或段(若干字节)为分配单位，长度变化大；外存都以盘块或簇(若干连续盘块)为分配单位，长度固定。

29. 现代操作系统一般都提供多进程(或称多任务)运行环境，回答下列问题：(华中科大2000)

* 1. 为支持多进程并发运行，系统必须建立哪些关于进程的数据结构?
  2. 为支持进程状态变迁，至少应提供哪些进程控制原语?
  3. 执行每一个进程控制原语时，进程状态发生什么变化?相应数据结构发生什么变化?

答：(1)系统为每个进程建立一个数据结构-----进程控制块PCB，用于进程的控制和管理。PCB中记录了有关进程的标识、调度信息、文件信息、存储信息、现埸信息、设备及各种资源信息，以便系统控制和管理进程。PCB是进程存在的唯一标志，也是系统掌握的进程的帷一资料结构。

(2)进程生命周期中，会经历多种状态。系统应具有创建、撤销进程、实现进程状态转换等功能，为此操作系统内核中提供一组进程控制原语，至少包括：创建进程、撤销进程、阻塞进程、唤醒进程、终止进程原语等。用户可通过系统调用接口耒使用它们。

(3) 创建进程、撤销进程、阻塞进程、唤醒进程、终止进程原语的功能简述略。

30.何谓临界资源?使用临界资源诸进程如何实现进程同步?(北京邮电大学2000)

答：一次仅允许一个进程使用的资源称临界资源，进程中对临界区访问的程序段称临界区或临界段。对临界资源的使用必须互斥，进程进入临界区必须满足临界区调度的基本原则，实现进程互斥的方法有专门硬件指令(如test and set或swap)、信号量与PV操作、管程等方法。

31.何谓管程?它由哪几部分组成?说明引入管程的必要性。(北京邮电大学2000)

答：管程是管理进程同步的一种同步机制，它保证进程互斥地访问共享变置(代表共享资源)，并提供了一个阻塞和唤醒进程的设施----条件变量。它有三部分组成：(1)局部于管程的数据结构----共享变量，该共享变量表示了共享资源的状态。(2)局部于管程对上述数据结构进行规定操作的若干过程。(3)数据结构的初始化操作。

在引入管程前，对临界区的访问都分散在各个进程中，不但加重用户编程负担，而且容易造成程序设计错误(如用户不正确使用PV操作)。管程把分散在各进程中的临界区集中起来加以控制和管理，每次仅允许一个进程进入管程内，既便于系统管理共享资源，又能保证进程间的互斥，还能方便地利用高级程序设计语言编写程序。

32.现有操作系统对进程的定义不尽相同，有的还引入了挂起状态。试简要分析挂起状态的意义。(浙江大学2000)

答：在一些操作系统中，为了更好地管理和调度进程，以适应系统的功能目标，提高系统的整体效率，引入了挂起状态：

1. 系统出现故障或功能模块受到破坏，可暂时挂起系统进程，以便修复或消除故障后再把进程转换到原来状态。
2. 用户检查作业执行情况或中间结果，可请求系统挂起该作业进程。
3. 系统负荷过重，如进程数量过多、资源相对不足，造成系统性能下降。此时，需要挂起部分进程以调整系统负荷，待系统负荷较轻时再恢复这些进程运行。
4. 在虚存管理系统中，需要区分进程是驻留在内存还是外存，可用挂起状态表示进程驻留在外存。

33.从进程的定义可看出，它有哪几个基本特征?进程是一个可并发执行的具有独立功能的程序关于某个数据集合的一次执行过程，是操作系统进行资源分配和保护的基本单位。(哈工大2002)

答：它具有如下属性：

结构性：进程包含了数据集合和运行于其上的程序，为了描述和记录进程的动态变化过程使其能正确运行，还需配置一个进程控制块，所以，每个进程至少有三要素组成：程序块、数据块和进程控制块。

共享性：同一程序同时运行于不同数据集合上时，构成不同的进程。或者说，多个不同的进程可以共享相同的程序，所以，进程和程序不是一一对应的。

动态性：进程是程序在数据集合上的一次执行过程，是动态概念，同时，它还有生命周期，由创建而产生，由调度而执行，由撤销而消亡；而程序是一组有序指令序列，是静态概念，所以，程序作为一种系统资源是永久存在的。

独立性：进程既是系统中资源分配和保护的基本单位，也是系统调度的独立单位（单线程进程）。凡是未建立进程的程序，都不能作为独立单位参与运行。通常，每个进程都可以各自独立的速度在CPU上推进。

制约性：并发进程之间存在着制约关系，进程在进行的关键点上需要相互等待或互通消息，以保证程序执行的可再现性和计算结果的唯一性。

并发性：进程可以并发地执行，进程的执行是可以被打断的，或者说，进程执行完一条指令后在执行下一条指令前，可能被迫让出处理器，由其他若干个进程执行若干条指令后才能再次获得处理器而执行。进程的并发性能改进资源利用率和提高系统效率。

34. 画出作业和进程状态转换图。(厦门大了2001)

答：

执行

状态

运行

就绪

等待

作

业

输

入

状

态

作

业

后

备

状

态

作

业

完

成

状

态

进程调度

中级调度

缓输出

作业调度

(创建用户进程)

预输入完成

作业调度与低级调度关系及作业和进程状态转换

作业运行结束

(撤销用户进程)

1. 什么叫“可再入”程序?它有什么特性?(西安电子科大2000)

答：可被多个进程同时调用的程序称“可再入”程序。它必定是纯代码，即执行过程中自身不会改变，故调用它的进程应该提供数据与工作区。

1. 分页存储管理中，页表的功能是什么?当系统中的地址空间变得非常大时(如32位地址空间)，会给页表设计带来什么样的新问题?请给出一种解决方案，分析它的优缺点。(中科大1996)

答：分页存储管理中，允许把进程的页面离散地存放在物理块中，为保证正确的地址转换，系统为进程建立了页表。进程地址空间内的所有页，依次在页表中有一页表项，其中记录了相应页在内存中的物理块号。进程运行时，通过查找页表，就可找到页面在内存中的物理位置。所以，页表的功能是实现从页号到物理块号的地址映射。

当地址空间变大后，页表也变得很大，会占用相当多的内存空间。如对于32位地址空间，若规定页面大小兴4KB，则每个进程页表的页表项有1KB。若每个页表项占4B，故每个进程页表要占用4MB内存空间，且还要求是连续的，这显然不现实。为此，可如下来解决：(1)对页表所需空间也采用离散分配和存放。(2)只将当前使用的页表页调入内存，其余的和暂不用的页面一样存放在外存中，待需要时再调入。

具体做法是采用两级页表。在用户逻辑地址原耒划分的基础上，把页表部分再细分为

页目录表和页表页、即对页表也分页(称页表页)。每个页表页的大小与物理块大小相同，32位地址空间可划分成如图。

对投入运行的程序，将其页目录表调入内存，而页表页仅调入当前使用的。程序运行若找不到相应页表页，则产生—个缺页表页中断，请求系统将该页表页调入内存。

两级页表能适应大地址空间的需要，实现了虚拟存储系统，但增加了地址变换开销和系统管理上的复杂性。

页框号 页内位移

目录位移 页表页位移 页内位移

页框号

页表页地址

进程一级页表

(页目录表)

进程二级页表

(页表页)

物理地址

逻辑地址

页目录表

控制寄存器

二级页表地址转换过程

1. 假定有一个请求分页管理系统，在某时刻测得各相关成份的利用率为CPU 20%，磁盘交换区 99%，其他I/O设备 10%。下面哪些措施将(可能)改进CPU利用率，为什么?(1)增加一个更快的CPU。(2)增加磁盘交换区大小。(3)增加多道程序道数。(4)减少多道程序道数。(5)增加其他更快速的I/O设备。(东南大学2001)

答：(1) CPU还有潜力，不必增加。(2) 磁盘容量己成瓶颈，更换更大的分页磁盘。(3) 因交换区己满，不宜增加多道程序道数。(4) 适当挂起一些用进程，减少对交换区的压力。(5)

由于其他I/O设备利用率很低，增加其他更快速的I/O设备是不必要的。

1. 页式虚拟存储器的最大容量由什么决定?如何计算?(华中科技大2000)

答： 由计算机的地址结构和辅存的容量决定，与实际主存的大小无关。若有一个地址结构为32位的计算机系统中，其虚存的最大容量是232字节。由于用户程序执行前需装入辅存，因此，虚拟存储器的的大小还受到辅存容量的限制。

1. 在虚拟页式存储管理方式下，页表有哪几种?描述各方式下存储访问过程。(清华大学2002)

答：一级页表 虚拟页式存储管理方式下需要把逻辑地址(页号+位移)转换为物理地址(页框号+位移)，一个进程在运行时，它的页表的首地址由页表控制寄存器指出，虚地址中的页号被用作索引来查找页表项，以找出对应页框号，将页框号与虚拟地址中的位移部分组合起来生成物理地址。

二级页表 在大地址空间的情况下，为了节省页表内存占用空间，可设计成两级(或多级)页表，即页表也分成一张张页表页(大小等于页面)，并不全部放入内存，虚地址分成三部分：页目录表、页表页、位移，通过页目录索引找页表页，通过页表页索引找到对应页框号，并与位移一起形成物理地址。

反置页表 反置页表为内存中的物理块建立一个页表并按照块号排序，该表的每个表项包含正在访问该页框的进程标识、页号及特征位，和哈希链指针等，用来完成内存页框到访问进程的页号，即物理地址到逻辑地址的对应转换。

40. 在设备管理中数据传输控制有哪几种?并用流程图描还DMA传输控制处理过程。(南开大学2001)(清华大学2000)

答：(1)询问方式

(2)中断方式

(3)DMA方式

* 1. 通道方式

详细参见教材。

1. 简述操作系统虚拟性持性在设备管理中的体现。(中科院2001)

答：操作系统中，并发进程共享各种软硬件资源，必须提出资源分配办法和解决资源冲突的策略与技术，为用户提供简单有效的资源使用方法，充分发挥系统资源的利用率。解决资源冲突的一种基本技术----称虚拟化技求。在设备管理中，通过用—类物理设备来模拟另一类物理设备，或通过分时地使用一类物理设备，把一个物理实体变成若干逻辑上的对应物。物理实体是实际存在的，逻辑上的对应物是虚幻的。最典型的例子，借助SPOOLing技术，用—类物理设备来模拟另一类物理设备，可把独占设备(纸带、打印机等)虚化出许许多多台独占设备供用户使用。

1. 缓冲技术主要包括哪几种方式?(哈工大2001)

答：(1)单缓冲

(2)双缓冲

(3)多缓冲

详细参见教材。

1. 在UNIX中，当一个进程向另一个进程发送信号后，接收信号的进程如何处理?(华南理工大2002)

答：当一个进程接获一个信号后，处理方式与硬中断类似。若它正在运行则立即暂停正在执行的程序，转去执行该信号事先规定的信号处理程序func，完成后再返回原先正在执行的程序；否则，信号的处理将会被延期直到进程下次被调度运行时。所不同的只是信号的设置检查都由软件实施，故又称软中断。

1. UNIX系统是如何实现文件共享的?(西北工大2000)

答：

文件的静态共享

文件的动态共享

详细参见教材。

1. 进程调度是内核的进程控制原语之一，该原语在什么情况下被启动?(北方交大1999)

答：

* 1. 进程运行完成而终止。
  2. 进程因等待而进入阻塞状态。
  3. 发现有比现行进程更高优先权进程。
  4. 进程用完了规定的时间片。
  5. 系统出现异常强行进入进程调度

1. 假设一个计算机系统具有以下性能特征：(1)处理一次中断，平均耗时1ms。(2)一次进程调度，平均需要2ms。(3)把CPU分配给选中进程，又需要平均1ms。再假设其定时器芯片每秒产生100次中断。现问：(1)操作系统将百分之几的CPU时间用于中断处理?(2)如果操作系统采用轮转法调度，10个时钟中断为1个时间片，则操作系统将百分之几的CPU时间用于进程调度(包括调度、分配CPU和引起调度的时钟中断处理时间)?(浙江大学2002)

答1：

* 1. 因处理一次中断，平均耗时1ms，而每秒产生100次中断。所以，每秒中花在中断处理上的平均时间为100ms，操作系统用于中断处理的CPU时间比率=100ms/1s=10%。
  2. 因10个时钟中断为1个时间片，亦即每秒内共有10个时间片，故每秒内进程调度花费=10×2ms=20ms。此外，进程切换也为10次，共花时间=10×1ms=10ms。而每秒100次中断仍花100ms。故操作系统用于进程调度的CPU时间比率=(20ms+10ms+100ms)/1s=130/1000=13%。

答2：

* 1. 因处理一次中断，平均耗时1ms，而每秒产生100次中断。所以，每秒中花在中断处理上的平均时间为100ms，操作系统用于中断处理的CPU时间比率=100ms/1s=10%。
  2. 因10个时钟中断为1个时间片，亦即每秒内共有10个时间片，故每秒内进程调度花费=10×2ms=20ms。此外，进程切换也为10次，共花时间=10×1ms=10ms。而中断与进程调度有关的应为10次，花了10ms。故操作系统用于进程调度的CPU时间比率=(20ms+10ms+10ms)/1s=40/1000=4%。

## 47. UNIX进程0的主要任务是什么？(清华大学1996)

## 解：当UNIX操作系统装入内存后，系统的控制权便由自举转到核心程序，即操作系统程序上来。核心首先成系统进程0，然后，由0号进程创建一个1号进程（即init进程），进程1负责初始化所有新的用户进程。

## 实际上，1号进程是除了0号进程之外所有用户进程的祖先。UNIX系统的调度与交换都是0进程的两部分，它们分别由swtch过程和sched过程实现。Sched过程把处于外存就绪态的进程换入内存，swtch则从就绪队列中寻找一优先级最高的进程。

## 因此，进程0的作用是：创建进程1，进行进程的调度和交换。

## 48. 使用文件系统时，通常要显式地进行OPEN，CLOSE操作。(北京大学1992)

## （1） 这样做的目的是什么？

## （2） 能否取消显式的OPEN，COLSE操作？应如何做？

## （3） 取消显式的OPEN，COLSE操作有什么不利？

# 解：

## （1） 显式的OPEN操作完成文件的打开功能。它将待访问文件的目录信息读入内存活动文件表中，建立起用户进程与文件的联系。显式的CLOSE操作完成文件的关闭操作。该命令撤消主存中有关文件的目录信息，切断用户与该文件的联系；或在文件打开期间，该文件作过某种修改，还应将其写顺回辅存。

## （2） 可以取消显式的OPEN与CLOSE操作。如果取消了显式OPEN与CLOSE操作，系统在进行文件操作之前需判断文件是否已打开，则应自动完成文件的打开功能，以建立用户与文件间的联系。同时，在系统结束时，还应自动关闭所有打开文件。

## （3） 取消显式的OPEN与CLOSE操作得文件的读写的系统开销增加。因为在每次读写前都需要判断文件是否已被打开。系统在结束时也要做一些额外的工作，以完成CLOSE命令的功能。当用户进程已使用完一个文件但尚未执行完成时，因无显式的CLOSE命令也无法关闭文件，从而不利于系统资源的回收。

## 49. 假定占有m块（初始块为空）的进程有一个页访问串，这个页访问串的长度为p,其中涉及到q个不同的页号，对于任何页面替换算法，计算：（1）缺页中断次数的下界是多少？

## （2）缺页中断次数的上界是多少？ (武汉大学2002)

## 答：对于任何页面替换算法，缺页中断的上限和下限：为p和q。因为有q个不同页面，无论怎样安排，不同页面进入内存至少要产生一次缺页中断，故下限为n次。一般讲m<q，引用串中有些页可能进入内存后又被调出，而多次发生缺页中断。极端情况，访问的页都不在内存，这样共发生了p次缺页中断。例如，当m=3，p=12，q=4时，有如下访问中：1，1，1，2，2，3，3，3，4，4，4，4。缺页中断为下限4次。而访问串：2，3，4，1，2，3，4，1，2，3，4，1。缺页中断为上限12次。

## 50. 操作系统的基本特征是什么？(北京航空2002)

## 答：**并发性**是指两个或两个以上的事件或活动在同一时间间隔内发生。操作系统是一个并发系统，并发性是它的重要特征，操作系统的并发性指计算机系统中同时存在若干个运行着的程序，因此，它应该具有处理和调度多个程序同时执行的能力。

## **共享性**是指计算机系统中的资源（包括硬件资源和信息资源）可被多个并发执行的用户程序和系统程序共同使用，而不是被其中某一个程序所独占。又可分互斥访问和共享访问。

## **异步性**是指在多道程序环境中，允许多个进程并发执行，由于资源有限而进程众多，多数情况下，进程的执行不是一贯到底，而是“走走停停”。异步性给系统带来了潜在的危险，有可能导致进程产生与时间有关的错误，操作系统必须保证多次运行进程，都会获得完全相同的结果。

## **虚拟性**是指操作系统中的一种管理技术，它是把物理上的一个实体变成逻辑上的多个对应物，或把物理上的多个实体变成逻辑上的一个对应物的技术。例如，在多道程序系统中，物理CPU可以只有一个，每次也仅能执行一道程序，但通过多道程序和分时使用CPU技术，宏观上有多个程序在执行，就好像有多个CPU在为各道程序工作一样，物理上的一个CPU变成了逻辑上的多个CPU。虚拟存储器则是把物理上的多个存储器(主存和辅存)变成逻辑上的一个(虚存)的例子。

1. **计算题**

1. 磁盘系统调度中，采用SCAN调度算法为任务队列67、65、124、14、122、37、183、98服务。试计算服务结束时，磁头总共移动了几个磁道。假设磁头总在第0至199道之间移初，开始服务时，磁头刚从60移到67。(浙江大学1998)

答：服务次序为67—98—122—124—183—199—65—37—14，磁头总共移动磁道=317道。

2.旋转型存储设备上信息的优化分布能减少若平I/O服务的总时间。例如，有10个记录A、B、C、…..、J在放在某磁盘的磁道上，假定这个磁道划分为10个扇区，每个扇区存放一个记录(如图)。现在要从磁道上顺序地将记录A、B、C、…..、J读出，如果磁盘转速为20ms转1周，处理程序每读出一个记录后花4ms进行处理。试问处理完10个记录的总时间是多少(从找到A记录开始计算)?为了缩短处理时间应进行优化分布，试问应如何安排这些记录，并计算处理的总时间(从找到A记录开始计算)?(国防科大2000)

扇区 1 2 3 4 5 6 7 8 9 10

记录 A B C D E F G H I J

答：

已知磁盘转速为20ms转1周，故读取一个记录花2ms，处理一个记录时间为4ms。在读取和处理它记录A时，由于磁盘已转到记录D的开始位置，为了顺序处理记录B，必让磁盘再转过8个记录才能到达记录B的开始位置。

定位记录B花8×2ms=16ms，读出记录B花2ms，处理记录B花4ms，总共22ms。这种情况对记录B~J均出现，故读取和处理记录B~J总时间=22ms×9=198ms。再加读取和处理记录A的时间6ms，合计为204ms。

若把记录如下图优化分布，即记录A安排后，顺序处理的下一个记录均隔开2个扇区(确好是处理时间4ms)，待读出并处理完上一记录后，确好到了顺序处理的下一记录的开始位置。故10个记录合计读出和处理总时间=6ms×10=60ms。

扇区 1 2 3 4 5 6 7 8 9 10

记录 A H E B I F C J G D

3. 有一个多道批处理系统，作业调度采用“短作业优先”调度算法；进程调度采用**“优先数抢占”**式调度算法，且优先数越小而优先级越高。如系统拥有打印机一台，采用静态方法分配，忽略系统的调度开销。现有如下作业序列到达系统：(南京大学2004)

作业名 到达系统时间 估计CPU运行时间 打印机需求 进程优先数

J1 14:00 40分钟 1 4

J2 14:20 30分钟 0 2

J3 14:30 50分钟 1 3

J4 14:50 20分钟 0 5

J5 15:00 10分钟 1 1

试回答：(1)按作业运行结束的次序排序，即哪一个作业第一个、第二个、…、最后一个运行结束？

(2)平均作业周转时间和平均作业带权周转时间多少？

答：分析：

14:00 14:20 14:30 14:50 15:00 15:10 15:20 16:10 16:30

J1

J2

J3

J4

J5

就绪队列等待

CPU

CPU

CPU

CPU

CPU

后备队列等待

就绪队列等待

后备队

列等待

CPU

14:00 J1到达，分得打印机，被作业调度调入主存运行。

14:20 J2到达，不要求打印机且优先权高于J1，J2运行而J1进就绪队列等待。

14:30 J3到达，因打印机不满足进后备队列等待。

14:50 J4到达，这时J2运行第一个结束，J4被作业调度调入主存但优先权小于J1，故J1继续运行。

15:00 J5到达，因打印机已分给J1，进后备队列等待。

15:10 J1运行第二个结束，归还打印机。根据“短作业优先”的作业调度算法，J5调入主存且其优先权高于J4，故J5运行。

15:20 J5运行第三个结束，归还打印机。这时，J3资源满足被调入主存并且其优先权高于J4，故J3运行。

16:10 J3运行第四个结束。接着J4最后一个运行，直到16:30分结束。

结论：

（1）按作业运行结束的次序排序为：J2、J1、J5、J3和J4。

（2）作业J1周转时间30+40

作业J2周转时间0+30

作业J3周转时间50+50

作业J4周转时间80+20

作业J5周转时间10+10

平均作业周转时间=（70+30+100+100+20）/5=320/5=64分钟

平均作业带权周转时间=（7/4+1+2+5+2）/5=2.35

4. 在实现文件系统时，为加快文件目录的检索速度，可利用“文件控制块分解法”。假设目录文件存放在磁盘上，每个磁盘块512字节。采用“文件控制块分解法”前，文件控制块占64字节，采用“文件控制块分解法”后，文件控制块分解成两部分，第1部分占10字节(包括文件名和文件内部号)，第2部分占56字节(包括文件内部号和文件其文描述信息)。(北京大学1997)

1. 假设某一目录文件共有254个文件控制块，试分别给出采用“文件控制块分解法” 前和采用“文件控制块分解法”后，查找该目录文件的某一个文件控制块的平均访盘次数?
2. 一般地，若目录文件分解前占用n个磁盘块，文件分解后改为占用m个磁盘块存放文件名及文件内部号部分，请给出访问磁盘次数减少的条件。

# 解：

1. “文件控制块分解法” 前：

文件控制块占64字节，而目录文件共有254个文件控制块，故共有字节数为64×254，

故占用磁盘块数=64×254/512，而查找该目录文件的某一个文件控制块的平均访盘次数=64×254/512/2=16。

“文件控制块分解法” 后：

文件控制块笫一部分占10字节，而目录文件共有254个文件控制块，故共有字节数为10×254，而查找该目录文件的某一个文件控制块的笫一部分平均访盘次数=10×254/512/2。为了找到文件控制块的笫二部分，还需加一次读盘，故平均访盘次数=10×254/512/2+1=4。

1. 访问磁盘次数减少的条件为：

n/2(分解前平均读磁盘块数)-m/2(分解后平均读磁盘块数)>1

即m<n-2

5. 某文件系统中，外存为硬盘，物理块大小为512B。有文件A，包含590个逻辑记录，每个记录占255B，每个物理块存放2个逻辑记录。文件A所在的目录如图，此树型目录结构由根目录节点、作为目录文件的中间节点和作为信息文件的叶节点组成。每个目录项占127B，每个物理块放4个目录项，根目录的第一块常驻内存。现问：(北京大学1991)(华中科技大2002)

(1)若文件采用串连结构，链接字占2B，那么，要将A读入内存，至少要存取几次硬盘?为什么?

(2)若文件采用连续结构，那么，要将A的第480号记录读入内存，至少要存取几次硬盘?为什么?

(3)为减少读盘次数，可采用什么措施?

root

bin

dev

etc

boot

usr

tmp

mike

mary

you

he

file1

dir1

dir2

A

B

C

D

E

(1)为把文件A读入内存，先找相关目录信息。

由于127×4+2=510<512，故可知一个物理块在串连结构下可存放4个目录项及1个链接字。由root起，第1次读硬盘得到bin、dev、etc、boot的信息和下一盘块地址。第2次读硬盘得到usr的地址，第3次读硬盘得到you的地址，第4次读硬盘得到dir1的地址，第5次读硬盘得到文件A的地址。

由于255×2+2=512，故可知一个物理块在串连结构下可存放2个逻辑记录及1个链接字(下—块的物理地址)。而文件A共有590个逻辑记录，故读取A的记录所需读盘次数=590/2=295次。

所以，为把文件A读入内存需读盘次数=295+5=300次。

(2)当文件为连续结构时，由于一次读盘便可获取usr的物理块地址，故4次读硬盘便得到文件A的地址。知道了文件A的地址，通过计算，只需要1次读盘就可读出第480个逻辑记录。

1. 为减少读盘次数，文件可采用索引节点方法。检索目录文件过程中仅用到文件名，仅当找到一个文件名相匹配的目录项时，才需从目录项中读出文件的物理地址。可用类似UNIX中把文件名与文件描述信息分开的方法，即把文件描述信息单独构成一个称为索引节点的数据结构。若设一个目录项占16B，则一个盘块可存放512/16=32个目录项。与此题一个盘块仅放4个目录项相比，可让平均启动磁盘次数减少为原来的1/8。

6有一虚拟存储系统，采用FIFO页面淘汰算法和固定分配、局部置换策略。设某—作业分得3个页框，作业运行中使用的页号依次为： 4、3、2、1、4、3、5、4、3、2、1、5

(1)作业运行中共出现多少次缺页?(2)若作业拥有4个页框，将出现多少次缺页?(3)比较上面两种情况并解释产生的现象。(北京大学1990)

答：(1) 作业分得3个页框，作业运行页面失效情况如图，共9次。

作业页面 4 3 2 1 4 3 5 4 3 2 1 5

页框3 2 2 2 3 3 3 3 3 1 1

页框2 3 3 3 4 4 4 4 4 2 2 2

页框1 4 4 4 1 1 1 5 5 5 5 5 5

缺页情况 缺 缺 缺 缺 缺 缺 缺 缺 缺

(2) 作业分得4个页框，作业运行页面失效情况如图，共10次。

作业页面 4 3 2 1 4 3 5 4 3 2 1 5

页框4 1 1 1 1 1 1 2 2 2

页框3 2 2 2 2 2 2 3 3 3 3

页框2 3 3 3 3 3 3 4 4 4 4 5

页框1 4 4 4 4 4 4 5 5 5 5 1 1

缺页情况 缺 缺 缺 缺 缺 缺 缺 缺 缺 缺

(3)可见FIFO替换算法有一个异常现象，当分给作业的页框数增加时，作业缺页数不降反升，即产生了抖动现象。其原因在于FIFO替换算法完全不考虑程序的动态特性。

1. **编程题**

1. 一组生产者进程和一组消费者进程共享10个缓冲区，每个缓冲区可以存放一个整数；生产者进程每次一次性向3个缓冲区写入整数，消费者进程每次从缓冲区取出一个整数。请用PV操作写出能够正常执行的程序。**(南京大学2001)**

答：

VAR buf:array[0..9] of integer；

count,getptr,putptr:integer;

count:=1;getptr:=0;putptr:=0;

S1,S2,SPUT,SGET:semaphore;

S1:=1;S2:=1;SPUT:=3;SGET:=0;

main( )

{

cobegin

process producer\_i

begin

L1:生产3个整数；

P（SPUT）；

P（S1）；

buf[putptr]:=整数；

putptr:=(putptr+1) mod 10;

buf[putptr]:=整数；

putptr:=(putptr+1)mod 10;

buf[putptr]:=整数；

putptr:=(putptr+1)mod 10;

V(SGET);

V(SGET);

V(SGET);

V(S1);

GOTO L1;

end;

process consumer\_j

VAR y:integer;

begin

L2:P(SGET);

P(S2);

Y:=buf[getptr]:

getptr:=(getptr+1)mod 10;

count:=count+1;

if count=3 then

begin

count:=0;

V(SPUT);

end;

V(S2);

消费一个整数y；

COTO L2；

end；

coend.

2. 有如图所示的工作模型。其中有3个进程P0、P1、P2和3个缓冲区B0、B1、B2。进程间借助于相邻缓冲区传递信息，Pi每次从Bi中取一条消息，经加工送义B(i+1) mod 3中。B0、B1、B2分别可存放3、2、2个消息。初始时，仅B0有一个消息。用P、V操作写出P0、P1、P2的同步及互斥流程。**(北京理工大1999)**

B0

P0

P2

P1

B1

B2

解：

本题中，有3组生产者---消费者，第一组 P2为生产者，P0为消费者，使用缓冲区B0；第二组 P0为生产者，P1为消费者，使用缓冲区B1；第三组 P1为生产者，P2为消费者，使用缓冲区B2。P2每生产一个产品放入B0，P0从B0取出一个产品加工后放入B1。然后，P0又变成第二组的生产者，而P1为消费者，工作过程类似，直到第三组生产者---消费者进程。注意，每个Pi均要与Bi和B(i+1)两个缓冲区联系。

对B0、B1、B2可设置互斥信号量mutex0、mutex1、mutex2。由于缓冲区容量均大于1，可把B0、B1、B2构造成环形队列，相应的输入指针为in0、in1、in2，而输出指针为out0、out1、out2。empty0、empty1、empty2分别代表B0、B1、B2空缓冲个数，而full0、full1、full2分别代表B0、B1、B2装满消息的缓冲区个数，已知初始时B0中装了一个消息。

var

mutex0 ，mutex1，mutex2:semaphore;

empty0，empty1，empty2:semaphore;

full0，full1，full2:semaphore;

in0，in1，in2，out0，out1，out2:integer;

mutex0 :=mutex1:=mutex2:=1;

empty0:=empty1:=empty2:=2; /\*B0为三个缓冲区，二个是空的

full0:=1;full1:=full2:=0; /\*B0中有一个是满的

in0:=1;in1:=in2:=0; /\*in0=1是因B0中已有一消息

cobegin

process P0

begin

repeat

P(full0);

P(mutex0);

从B0(out0)取一个消息;

out0:=(out0+1) mod 3;

V(mutex0);

V(empty0);

加工消息;

P(empty1);

P(mutex1);

加工消息送B1(in1);

in1:=(in1+1) mod 2

V(mutex1);

V(full1);

until false;

end;

process P1

begin

repeat

P(full1);

P(mutex1);

从B1(out1)取一个消息;

out1:=(out1+1) mod 2;

V(mutex1);

V(empty1);

加工消息;

P(empty2);

P(mutex2);

加工消息送B2(in2);

in2:=(in2+1) mod 2

V(mutex2);

V(full2);

until false;

end;

process P2

begin

repeat

P(full2);

P(mutex2);

从B2(out2)取一个消息;

out2:=(out2+1) mod 2;

V(mutex2);

V(empty2);

加工消息;

P(empty0);

P(mutex0);

加工消息送B0(in0);

in0:=(in0+1) mod 3

V(mutex0);

V(full0);

until false;

end;

coend.

3. 有n个进程将字符读入到一个容量为80的缓冲区中（n>1），当缓冲区满后，由另一个进程Pb负责一次取走这80个字符。这种过程循环往复，请写出n个读入进程（P1， P2，…Pn）和Pb的动作序列。（可以用文字或表达式来描述动作序列，并假设Pi每次读一个字符到缓冲区中。）**(大连理工2002)**

解：

var mutex,empty，full:semaphore;

count,in:integer

buffer:array[0..79] of char;

mutex:=1;empty:=80;full:=0;

count:=0;in:=0;

cobegin

process P i(i=1,...,n))

begin

L: 读入一字符到x;

P(empty);

P(mutex);

Buffer[in]:=x;

in:=(in+1) mod 80;

count++;

if (count==80) then

{count:=0;V(mutex);V(full);}

else V(mutex);

goto L;

end;

process Pb

begin

P(full);

P(mutex);

读出字符从buffer[0];

…

读出字符从buffer[79];

in:=0;

V(mutex);

for j:=1 to 80 do

V(empty);

end;

coend.

4. 有座东西方向架设、可双向通行的单车道简易桥，最大载重负荷为4辆汽车。请定义合适的信号量，正确使用PV操作，给出任一车辆通过该简易桥的管理算法，解答中请回答下列问题：**(南京大学2003)**

(1) 临界区管理的原则是什么?

(2) 分析该题中的互斥现象和同步现象。

(3) 说明信号量的声明和初值设定的理由。

(4) 给出上述问题的解决算法，结合该算法，简述PV操作解决该问题的基本思路。

解：

（1），临界区管理原则有：a)一次只能一个进程进入临界区；b)等待进入临界区的时间必须是有限的；c)临界区空闲时，访问临界区的请求应立即响应；d)进程在临界区中的驻留时间必须有限。

（2）该题中的互斥现象有两点：第一，因为简易桥是单向通过，因此东西方向过桥的车辆对桥的使用是互斥的，第二，同方向上可能会同时有多辆车辆过桥，为了正确释放桥的使用，必须设置桥上车辆的计数器，对计数器的使用也是需要互斥的。该题中的同步现象有：当桥上已有4辆车辆时，同方向上的第5辆之间存在同步现象。

（3）据上分析，信号量应该有4个：S，初值为1，代表桥的互斥使用的信号量；Scounteast, 初值为1，代表由东向西方向行驶的桥上的车辆计数器的互斥使用；Scountwest,初值为1，代表由西向东方向行驶的桥上的车辆计数器的互斥使用；Scount4，初值为4，代表桥上车辆的计数信号量，用于同步管理。

（4）算法如下：

var S, Scounteast, Scountwest, Scount4;:semaphore;

S:=1;Scounteast:=1;Scountwest:=1;Scount4:=4;

Counteast, Countwest:integer;

Counteast:=0;Countwest:=0;

Cobegin

process east(i)

begin

P(Scounteast);

if Counteast=0 then P(S);

Counteast:=Counteast+1;

V(Scounteast);

P(Scount4);

上桥；过桥；下桥；

V（Scount4）;

P(Scounteast);

Counteast:=Counteast-1;

if Counteast=0 then V(S);

V(Scounteast);

end;

process west(i)

begin

P(Scountwest);

if Countwest=0 then P(S);

Countwest:=Countwest+1;

V(Scountwest);

P(Scount4);

上桥；过桥；下桥；

V(Scount4);

P(Scountwest);

Countwest:=Countwest-1;

if Countwest=0 then V(S);

V(Scountwest);

end;

coend.

解题的基本思路如下：

●关于桥的互斥使用：互斥管理只发生在：同方向上的第一辆车在上桥前需要查看桥的状态是否为“可用”，此时，可以使用P(S) 操作；类似，同方向上的最后一辆车下桥时，必须释放桥的使用，此时，可以使用V(S)操作。

●为判断桥上车辆数目，必须设置计数器Counteast和Countwest.，而上述两变量则是为新引入的共享变量，必须互斥使用，因此，设置两个信号量Scounteast和Scountwest，使用PV操作进行管理。

●桥的最大负荷为4辆，实际上类似于“读者—写者问题”中共享有四个缓冲区的缓冲池，这是一种同步管理。根据题意，同步管理应该设置在“上桥”和“下桥”时，否则会出现等待车辆可能没有及时过桥的错误。

5. 有一共享文件F和两组并发进程（A组和B组），该两组进程在共享文件F时只进行读且受到如下限制：**(南京大学1996)**

·同一组的进程可同时读文件F；

·当某组有进程在读文件F时，不允许另一组中的任何进程读文件F；

·当无进程在读文件F时则允许任何一组中的进程去读文件F。

请用管理（monitor）来实现对共享文件F的管理。

答：

|  |
| --- |
| type sharefile = MONITOR  var Acount, Bconut : integer;  A, B : condition;  Define Astart\_read, Aend\_read, Bstart\_read, Bend\_read;  use wait, signal, check, release; |
| procedure Astart-read;  begin  check(IM);  if Bcount>0 then wait(A,IM);  Acount := Acount + 1;  signal(A, IM);  release(IM);  end;  procedure Aend-read;  begin  check(IM);  Acount := Acount - 1;  if Acount=0 then signal(B,IM);  release(IM);  end;  procedure Bstart-read;  begin  check(IM);  if Acount>0 then wait(B,IM);  Bcount := Bcount + 1;  signal(B, IM);  release(IM);  end;  procedure Bend-read;  begin  check(IM);  Bcount := Bcount - 1;  if Bcount=0 then signal(A,IM);  release(IM);  end; |
| begin  Acount := 0; Bcount := 0; A := 0; B := 0;  end. |

任何一个进程读文件前，首先调用start-read，执行完读操作后，调用end-read。即：

|  |
| --- |
| cobegin  process A-group-reader  begin  ……  call sharefile.Astart-read;  ……  read the F;  ……  call sharefile.Aend-read;  ……  end;  process B-group-reader  begin  ……  call sharefile.Bstart-write;  ……  read the F;  ……  call sharefile.Bend-write;  ……  end;  coend. |

6.进程P使用缓冲区B向进程Q1,Q2,…,Qm发送消息，要求每当P向B中发消息，只有当所有的进程Q1,Q2,…,Qm都读取这条消息后，P才可向B发新消息。利用P,V原语写进它们的动作序列。(大连理工1999)

解：在本题中，进程间发/收消息如图进行。

P

### B

Q1

Qm

…

可以把缓冲区B分作有m个空槽，每个可存一个消息，发消息时P共发m个，取消息时Qi仅取自已的一个消息，这样可满足题意。于是，应设置一个信号量mutex实现诸进程对缓冲区B的互斥访问；一个信号量empty和信号量数组full[m]描述缓冲区的消息收发情况，应符合发一次、Qi各收1次而共为m次。mutex的初始值为1；empty初值为m；full[m]初值为0。其同步关系描述如下：

var mutex,empty,full[m]：semaphore;

B:array [0..m-1] of message;

int i;mutex=1;

empty=m;

for(i=0;i<=m-1;i++)

{

full[i]=0;

}

cobegin

process P

begin

while (1)

{

send( );

}

end;

process Qi (i=1,2,…,m)

begin

while (1)

{

receive(i);

}

end;

coend

send ( ) /\*P发送消息的动作\*/

{ int i;

for (i=0;i<=m-1;i++)

P(empty);

p(mutex);

for(i=0;i<=m-1;i++)

{

B[i]:=new message;

}

V(mutex）;

for(i=0;i<=m-1;i++)

V(full[i]);

}

receive(i) /\*进程Qi接收消息动作\*/

{

P(full[i]);

P(mutex);

take the message from B[i];

V(mutex);

V(emtpy);

}