

梦享考研系列

2016 年考研核心考点命题思路解密

计算机操作系统

梦享团队 组编

内容简介

《2016 年考研核心考点命题思路解密——计算机操作系统》严格按照最新计算机考研 408 统考大纲的计算机操作系统部分编写，涵盖大纲指定的所有考试内容。本书对统考大纲所涉及的知识点进行深入剖析和总结，并精心策划和部署每一个章节，对每一个章节的考点做了独家策划。

本书每一个考点中的命题，绝大部分来源于历年名校计算机考研真题和统考真题，少部分来源于名校期末考试试题中的精华部分，是全国 408 统考大纲和高校考研真题的较好结合。为了提高考题的质量和解析的准确度，参考资料采用以考研权威教材、习题、考研真题为主，多方借鉴众多高校从事多年教育的教师课堂资料。梦享团队对每一个命题的思路和解题方法进行深入详细地讲解，并附上大量的图来帮助考生理解记忆，力求考生能够通过掌握一个题目而达到举一反三，有利于考生利用更少的时间掌握更多的知识。

本书可作为考生参加计算机专业研究生入学考试的备考复习用书，也可作为计算机专业的学生学习计算机操作系统的练习用书。

前言

梦享团队队员以中科院、清华大学和北京交通大学 3 所高校的学生为主，其他名校学生为辅，都是上研不久的研究生，以及一些考研论坛上参与答疑多年的版主等。在考研复习和辅导上，梦享团队队员有着相对丰富的阅历。在考研的路上，梦享团队队员也经历过和大家一样的坎坷辛苦。我们深切地体会到，每一个考研的同学十分不容易。

计算机考研的同学注重基础知识的掌握，更加看重实战能力。但目前的考研教材绝大多数倾向于知识点的讲解，不注重培养考生的实战能力，导致很多考生知识很丰富，但是很难这些知识很好地运用于解题。编写偏向于实战的参考书不同于知识讲解，需要编者花费大量的时间来规划和布置章节、考点和解析考题。目前市面上现在能找到的计算机考研命题解析类参考资料，要么题目特别少但讲解特别详细啰嗦，要么题目太多的而对命题的讲解十分粗略甚至只有一个最终答案。因而，梦享团队决定写一套注重实战、解析详细、严格遵照大纲的参考书。

经过两年多的努力，“梦享考研系列”参考书终于一本一本地和大家见面了，到目前为止，我们总共有 5 本图书已经出版。其中，《计算机网络》完成于 2013 年 10 月，《数据结构》完成于 2014 年 3 月，《计算机操作系统》完成于 2014 年 9 月，《计算机组成原理》完成于 2015 年 3 月。最后一本书，《核心题型》，初稿完成于 2015 年 5 月。

为了提高图书的权威性，本套图书严格按照 408 统考大纲编写，涵盖了统考大纲所有指定的内容，并融合了历年名校考研真题的精华，是全国 408 统考大纲和高校考研真题的较好结合。为了提高考题的质量和解析的准确度，参考资料采用以考研权威教材、习题、考研真题为主，多方借鉴众多高校从事多年教育的教师课堂资料。

本套图书具有以下特色：

1. 组织严谨，结构清晰

梦享考研系列图书通过对统考大纲和历年高校考研真题的深入剖析和总结，精心规划和部署了各个章节，对每一个章节的考点作了独家策划，使得本套图书组织严谨，结构清晰，便于大家对各章考点各个击破。

2. 突出重点，注重实战

对于每一个计算机专业的考研同学而言，时间是相当有限的。除了四门统考的繁重专

业课之外，我们还有数学、英语和政治，复习工作量相当大。所以，突出重点，让同学们把极其有限的时间都花在刀刃上，是我们的首要工作。而提高同学们的实战能力，是我们系列图书的最终目的。

因此，在考题的挑选上，我们通过对统考和自主命题高校常出现的考题类型和知识点进行深入的总结，抛开在统考或者自主命题的考研真题上极少出现的极难、极易、极偏知识点，精心挑选了和考研难度相近的题型和考题供大家练习，提高同学们的实战能力。

此外，我们还根据考点的重要程度来完成考点内容分布，在较重要的考点部署较多的内容，在较重要的内容部署较多的命题，在较为不重要的知识点抓住重点布置核心题型。目的也在于突出考试难度、突出考试重点，方便大家进行实战训练，提高学习效率，让考生在更短的时间内掌握更多的知识点。

3. 解析详细，深入剖析

“梦享考研”系列图书一共 5 本，每一本都很厚，可能会吓怕很多同学。是不是题目太多了？不是的，其实考题并不多，我们并不提倡题海战术，也不提倡对于同一个知识点反复命题和赘述，我们提倡“少而精”。针对每一个考点可能出现的命题类型，我们精心挑选了极具代表性的命题供大家实战训练，并对这些习题进行详细、深入的剖析，揭露问题的本质和解题的精髓，有助于大家掌握解题方法和技巧，提高大家的实战能力，在较短的时间掌握更多的知识。

《2016 年计算机考研核心考点命题思路解密》系列图书是我们团队 2 年多的汗水结晶，融入了我团队的集体智慧。另外，真诚感谢我们团队新成员单开元、殷巧云、张丽方、胡明明、刘春、白洋等 10 几位同学提供的建议和帮助！

在接下来的更长时间里，我们团队将日益强大，用我们最诚挚的心和最大的努力，给大家展示出更好的图书。我们每年都会合理调整这套图书，使得这套图书更加受到大家青睐。

梦享团队我们会牢牢记住这样一句话——“助你们实现研究生梦想，是我们的梦想！”伴随着 2016 年考研的同学一起度过艰辛的追梦季节，伴随着大家一起度过每一个艰辛的日日夜夜！也祝福 2016 年考研的你们，获得圆满的成功！

安楠

2015 年 1 月于北京

目 录

第一章 操作系统概述	7
考点 1 操作系统的基本概念	7
考点 2 操作系统的发展与分类	10
考点 3 操作系统的运行环境	13
第二章 进程管理	18
考点 1 进程的概念	18
考点 2 进程的状态与转换	22
考点 3 进程控制、组织与通信	25
考点 4 线程概念和多线程模型	27
考点 5 进程同步的基本概念	29
考点 6 实现临界区互斥的基本办法和信号量机制	31
考点 7 经典同步问题与 PV 算法	35
第三章 处理机调度和死锁	45
考点 1 调度的概念和基本准则	45
考点 2 典型的处理机调度算法	49
考点 3 死锁	60
考点 4 系统安全状态、银行家算法	63
第四章 存储器管理	71
考点 1 内存管理的基本概念	71
考点 2 交换与覆盖	74
考点 3 连续分配管理方式	76
考点 4 非连续分配管理方式	84
考点 5 虚拟内存的基本概念	88
考点 6 请求分页管理方式	92
考点 7 页面置换算法和抖动	97
考点 8 请求分段管理方式和请求段页式管理方式	103

第 5 章 文件管理	108
考点 1 文件系统的基本概念	108
考点 2 文件结构	111
考点 3 目录结构	114
考点 4 文件共享和保护	119
考点 5 文件系统的实现	120
考点 6 磁盘存储器管理	124
第 6 章 输入输出 (I/O) 管理	130
考点 1 设备管理的基本概念	130
考点 2 I/O 控制方式	131
考点 3 缓冲管理	136
考点 4 SPooling 技术	139

第一章 操作系统概述

考点 1 操作系统的基本概念

温馨提示：本考点考查操作系统的基本概念、特征、功能，以及操作系统提供的服务，都是基本知识，请同学们理解和掌握操作系统的并发、共享、虚拟和异步四个主要特征，操作系统对计算机资源的管理（包括处理机调度、存储器管理、文件管理和设备管理），以及操作系统提供的用户与计算机硬件之间的程序接口和命令接口。

一. 选择题部分

1. (原书 第 2 题)从用户的观点看，操作系统是（ ）。
- A. 用户与计算机之间的接口
 - B. 控制和管理计算机资源的软件
 - C. 合理地组织计算机工作流程的软件
 - D. 由若干层次的程序按一定的结构组成的有机体

【解析】本题考查操作系统的功能。研究 OS 的应用主要有以下几个角度：

- (1). 从资源管理的角度看，OS 是计算机资源的管理者。
- (2). 从用户的角度看，操作系统是计算机提供给用户的接口。
- (3). 从机器扩充的角度看，操作系统是部署在裸机之上的第一层软件，配置了操作系统的机器，我们称之为虚拟机。

故而，本题选择 A 答案。

参考答案： A

2. (原书 第 5 题)操作系统的两个最主要的特征是（ ）。
- A. 并发性和虚拟性
 - B. 并发性和共享性
 - C. 共享性和异步性
 - D. 共享性和虚拟性

【解析】本题考查操作系统的基本特征。OS 主要有以下特征：

(1). **并发**。在一段时间内，有多道程序在宏观上同时执行，多道程序设计导致多个程序并发执行。

(2). **共享**。多道程序或者多个用户作业共享有有限的计算机系统资源。

(3). **虚拟**。把物理上的一个变成逻辑上的多个或扩大容量。

(4). **异步**。在 OS 控制下的多个作业的执行顺序和每个作业的执行顺序时间是不确定的。值得一提的是，异步性不是指 OS 本身的功能不确定或用户程序的结果不确定。程序的并发执行，使得程序执行的时间顺序不确定，但是程序执行的结果却是一定的。

其中，并发和共享是操作系统最基本的两个特征。

【理解帮助】

并发——“大家都前进了”

共享——“一件东西大家用”

异步性——“你走我停，走走停停”

参考答案： B

3. (原书 第 7 题) 下列选择中，() 不是操作系统关心的主要问题。

- A. 管理计算机裸机
- B. 设计、提供用户程序与计算机硬件系统的界面
- C. 管理计算机系统资源
- D. 高级程序设计语言的编译器

【解析】本题考查操作系统的作用。操作系统有两个重要的作用：

第一，通过资源管理，提高计算机系统的效率。操作系统是计算机系统的资源管理者，通过 CPU 管理、存储管理、设备管理和文件管理，对各种资源进行合理的分配，改善资源的共享和利用程度，提高资源利用率和系统吞吐量。

第二，改善人机界面，向用户提供友好的工作环境。操作系统不仅是计算机硬件和各种软件之间的接口，也是用户与计算机之间的接口。在裸机上安装操作系统后，用户可以利用操作便利、界面友好的操作系统来使用计算机，提高工作效率。

选项 D 所说的“高级程序设计语言的编译器”不是操作系统关心的主要问题。

参考答案： D

4. (原书 第 10 题) 在计算机系统中，操作系统是()。

- A. 处于裸机之上的第一层软件
- B. 处于硬件之下的低层软件
- C. 处于应用软件之上的系统软件

D. 处于系统软件之上的用户软件

【解析】计算机系统抽象层次结构如图 1.1 所示。



图 1.1 计算机系统抽象层次结构

操作系统是处于裸机上的第一层软件。在裸机上安装了操作系统，则成为虚拟机。故而，选择 A 答案。

参考答案： A

5. (原书 第 11 题)所谓 () 是指将一个以上的作业放入主存，并且同时处于运行状态，这些作业共享处理机和外围设备等其他资源。

- A. 多重处理
- B. 多道程序设计
- C. 实时处理
- D. 共同执行

【解析】本题考查多道程序设计的基本概念。多道程序设计是指在内存中同时存放多道程序，这些程序在管理程序的控制下交替进行，共享处理机及系统中的其他资源。

【补充】一个作业进入系统到运行结束，一般要经历进入、后备、运行和完成四个阶段。相应地，作业亦有进入、后备、运行和完成四种状态。

(1). **进入状态**：作业的信息从输入设备上预输入到输入井，此时称为作业处于进入状态。

(2). **后备状态**：当作业的全部信息都已输入，且由操作系统将其存放在输入井中，此时称作业处于后备状态。系统将所有处于后备状态的作业组成后备作业队列，等待作业调度程序的调度。

(3). **运行状态**：一个后备作业被作业调度程序选中，分配了必要的资源，调入内存运行，称作业处于运行状态。请同学们注意作业的运行状态与进程的运行状态的区别。如图 1.2 所示。

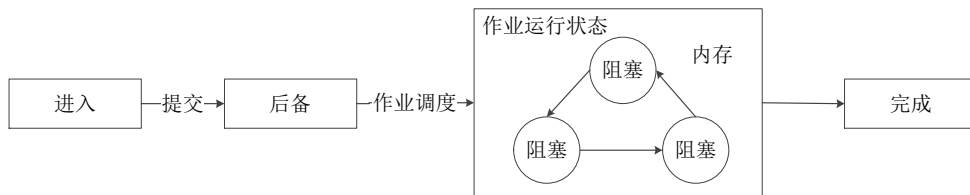


图 1.2 作业和进程的状态关系

(4). **完成状态**：当作业正常运行完毕或因发生错误非正常终止时，作业进入到完成状态。

参考答案： B

考点 2 操作系统的发展与分类

温馨提示：本考点主要考查操作系统的分类，操作系统的发展历年考得比较少。请同学们多注意批处理操作系统、分时系统、实时系统的特点，并注意区别分时与实时的区别。对网络操作系统、分布式计算机系统和个人计算机操作系统，也稍作了解。

一. 选择题部分

1. (原书 第 2 题) () 不是分时系统的基本特征。

- A. 同时性 B. 独立性 C. 实时性 D. 交互性

【解析】 本题考查操作系统的基本特征。分时系统有如下特征：

- (1). 同时性：计算机系统能被多个用户同时使用；
- (2). 独立性：用户和用户之间对系统是独立操作的，在同时操作时并不会发生冲突，破坏，混淆等现象；
- (3). 及时性：系统能以最快的速度将结果显示给用户；
- (4). 交互作用性：用户能和电脑进行人机对话。

实时性不是分时操作系统的基本特征，而是实时系统的基本特征。要注意区别实时和分时的区别。

参考答案： A

2. (原书 第 9 题) 批处理系统的主要缺点是 ()。

- A. CPU 的利用率不高 B. 失去了交互性

C. 不具备并行性

D. 以上都不是

【解析】本题考查批处理操作系统的特点。在批处理系统中，操作人员将作业成批地装入计算机中，由操作系统在计算机中某个特定磁盘区域（输入井）将其组织好，并按一定的算法选择其中的一个或多个作业，将其调入内存使其运行。运行结束后，把结果放入磁盘“输出井”，由计算机统一输出后交给用户。批处理操作系统在系统中配置了一个监督程序，并在该监督程序的控制下，系统能够对一批作业自动进行处理。

多道程序设计技术和批处理技术的采用，使多道批处理系统具有资源利用率高和系统吞吐量大的优点。但是，多道批处理系统将用户和计算机操作员分开，使用户无法直接与自己的作业进行交互。另外，由于在多道批处理系统中，作业要进行排队，依次处理，因而作业的平均周转时间较长。

参考答案： B

3. (原书 第 16 题)为了使系统中所有的用户都能得到及时的响应，该操作系统应该是 ()。

A. 多道批处理系统

B. 分时系统

C. 实时系统

D. 网络系统

【解析】本题考查分时系统的性质。

我们对分时操作系统和实时操作系统作一个简单比较，二者的主要区别如下：

(1). **设计目标不同**。前者为了给多用户提供一个通用的交互型开发运行环境，后者通常为特殊用途提供专用系统；

(2). **交互性强弱不同**。前者交互性强，后者交互性弱；

(3). **响应时间要求不同**。前者以用户能接受的响应时间为标准，后者则与受控对象及应用场合有关，变化范围很大。

为了使系统中所有用户都能得到及时响应，应更加重视系统与用户的交互能力，应采用分时系统。

参考答案： B

4. (原书 第 23 题)下面关于操作系统论述中正确的是 ()。

A. 批处理作业必须具有作业控制信息

B. 分时系统不一定都具有人机交互功能

C. 从响应时间角度看，实时系统与分时系统相似

D. 由于采用了分时技术, 用户可以独占计算机资源

【解析】注意分时系统的定义：分时操作系统是指在一台主机上连接多个带有显示器和键盘的终端，同时允许多个用户通过主机的终端，以交互方式使用计算机，共享主机中的资源。故而分时系统一定具有人机交互能力。

从响应的角度看，实时系统对响应时间的要求比分时系统高出很多。

分时操作系统是一个多用户交互式操作系统。分时操作系统，主要分为三类：单道分时操作系统、多道分时操作系统、具有前台和后台的分时操作系统。分时操作系统将 CPU 的时间划分成若干个片段，称为时间片。操作系统以时间片为单位，轮流为每个终端用户服务，用户并没有独占计算机。

批处理系统没有交互能力，用户的作业从提交到完成都由监督程序来处理，故而必须具有作业控制信息。

参考答案: A

5. (原书 第 24 题)对系统的如下指标

I. 内存容量 II. 设备数量 III. CPU 速度 IV. 中断响应时间

在多道程序设计中，道数限制要考虑的因素是

A. I 和 II

B. II 和 IV

C. III和IV

D. I 和 IV

【解析】 本题考查了多道程序设计。

多道程序设计通过提高处理机、设备、内存等各种资源的利用率，来提高系统效率。说透了，就是通过增加同时运行程序的道数可以提高资源利用率，从而提高系统效率，但道数应与系统资源数量相当，道数过少，系统资源利用率低，道数过多，系统开销增大，程序响应速度下降。

参考答案: A

二. 综合应用题部分

1. (原书第1题)为何引入多道程序设计?在多道程序系统中,内存中作业的道数是否越多越好?请说明原因。

【解析】引入多道程序设计技术是为了提高计算机系统资源的利用率。在多道程序系统中，内存中作业的道数并非越多越好。一个计算机系统内的内存、外设等资源是有限的，

只能容纳适当数量的作业,当作业道数增加时,将导致对资源的竞争激烈,系统开销增大,从而导致作业的执行缓慢,系统效率下降。

考点 3 操作系统的运行环境

温馨提示：本考点主要考查：1、操作系统的运行机制，包括时钟管理、中断、原语；2、中断和异常的概念和区别；3、系统调用。请同学们注意用户态和核心态的转换的典型事件，并注意区别中断和系统调用。我们在考题中给出了相关的例题，供大家参考。

一. 选择题部分

1. (原书 第3题) 计算机系统中设置的访管指令, () 执行。
- A. 只能在目态
B. 只能在管态
C. 既可在目态又可在管态
D. 在目态和管态下都不能

【解析】本题考查访管指令的执行。关于这个问题，不同的书有不同的说法。有的书说访管指令只能在目态下执行，有的书说只能在管态下执行，还有的书说既可在目态下执行，也可在管态下执行。

“**访管指令**”是在目态（用户态）下执行的指令。当源程序中有需要操作系统服务的要求时，编译程序就会在由源程序转换成的目标程序中安排一条“访管指令”并设置一些参数。当目标程序执行时，中央处理器若取到了“访管指令”就产生一个中断事件，中断装置就会把中央处理器转换成管态，并让操作系统处理该中断事件。操作系统分析访管指令中的参数，然后让相应的“系统调用”子程序为用户服务。系统调用功能完成后，操作系统把中央处理器的管态改为目态，并返回到用户程序。

值得注意的是，访管指令不是特权指令。特权指令是指用于操作系统或者系统软件的指令，一般不给用户使用。

参考答案: A

2. (原书第5题)当硬件中断装置发现有事件发生,就会中断正在占用 CPU 的程序执行,让操作系统的()占用 CPU。
- A. 系统调用程序 B. 中断处理程序
- C. 作业管理程序 D. 文件管理程序

【解析】本题考查中断机制。中断机制包括硬件中断装置和操作系统的中断处理服务程序。中央处理器每执行完一条指令，均由中断装置判别是否有事件发生。若无事件发生，CPU 继续执行；若有事件发生，则中断装置中断原占有 CPU 的程序的执行，让操作系统的处理事件服务程序（中断处理程序）占用 CPU，对出现的事件进行处理，事件处理完后，再让原来的程序继续占用 CPU 执行。

参考答案： B

3. (原书 第10题)为了使计算机在运行过程中能及时处理内部和外部发生的各种突发性事件，现代操作系统采用了（ ）机制。
- A. 查询 B. 中断 C. 调度 D. 进程

【解析】本题考查引入中断机制的原因。中断是指程序执行过程中，遇到急需处理的事件时，暂时中止 CPU 上现程序的运行，转去执行相应的事件处理程序，待处理完成后再返回原程序被中断处或调度其他程序执行的过程。

【故事助记】CPU 老板是一家公司的光杆司令，所有的顾客都要他亲自跑去处理，还要跟有关部门打点关系，CPU 觉得顾客和公关这两样事它一个人搞不来，这就是轮询；终于这家公司升级发展了，CPU 老板请了一个秘书，所有的顾客都先由秘书经手，CPU 心情好的时候就去看一下，大部分时间都忙着去公关了，这时它觉得轻松了很多，这就是中断。

【补充】借助这个题，我们来简单介绍一下中断。中断的过程如图 1.3 所示。

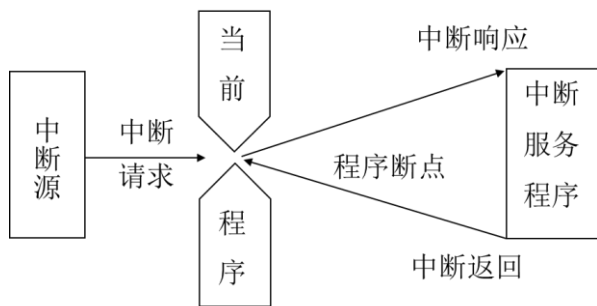


图 1.3 中断过程

中断是处理器处理外部突发事件的一个重要技术。它能使处理器在运行过程中对外部事件发出的中断请求及时地进行处理，处理完成后又立即返回断点，继续进行处理器原来的工作。再来看看中断的相关概念：

- I. 整数除以零 II. `sin()` 函数调用 III. `read` 系统调用
- A. 仅 I、II B. 仅 I、III
- C. 仅 II、III D. I、II 和 III

【解析】用户态切换到内核态的 3 种方式：

(1). 系统调用

这是用户态进程主动要求切换到内核态的一种方式，用户态进程通过系统调用申请使用操作系统提供的服务程序完成工作。

(2). 异常

当 CPU 在执行运行在用户态下的程序时，发生了某些事先不可知的异常，这时会触发由当前运行进程切换到处理此异常的内核相关程序中，也就转到了内核态，比如缺页异常。

(3). 外围设备的中断

当外围设备完成用户请求的操作后，会向 CPU 发出相应的中断信号，这时 CPU 会暂停执行下一条即将要执行的指令转而去执行与中断信号对应的处理程序，如果先前执行的指令是用户态下的程序，那么这个转换的过程自然也就发生了由用户态到内核态的切换。比如硬盘读写操作完成，系统会切换到硬盘读写的中断处理程序中执行后续操作等。

这 3 种方式是系统在运行时由用户态转到内核态的最主要方式，其中系统调用可以认为是用户进程主动发起的，异常和外围设备中断则是被动的。I 属于异常，III 属于系统调用。`sin()` 函数的调用，不属于系统调用。故而，选择 B 答案。

参考答案： B

6. (原书 第 20 题) 对如下中断事件

- I. 时钟中断 II. 访管中断 III. 缺页中断 IV. 控制台中断
- 能引起外部中断的事件是 ()。
- A. I 和 II B. II 和 III
- C. III 和 IV D. I 和 IV

【解析】本题考查系统调用。系统调用是指操作系统提供给用户程序调用的一组“特殊”接口，用户程序可以通过这组“特殊”的接口获得操作系统内核提供的服务。例如用户可以通过进程控制相关的系统调用来创建进、实现进程调度等。

参考答案： A

二. 综合应用题部分

1. (原书 第 1 题)什么为核心态、用户态、特权指令? 下列哪些指令为特权指令?

- (1). 改变存储器管理寄存器
- (2). 写程序计数器
- (3). 读日历钟
- (4). 设置日历钟
- (5). 改变处理器优先级
- (6). 写指令寄存器

【解析】核心态是 CPU 运行操作系统代码, 用户态是 CPU 运行用户程序代码的状态。通过系统调用、Trap、中断可以使得系统从用户态到核心态。特权指令指的是只能由操作系统而不是用户调用的指令。

(1) 是 (2) 否 (3) 否 (4) 是 (5) 是 (6) 否

本章到此就结束了, 你觉得本章有什么地方不明白吗? 任何问题, 欢迎您与我们作者进行交流!



shareOurDreams
梦享团队微信号



weCSdream
梦享团队官方微信公众号



梦享论坛团队
梦享团队新浪微博

第二章 进程管理

考点 1 进程、线程的概念

温馨提示：本考点考查进程和线程的概念、特点，请同学们掌握进程调度算法，进程和线程的区别。我们在本考点布置了进程原语的基本概念，请同学们掌握这些基本知识。

一. 选择题部分

1. (原书 第 1 题)引入进程概念的关键在于()。

- A. 共享资源 B. 独享资源 C. 顺序执行 D. 便于执行

【解析】本题考查引入进程概念的目的。在多道环境下，多个程序的并发执行使得它们失去了封闭性，使得执行结果具有不可再现性。为此引入了进程的概念，便于更好地描述和控制程序的并发执行，实现资源共享。

参考答案： A

2. (原书 第 8 题)在单一处理机上执行程序，多道程序的执行是在()进行的。

- A. 同一时刻 B. 同一时间间隔内
C. 某一固定时刻 D. 某一固定时间间隔内

【解析】本题重点考查了考生对并发和并行区别的理解。并发和并行不是相同的概念。并行是指在同一时刻，有多条指令在处理机上同时执行。并发是指在同一时刻，只有一条指令执行，但是多个进程指令被快速轮换执行，使得宏观上具有多个进程同时执行的效果。

【故事助记】要吃完三个馒头的任务，一个人可以这个馒头吃一口，那个馒头咬一口，这样交替进行，最后吃完三个馒头，这就是并发。因为三个馒头同时发生了吃的行为。而如果三个馒头分给三个人吃，这样的任务完成方式叫并行。

参考答案： B

3. (原书 第 12 题) 程序运行时, 独占系统资源, 只有程序本身能改变系统资源状态, 这指的是 ()。

- A. 程序顺序执行的再现性
- B. 程序顺序执行的封闭性
- C. 并发程序失去封闭性
- D. 并发程序失去再现性

【解析】封闭性是指, 程序运行在一个封闭的环境中, 即程序运行时独占系统的全部资源, 这些资源的状态只能因程序的执行而改变, 不受任何外界因素的影响。故而在 B。

程序的可再现性是程序顺序执行的另一个重要特征, 它指的是, 只要程序顺序执行的初始条件相同, 无论何时执行, 无论程序在执行期间是否停顿, 程序所得的结果都应该相同。

参考答案: B

4. (原书 第 15 题) 一个进程是 ()。

- A. 由协处理器执行的一个程序
- B. 一个独立的程序 + 数据集
- C. PCB 结构、程序和数据的集合
- D. 一个独立的程序

【解析】本题考查进程的结构。进程实体由程序、数据与 PCB 三个部分组成。

参考答案: C

5. (原书 第 17 题) 进程具有的特性包括: ()。

- ①动态性 ②共享性 ③并发性 ④相互制约性 ⑤独立性 ⑥静态性
- A. ①③④⑤
- B. ①②④⑤
- C. ②④⑤⑥
- D. ①②④⑥

【解析】本题考查进程的特征。进程的主要有以下特性:

(1). 动态性

进程是一个动态的概念, 实质上是程序的一次执行过程。进程具有生命期: 它由内核“创建”而产生, 由调度程序“调度”而执行, 在请求 I/O 操作时可能被阻塞, 执行时走走停停, 当它完成自己的任务后, 由内核终止这个进程而使其消亡。

(2). 并发性

在一个系统中, 可以同时存在多个进程, 它们并发地执行, 共享系统资源。正由于多个进程的并发执行, 才使系统资源得到充分的利用, 系统的处理能力大大增强。

(3). 独立性

进程是一个能独立运行的基本单位。它有自己的程序计数器和内部状态，是系统进行资源分配和调度的基本单位。未建立进程的程序不能作为一个独立单位参与运行，并发进程获得所需的资源才能向前推进。

(4). 异步性

每个进程按各自独立的不可预知的速度向前推进。正由于进程的异步性，并发进程相互穿插地执行，才失去了程序的可再现性。但由于进程之间相互制约，进程执行时也常间断，出现走走停停的现象。

(5). 结构特征

从结构上看，一个进程实体由三个部分组成，即程序、数据和进程控制块。进程要完成一个任务，就必须执行描述其任务的程序，并对要求加工的数据进行处理。为对进程实施有效的管理，系统就必须为进程设立一个专门的数据结构——进程控制块。

静态性与相互制约性不是进程的特征。

参考答案： A

6. (原书 第 22 题) 进程与程序之间有密切联系，但又是不同的概念。二者的一个本质区别是 ()。

- A. 程序是静态概念，进程是动态概念。
- B. 程序是动态概念，进程是静态概念。
- C. 程序保存在文件中，进程存放在内存中。
- D. 程序顺序执行，进程并发执行。

【解析】本题考查进程与程序的区别。关于程序与进程的区别，这部分我们已经讲述很多了。进程与程序最本质的区别在于，进程是动态的，程序是静态的。动态性是进程最重要的特性。

参考答案： A

二. 综合应用题部分

1. (原书 第 1 题) 何谓并行？何谓并发？在单处理机系统中，下述并行和并发现象哪些可能发生，哪些不会发生？

- (1). 进程与进程之间的并行。
- (2). 进程与进程之间的并发。
- (3). 处理机与设备之间的并行。
- (4). 处理机与通道之间的并行。
- (5). 通道与通道之间的并行。
- (6). 设备与设备之间的并行。

【解析】本题考查并发与并行的概念和区别。所谓并行是指同一时刻同时进行，进程并行需要多处理器的支持。所谓并发，是指在一段时间内，多个进程都在向前推进，而在同一时刻，可能只有一个进程在执行，多个进程轮流使用处理器。在单处理器系统中，可能发生的并行和并发现象如下：

(1). **进程与进程之间的并发**。例如，在 Windows 操作系统中，mp3 播放进程和 Word 字处理进程可以并发执行，这样用户就可以边听音乐边写文章了。

(2). **处理机与设备之间的并行**。例如，当处理机进行科学运算时，打印机可以打印文档。

(3). **处理机与通道之间的并行**。通道程序的执行可与处理机的操作并行。

(4). **通道与通道之间的并行**。通常一个系统中有多个通道，这些通道可以并行地执行相应的通道程序。

(5). **设备与设备之间的并行**。例如打印机打印文档时，磁带机在输入数据。

2. (原书 第 4 题)操作系统中为什么要引入进程的概念？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，操作系统在进程管理方面应做哪些工作？

【解析】本题考核操作系统引入进程的目的以及操作系统在进程管理中需要做的工作。

(1). 操作系统引入进程，是为了让程序并发方式执行，能够充分利用系统资源，提高系统的处理能力。但由于系统资源是有限的，诸多并发执行的程序在共享资源的同时，必将引起资源的竞争。此时如果不制定特定的规则和方法，必将使这种共享和竞争呈现无序状态。程序的执行结果也将不可避免地失去封闭性和可再现性，从而得不出正确的、预期的结果。

正因为如此，多道程序设计中需要引入一个能描述程序执行过程，且能用来共享资源的基本单位——那就是“进程”。因此，进程可以被定义为“可并发执行的程序在一个数据集上的执行过程”。

(2). 操作系统对进程管理方面应做如下工作：

- **进程控制**。包括进程创建与撤消，进程在运行过程中的状态转换，以及实现对进程控制块的维护等操作。

- **进程调度**。操作系统必须按一定算法在就绪进程中选择一个进程，把处理机分配给它，使它顺利地投入运行。为此，进程调度应具有 CPU 现场信息的保护和恢复功能。

● **进程同步**。对于一组合作的进程，它们的推进速度应当受到某种约束，以便协调一致地向前推进。因此系统必须设立同步控制机制，对所有进程的运行进行协调。协调方式包括进程互斥方式和进程同步方式。

● **进程通信**。在多道程序环境下，进程之间往往要互相发送一些信息。操作系统应提供有关的通信调用和通信规范，保证实现这些进程之间的信息交换。进程之间的通信种类是很多的，控制机制也有很多。

考点 2 进程的状态与转换

温馨提示：本考点考查进程的状态和转换，非统考的高校喜欢考查引起进程发生状态转换的典型事件，请同学们多了解。

一. 选择题部分

1. (原书 第 2 题) 系统在 ()，发生从目态到管态的转换。

- | | |
|-------------|--------------|
| A. 发出 P 操作时 | B. 发出 V 操作时 |
| C. 执行系统调用时 | D. 执行置程序状态字时 |

【解析】本题考查系统调用。系统调用是操作系统向用户程序提供支持的接口。通过这些接口，应用程序向操作系统请求服务，控制转向操作系统。操作系统在完成服务后，将控制和结果返回给用户程序。

从逻辑上看，系统调用可以被看成是一个内核与用户空间程序交互的接口。它好比一个中间人，把用户进程的请求传达给内核，待内核把请求处理完毕后再将处理结果返回给用户空间。

参考答案： C

2. (原书 第 7 题) 支持多道程序设计的操作系统在运行过程中，不断地选择新进程运行来实现 CPU 的共享，但其中 () 不是引起操作系统选择新进程的直接原因。

- | | |
|-------------------|---------------|
| A. 运行进程的时间片用完 | B. 运行进程出错 |
| C. 运行进程要等待某一事件的发生 | D. 有新进程进入就绪状态 |

【解析】本题考查引起进程创建的原因。引起进程调度的原因有以下几类：

(1). 正在执行的进程执行完毕。这时，如果不选择新的就绪进程执行，将浪费处理机资源。

(2). 执行中进程自己调用阻塞原语将自己阻塞起来进入阻塞状态。

(3). 执行中进程调用了 P 原语操作，因资源不足而被阻塞；或调用了 v 原语操作激活了等待资源的进程队列。

(4). 执行中进程提出 I/O 请求后被阻塞。

(5). 在分时系统中时间片已经用完。

(6). 在执行完系统调用等系统程序后返回用户进程时，这时可看作系统进程执行完毕，从而可调度选择一新的用户进程执行。

以上都是在不可剥夺方式下的引起进程调度的原因。在 CPU 执行方式是可剥夺时，还有：

(7). 就绪队列中的某进程的优先级变得高于当前执行进程的优先级，从而也将引发进程调度。

当运行的进程出错之后，CPU 停止运行该进程，进行错误处理，调度新的进程执行。但是，有新进程进入就绪队列，不是构成进程状态切换的直接原因。

参考答案： D

3. (原书 第 9 题) 进程状态从就绪态到运行态的转化工作是由 () 完成的。

- A. 作业调度 B. 中级调度 C. 进程调度 D. 设备调度

【解析】本题考查进程状态的转换。处理机是计算机系统最重要的资源，提高处理机的利用率，改善系统性能很大程度取决于处理机调度性能的优劣。处理机调度的主要目标有：

(1). 对 CPU 资源进行合理的分配使用，以提高 CPU 的利用率。

(2). 使得各用户公平地得到处理机资源。

(3). 减少用户的响应时间。

处理机调度分为三个层次：

(1). **高级调度**。高级调度又叫做作业调度/宏观调度，指的是按照一定原则选择外存输入井上后备作业队列中的一个或者多个作业进入内存，并为其建立相应的进程。高级调度决定允许哪些作业竞争系统资源，它仅涉及虚拟处理机分配，分时与实时系统中无需该层调度。

(2). **中级调度**。中级调度，又叫做交换调度，它决定哪些进程可以竞争处理机。作为作业进入系统和将处理机分配给这些作业二者之间的一个缓冲，中级调度通过使进程临时

挂起和激动的方法对系统负载波动做出反映,以便获得平稳的系统操作和实现较好的系统综合性能指标。引入中级调度是为了提高内存的利用率和系统吞吐量。

(3). **低级调度**。低级调度又称为进程调度/微观调度,简单的说,就是当内存不为空时,它决定将处理机分配给哪一个就绪进程,它将处理机实际地分配给某一个进程。

由以上分析可知,进程由就绪态转运行态,是由进程调度完成的。故而,选择 C 答案。

参考答案: C

4. (原书 第 11 题)如果某一进程在运行时,因某种原因暂停,此时将脱离运行状态,而进入()。

- A. 自由状态 B. 停止状态 C. 阻塞状态 D. 静止状态

【解析】正在运行的进程,因为等待某一事件的发生而暂停运行,进入阻塞状态。故而,本题选择 C 答案。

参考答案: C

5. (原书 第 22 题)关于进程切换有如下论述

- | | |
|----------------------|-----------------------|
| I. 根据系统栈保存下降进程的现场。 | II. 根据 PCB 保存下降进程的现场。 |
| III. 根据系统栈恢复上升进程的现场。 | IV. 根据 PCB 恢复上升进程的现场。 |
- 其中论述正确的是()。
- A. I 和 III B. I 和 IV
C. II 和 III D. II 和 IV

【解析】本题考查进程切换。

进程切换伴随着系统栈的切换。发生进程切换时,下降进程的现场信息从系统栈中弹出,保存到下降进程的 PCB 中。上升进程的现场信息从上升进程的 PCB 中恢复。

参考答案: B

二. 综合应用题部分

1. (原书 第 1 题)请画出进程的状态转换图。并说明是什么事件引起每种状态的变迁? 状态转换图 2.1 所示。

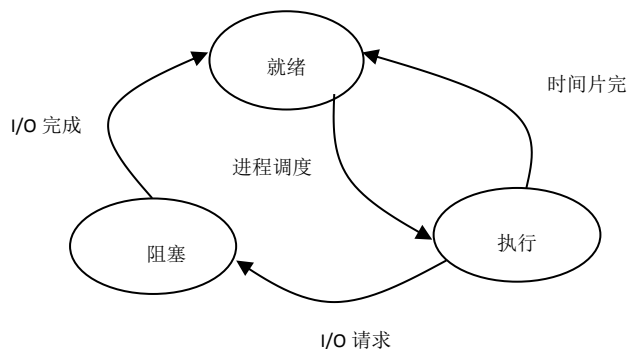


图 2.1 进程的状态转换图

【解析】本题考查进程状态的转换和引起转换的原因。引起下列进程状态转换的典型事件有：

(1). **就绪到执行**：处于就绪状态的进程，在调度程序为之分配了处理器之后，该进程就进入执行状态。

(2). **执行到就绪**：正在执行的进程，如果分配给它的时间片用完，则暂停执行，该进程就由执行状态转变为就绪状态。

(3). **执行到阻塞**：如果正在执行的进程因为发生某事件（例如：请求 I/O，申请缓冲空间等）而使进程的执行受阻，则该进程将停止执行，由执行状态转变为阻塞状态。

(4). **阻塞到就绪**：处于阻塞状态的进程，如果引起其阻塞的事件发生了，则该进程将解除阻塞状态而进入就绪状态。

考点 3 进程控制、组织与通信

温馨提示：本考点考查进程的控制、组织和通信，历年统考和非统考考查不多，请同学们稍微了解一下。

一. 选择题部分

1. (原书 第 2 题) 在实现进程通信时会导致调用 Send 原语的进程被设置成“等信箱”状态的原因是（ ）。

A. 指定的信箱不存在

B. 调用时没有设置参数

C. 指定的信箱中无信件

D. 指定的信箱中存满了信件

【解析】为了实现信箱通信，必须提供相应的原语，如创建信箱原语、撤消信箱原语、发送信件原语和接收信件原语等。

由于发送方和接收方都是独立工作的，如果发得快而收得慢，则信箱会溢出。相反，如果发得慢而收得快，则信箱会变空。因此，为避免信件丢失和错误地送出信件，一般而言通信应有如下的规则：

(1). 若发送信件时信箱已满，则发送进程应被置等信箱状态，直到信箱有空时才被释放。

(2). 若取信件时信箱中无信，则接收进程应被置成等信件状态，直到有信件时才被释放。

故而，本题选择 D 答案。

参考答案： D

2. (原书 第 5 题) P、V 操作是 ()。

A. 两条低级进程通信原语

B. 两组不同的机器指令

C. 两条系统调用命令

D. 两条高级进程通信原语

【解析】本题考查 PV 操作。利用 P、V 操作，进程间只能交换少量信息，而且交换的信息仅是控制信息，显然其通信效率极低。这样的通信原语，称为**低级通信原语**。故而，选择 A 答案。

与低级通信原语相比，我们把能在进程间传送大量数据信息的通信原语，称为**高级通信原语**。

参考答案： A

3. (原书 第 7 题) 用信箱实现进程间相互信息的通信机制要有两个通信原语，它们是 ()。

A. “发送”原语和“执行”原语

B. ”就绪“原语和”执行“原语

C. ”发送“原语和”接收“原语

D. ”就绪“原语和”接收“原语

【解析】用信箱实现进程间相互通信的通信机制要有两个通信原语，它们是“发送(send)原语”和“接收(receive)原语”。其中，Send(N,M)把信件 M 送到指定的信箱 N 中，Receive(N,Z)从指定信箱 N 中取出一封信，存放在指定的地址 Z 中。故而，本题选择 C 答案。

参考答案： C

二. 综合应用题部分

1. (原书 第 1 题) 进程之间有哪些基本的通信方式？它们分别有什么特点？适用于哪些场合？

【解析】进程通信根据交换信息量的多少分为高级通信和低级通信。低级通信一般只传送一个或几个字节的信息，以达到控制进程执行速度的作用（如 PV 操作）；高级通信则要传送大量数据，目的不是为了控制进程的执行速度，而是为了交换信息。

高级进程通信方式大致归为 3 类：共享存储器，管道文件，消息传递。

(1). **共享存储器**：在内存中分配一片空间作为共享存储区。需要进行通信的进程把它附加到自己的地址空间中，不需要则把它取消。

(2). **管道文件**：连接两个命令的一个打开文件。一个命令向该文件中写入数据，另一个命令从该文件中读出数据。

(3). **消息传递**：以消息为单位在进程间进行数据交换。

考点 4 线程概念和多线程模型

温馨提示：本考点考查线程的概念和多线程模型。请同学们注意线程的基本概念、线程和进程的区别、以及线程的实现方式，并对多线程模型作一个简单的了解。

一. 选择题部分

1. (原书 第 1 题) 下面关于线程的叙述中，正确的是（ ）。
- A. 不论是系统支持线程还是用户级线程，其切换都需要内核的支持。
 - B. 线程是资源的分配单位，进程是调度和分配的单位。
 - C. 不管系统中是否有线程，进程都是拥有资源的独立单位。
 - D. 在引入线程的系统中，进程仍是资源分配和调度分派的基本单位。

【解析】本题考查线程。在用户级线程中，线程管理的所有工作都由应用程序完成，内核感知不到线程的存在。而且，用户级线程的切换不需要内核的支持。

在只有进程的操作系统中，进程是资源分配和独立调度的基本单位。在同时有进程和线程的操作系统中，线程是独立调度的基本单位，而进程仍然是资源分配的基本单位。故而，C 答案正确。

参考答案：C

2. (原书 第4题)下面关于进程和线程的说法，正确的是()。

- A. 线程是进程中可独立执行的子任务，一个进程可以包含一个或多个线程，一个线程可以属于一个或多个进程。
- B. 多线程技术具有明显的优越性，如速度快、通信简便、设备并行性高等。
- C. 由于线程不作为资源分配单位，线程之间可以无约束地并行执行。
- D. 线程又称为轻型进程，因为线程都比进程小。

【解析】本题考查进程和线程的区别与联系。一个线程只能属于一个进程，所以 A 答案错误。进程切换涉及到当前 CPU 环境的保存和新调度执行进程 CPU 环境的设置。与进程切换不同的是，线程切换只需要保存和设置少量寄存器内容，切换速度快，开销小。

由于同属一个进程的多个线程共享进程的内存地址空间，使得这些线程之间通信变得简便，易于实现。B 答案正确。

线程虽然不作为资源分配的基本单位，但是线程之间不可以无约束的并行执行。

线程是轻型实体，是因为线程基本不拥有资源。更值得注意的是并不是所有线程都比进程小。

参考答案：B

3. (原书 第8题)【2012 年统考真题】下列关于进程和线程的叙述中，正确的是()。

- A. 不管系统是否支持线程，进程都是资源分配的基本单位
- B. 线程是资源分配的基本单位，进程是调度的基本单位
- C. 系统级线程和用户级线程的切换都需要内核的支持
- D. 同一进程中的各个线程拥有各自不同的地址空间

【解析】本题考查线程与进程的区别与联系。不管操作系统是否引入线程，进程都是资源分配的基本单位。不同的是，在引入线程的操作系统中，线程取代进程成为独立调度的基本单位。所以，A 正确。

用户级线程运行在用户空间，操作系统无法感知，用户线程间的切换不需要内核的支持。故而，C 选项错误。同一个进程的各个线程可以共享进程的地址空间，所以 D 答案错误。

参考答案：A

二. 综合应用题部分

1. (原书 第 2 题) 比较用户级线程与系统级线程间在以下方面的差别和各自的优缺点。

(1) 创建速度；(2) 切换速度；(3) 并行性；(4) TCB 的存储位置。

【解析】本题考查用户级线程和系统级线程的区别。用户级线程由系统库支持。线程的创建和撤销，以及线程状态的变化都由库函数控制并在目态完成。与线程相关的控制结构 TCB 保存在目态空间，并由运行系统维护。由于用户级线程对操作系统不可见，系统调度仍以进程为单位，核心栈的个数与进程个数相对应。

用户级线程的优点在于：

- (1). 线程不依赖于操作系统，可以采用与问题相关的调度策略，灵活性好；
- (2). 同一进程中的线程切换不需进入操作系统，因而实现效率较高。

用户级线程的缺点在于：

- (1). 同一进程中的多个线程不能真正并行，即使在多处理机环境中；
- (2). 由于线程对操作系统不可见，调度在进程级别，某进程中的一个线程通过系统调用进入操作系统受阻，该进程的其它线程也不能运行。

核心级别线程通过系统调用由操作系统创建，线程的控制结构 TCB 保存于操作系统空间，线程状态转换由操作系统完成。另外由于系统调度以线程为单位，操作系统还需要为每个线程保持一个核心栈。

核心级线程的优点是并发性好，在多 CPU 环境中同一进程中的多个线程可以真正并行执行。核心级别线程的缺点是线程控制和状态转换需要进入操作系统完成，系统开销比较大。

考点 5 进程同步的基本概念

温馨提示：本考点考查进程同步的基本概念，请同学们注意进程同步与互斥的区别，了解临界区、临界资源等基本概念。

一. 选择题部分

1. (原书 第 1 题)两个进程合作完成一个任务。在并发执行中,一个进程要等待其合作伙伴发来消息,或者建立某个条件后再向前执行,这种制约性合作关系被称为进程的 ()。
- A. 同步 B. 执行 C. 互斥 D. 调度

【解析】本题考查进程同步的相关概念。本题在考点 3 选择题第 4 小题已经考过一次,请读者参考该答案。进程同步与互斥,请读者认真理解,在写 PV 操作的时候,我们需要找到题目中隐含的进程同步与互斥关系。

参考答案: A

二. 综合应用题部分

1. (原书 第 1 题)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). 两队举行篮球比赛。
 - (3). 流水线生产的各道工序。
 - (4). 商品生产和消费。

【解析】本题考查进程间的同步与互斥。进程间存在着两种相互制约的关系:直接制约关系(即同步问题)和间接制约关系(即互斥问题)。同步问题是存在逻辑关系的进程之间相互等待产生的制约关系,互斥问题是相互无逻辑关系的进程间竞争使用相同的资源所发生的制约关系。

- (1). 属于互斥关系,因为书的数量是有限的,一本书只能借给一个同学。
- (2). 属于互斥关系,篮球只有一个,两队都要争夺。
- (3). 属于同步关系,各道工序的开始都依赖前道工序的完成。
- (4). 属于同步关系,商品没生产出来,消费无法进行,商品被未消费完,生产也无需进行。

2. (原书 第 2 题)同步机制应遵循的四个准则是什么?

【解析】同步机制应遵循的四个准则是：

- (1). **空闲让进**：当无进程处于临界区时，相应的临界资源处于空闲状态，因而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，以有效地利用资源。
- (2). **忙则等待**：当已有进程进入临界区时，表示相应的临界资源正被访问，因而所有其它试图进入相关临界区的进程必须等待，以保证这些进程互斥访问临界资源。
- (3). **有限等待**：对要求访问临界资源的进程，应保证该进程能在有限的时间内进入自己的临界区，以免陷入“永远等待”状态。
- (4). **让权等待**：当进程不能进入临界区时，应立即释放处理器，以免陷入“忙等”状态。

考点 6 实现临界区互斥的基本办法和信号量机制

温馨提示：本考点考查实现临界区互斥的基本办法和信号量机制。这部分内容比较杂乱，尤其是实现临界区互斥的基本方法有多种，不太好理解。同学们可以适当掌握浅显的一部分，难度比较大的，可以只作简单了解。

一. 选择题部分

1. (原书 第 2 题)我们把在一段时间内，只允许一个进程访问的资源，称为临界资源，因此，我们可以得出下列论述，正确的论述为（ ）。
 - A. 对临界资源是不能实现资源共享的。
 - B. 只要能使程序并发执行，这些并发执行的程序便可对临界资源实现共享。
 - C. 为临界资源配上相应的设备控制块后，便能被共享。
 - D. 对临界资源，应采取互斥访问方式，来实现共享。

【解析】本题考查临界资源。临界资源并不是独占资源，而是可以互斥共享的资源。与一个临界资源相关的多个进程，为了保证各自运行结果的正确性，当其中一个进程正在对该临界资源进行操作时，就不允许其他进程同时对其进行操作。进程间的这种制约关系，叫做互斥。对临界资源，应该采取互斥访问方式来实现共享。所以，D 答案正确。

程序只有并发，才会引出共享的概念。试想一下，若无论任何时刻，进程都不并发，进程从开始执行到退出系统可是使用系统的所有资源，那么无需共享。

参考答案：D

2. (原书第5题)有三个进程共享同一段程序段，而每次最多允许两个进程进入该程序段，若用PV操作作为同步机制，而信号量S的取值范围为()。
- A. 2, 1, 0, -1 B. 3, 2, 1, 0
- C. 2, 1, 0, -1, -2 D. 1, 0, -1, -2

【解析】本题考查信号量。三个进程共享一个程序段，程序段最多同时允许两个进程同时进入。那么，信号量 S 初值应该是 2。信号量的最小值，是三个进程先后都执行 P 操作，使得 S 由 2 变成 1，再变成 0，最后变成的-1。故而，信号量 S 的变化范围是 (-1,0,1,2)，选择 A 答案。

参考答案: A

3. (原书 第6题)下面临界区概念论述正确的是()。
- A. 临界区是指进程中用于实现进程互斥的那段程序代码
 - B. 临界区是指进程中用于实现进程同步的那段程序代码
 - C. 临界区是指进程中用于实现进程通信的那段程序代码
 - D. 临界区是指进程中用于访问临界资源的那段程序代码

【解析】本题考查临界区的相关概念。在进程程序中，只有涉及到共享变量的那一部分程序，才真正需要保证互斥地执行。通常把进程中“真正需要保证互斥执行”的那一段程序，称为该进程的“临界区”。故而，D 答案正确。

参考答案: A

4. (原书 第 12 题) V 原语对信号量做运算后, ()。

- A. 当 $S < 0$ 时进程继续执行
- B. 当 $S < 0$ 时要唤醒一个就绪进程
- C. 当 $S \leq 0$ 时要唤醒一个等待进程
- D. 当 $S \leq 0$ 时要唤醒一个就绪进程

【解析】本题考查 V 操作的实现。我们回顾一下 V 原语：

```
void signal(semaphore S)
{
    S.value++;
    if(S.value <= 0)
    {
        remove a process P from S.L;
        wakeup(P);
    }
}
```

从 V 操作原语中可以发现， $S.value++$ 之后，即对信号量做运算（++操作）后，才判断 $S.value$ 是否小于等于 0。若小于等于 0，则表示原来的 $S.value \leq -1$ ，仍有进程因等待该资源而被阻塞，故而调用 `wakeup()` 原语，唤醒一个等待进程，选择 C 答案。

参考答案： C

5. (原书 第 24 题) 对于有两个并发进程的系统，设互斥信号量为 `mutex`，若 `mutex=0`，则（ ）。
- A. 表示没有进程进入与 `mutex` 相关的临界区
 - B. 表示有一个进程进入与 `mutex` 相关的临界区
 - C. 表示有一个进程进入与 `mutex` 相关的临界区，另一个进程等待进入
 - D. 表示有两个进程进入与 `mutex` 相关的临界区

【解析】本题考查信号量机制。`mutex=1`，这是我们心中常有的概念。那么，`mutex` 是不是一直为 1 呢？显然不是，本题就是一个例子。两个进程互斥访问临界资源，都可以执行 `P(mutex)` 操作，即 `mutex` 要完成两次减 1，`mutex` 信号量的最小值是 -1。

其实 `mutex` 虽然特殊，但是我们千万不要忘了，`mutex` 的本质也是一个信号量，其在 PV 操作下与其他信号量基本一致的。

言归正传，本题中 `mutex=0`，表示两个进程中有一个进程对 `mutex` 信号执行了一次 P 操作，并进入了与 `mutex` 相关的临界区。但是另一个进程还没有进行 P (`mutex`) 操作，或

者已经执行完 $V(\text{mutex})$ 操作，没有在临界区中，也不处于等待进入临界区的状态。所以，本题选择 B 最合适。

参考答案：B

二. 综合应用题部分

1. (原书 第 2 题) 简述信号量 S 的物理含义。

【解析】 $S > 0$ 时， S 表示可使用的资源数； $S = 0$ 时，表示无资源可供使用；或表示不允许进程再进入临界区。 $S < 0$ 时， $-S$ 表示等待使用资源的进程个数或表示等待进入临界区的进程个数（假设每个进程只需要 1 个 S 资源）。当 $S > 0$ 时，调用 $P(S)$ 的进程不会等待；调用 $V(S)$ 后使可用资源数加 1 或使可用资源的进程数加 1。当 $S < 0$ 时，调用 $P(S)$ 的进程必须等待；调用 $V(S)$ 后将释放一个等待使用资源者或释放一个等待进入临界区者。

2. (原书 第 5 题) 图 2.2 给出了四个进程合作完成某一任务的前驱图，试说明这四个进程的同步关系，并用信号量描述它。

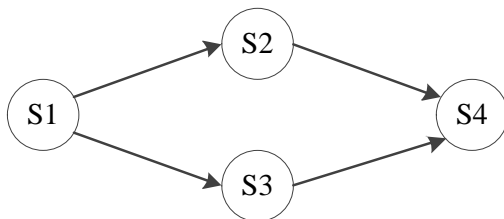


图 2.2 四个合作进程的前驱图

【解析】图 2.2 说明任务启动后 $S1$ 先执行，当 $S1$ 结束后， $S2$ 、 $S3$ 可以开始执行， $S2$ 、 $S3$ 执行完成后， $S4$ 才可以开始执行。为了确保这一执行顺序，设 4 个同步信号量 a 、 b 、 c 、 d 分别表示 $S1 \rightarrow S2$ 、 $S1 \rightarrow S3$ 、 $S2 \rightarrow S3$ 、 $S2 \rightarrow S4$ 的前驱关系，初始值均为 0。那么，利用信号量的 $Wait()$ 和 $Signal()$ 操作来实现同步的方法如下：

```

Semaphore a=b=c=d=0;
begin
  cobegin
    begin S1;signal(a);signal(b);end
    begin wait(a);S2;signal(c);end
    begin wait(b);S3;signal(d);end
  end
end
  
```

```
begin wait(c);wait(d);S4;end  
coend  
end
```

考点 7 经典同步问题与 PV 算法

温馨提示：本考点考查 PV 算法，是操作系统核心考点，统考和非统考高校常常在这里命题。PV 操作的一个难点是，同学们写 PV 操作可能和参考答案不一样，老是不知道自己写对了没有。我们认为，只要能够满足题意，写成什么风格都是正确的。不同的人，可能写法不一样。本考点特别重要，请一定掌握。

一. 综合应用题部分

1. (原书 第 3 题)生产者-消费者问题

用一个数组表示具有 n 个缓冲区的缓冲池；用输入指针 in ，指示下一个可投放消息的缓冲区；用输出指针 out ，指示下一个可获取消息的缓冲区，利用记录型信号量解决资源使用问题。

【解析】本题可用 PV 操作“五步曲”来解答，解题过程如下：

第一步：找进程

- 生产者
- 消费者

第二步：找操作

(1). 生产者

- 生产一个数据
- 将数据放入 in 指针指向的缓冲区

(2). 消费者

- 从 out 指针指向的缓冲区读取数据
- 消费数据

第三步：找关系

- 只有缓冲区有空闲，生产者才能将生产的数据放入（空闲）缓冲区中。

- 只有缓冲区有数据，消费者才能取数据（来消费）。

第四步：找初值

- n 个缓冲区，可用信号量 `empty` 表示。设初始状态下缓冲区全部为空，即为 n 。此时生产者可以有 n 个空闲的缓冲区用来存放数据，消费者初始状态下没有直接消费的数据，需等待。
- 设 `full` 表示当前已经被生产者生产和放入数据的缓冲区数量，初始值为 0。
- 缓冲区必须互斥访问，所以，设置互斥信号量 `mutex`，初值为 1。
- 用一个数组 `buffer`，用来表示缓冲区。
- 两个指针 `in` 和 `out`，`in` 指针指向生产者可以直接存放生产数据的缓冲区，`out` 指针指向消费者可以直接消费生产者生产好的数据的缓冲区。

【注意】`buffer` 数组是用来表示缓冲区的，而 `empty` 和 `full` 是用来表示缓冲区的空闲和使用情况的，这二者不一样，请读者仔细体会。

第五步：写算法

PV 算法如下：

```
Var mutex, empty, full: semaphore := 1, n, 0;    //三个信号量
buffer: array [0, ..., n-1] of item;           //n 个缓冲区
in, out: integer := 0, 0;                       //生产者和消费者指针
begin
  parbegin
    生产者进程 :
      begin
        repeat
          ...
          produce an item nextp;                //生产一个产品（数据）放入 nextp
          ...
          wait(empty);                          //看看有没有空闲的缓冲区
          wait(mutex);                          //有空闲的缓冲区
          buffer(in) := nextp;                  //把 nextp 中的产品送往 buffer(in);
          in := (in + 1) mod n;                  //in 指针指向下一个位置
          signal(mutex);                        //释放互斥信号
          signal(full);                         //满缓冲区加 1
        until false;
```

```

end
消费者进程:
begin
    repeat
        wait(full);           //消费者看看有没有满的缓冲区
        wait(mutex);         //有满的缓冲区
        nextc:=buffer(out);   //从 buffer(out)中取出产品放入 nextc
        out:= (out+1) mod n;   //out 指针指向下一个位置
        signal(mutex);        //释放缓冲区互斥访问信号
        signal(empty);        //空闲缓冲区的数量加 1
        ...
        consume the item in nextc; //消费 nextc 中的产品
    until false;
end
parend
end

```

2. (原书 第6题) 阅览室问题

假定一个阅览室最多可容纳 100 人，读者进入和离开阅览室时都必须在阅览室门口的一个登记表上进行登记，而且每次只允许一人进行登记操作，请用记录型信号量机制实现上述问题的同步。

【解析】阅览室问题，是我们常遇到的问题，其实阅览室问题、博物馆参观问题、银行问题、售票厅问题等等，这些问题都很相似，就是容量一定，顾客在容纳得下的情况下才能进入。利用解 PV 算法的“五步曲”，可解析本题如下：

第一步：找进程

- 读者（若干）（其实类似于阅览室问题的一系列问题如博物馆问题、售票厅问题等，通常都只涉及一类或者简单的几类进程）

第二步：找操作（每个读者的操作都相似）

- 进入登记
- 进入阅览室
- 阅读

- 离开登记
- 离开阅览室

第三步：找关系

- 读者在进入和离开时都需要互斥登记。
- 读者在阅览室人数不满的时候，才能进入阅览室。
- 读者离开了以后，给阅览室腾出空间，等待进入阅览室的读者才能进入。

第四步：找初值

- 设读者进入和离开登记的互斥信号量 `mutex`，初始值为 1。
- 阅览室最多自能容得下 100 人，设置信号量 `empty` 表示当前阅览室还能进来的人数。设初始状态下阅览室没有人，`empty=100`。

第五步：写算法

定义信号量 `sum`, `mutex`，初值分别为 100, 1。则第 i 个读者的活动描述为：

```

procedure Pi( $i=1,2,3,\dots$ )
begin
    wait(empty);           //进入阅览室之前，先看看阅览室是不是人满了
    wait(mutex);           //人没有满，访问信号量 mutex，互斥登记
    进入登记;
    signal(mutex);         //释放互斥信号量，让待入者可进，待出者可出
    进入阅览室;
    阅读;
    wait(mutex);           //要离开，互斥登记
    离开登记;
    signal(mutex);         //释放互斥登记的信号量
    离开阅览室;
    signal(empty);         //读者离开后，阅览室可进如的人数加 1
end
  
```

3. (原书 第 11 题) 读者—写者问题(Readers-Writers Problem)

问题描述：有一个许多进程共享的数据区，这个数据区的一块空间；有一些只读取这个数据区的进程（Reader）和一程(Writer)，此外还需要满足以下条件：

- (1). 任意多个读进程可以同时读这个文件；
- (2). 一次只有一个写进程可以往文件中写；

(3). 如果一个写进程正在进行操作，禁止任何读进程读文件。

用信号量来实现读者写者问题的以下调度算法。

(1). 读者优先

要求指一个读者试图进行读操作时，如果这时正有其直接开始读操作，而不需要等待。

(2). 写优先

一个读者试图进行读操作时，如果有其他写者在等写操作，他要等待该写者完成写操作后才开始读操作。

【解析】本题考查经典同步问题中读者写者问题。

(1). 读者优先算法

对于读者优先，应满足下列条件：

如果新读者到：

- 1) 无读者、写者，新读者可以读；
- 2) 有写者等待，但有其它读者正在读，则新读者也可以读；
- 3) 有写者写，新读者等待。

如果新写者到：

- 1) 无读者，新写者可以写；
- 2) 有读者，新写者等待；
- 3) 有其它写者，新写者等待。

读者优先的设计思想是读进程只要看到有其它读进程正在读，就可以继续进行读；写进程必须等待所有读进程都不读时才能写，即使写进程可能比一些读进程更早提出申请。

单纯使用信号量是不能解决读者-写者问题的，必须引入计数器 `Readcount` 来对读者进行计数。

读者优先算法描述如下：

```
semaphore wmutex=1;    // 用于写者与其他读者/写者互斥的访问共享数据
semaphore rmutex=1;    // 用于读者互斥的访问
int readcount=0;       // 读者计数器
cobegin
    procedure reader_i    //读者进程
    begin                //i=1,2,...
        P(rmutex);       //对共享变量 Readcount 操作的互斥信号量
```

```

Readcount++;          //读者数量加 1
if(readcount==1)      //若是第一个读者，需等待 wmutex 信号，封锁写者写
    P(wmutex);
V(rmutex);            //修改完 Readcount 信号量，释放互斥访问信号
读数据；
P(rmutex);            //读完数据，需要修改 Readcount 信号
Readcount--;          //读者数量减 1
if(readcount==0)      //如果当前已无读者在读，释放 wmutex，允许写者写
    V(wmutex);
V(rmutex);            //释放修改共享变量 Readcount 的互斥信号量
end
procedure Writer_j
begin                //j=1,2,...
    P(wmutex);        //看看有没有读进程，若没有读进程，则可写
    写数据；
    V(wmutex);        /*写完数据，释放 wmutex 信号，若有读者要进入
                        //临界区，唤醒一个读者进程进入临界区 */
end
coend

```

(2). 写者优先算法

写者优先算法的设计思想是一个写者到达时，如果有正在工作的读者，该写者只要等待正在工作的读者完成就可以进行写操作，而不必等候其后面到来的读者也完成它们的工作。

该算法当一个写者在等待时，后到达的读者是在写者之后被挂起，而不是立即允许进入。关于实现写者优先的算法，其实有很多种，我们可采用下面的算法来实现写者优先。

```

semaphore wmutex=1;    //写者访问 writecount 的互斥信号量
semaphore rmutex=1;    //读者访问 readcount 的互斥信号量
int readcount=0,writecount=0; // 读者写者计数器
semaphore rwmutex=1;    //读者、写者互斥访问文件
semaphore r=1;         //读者队列
semaphore read=1;       //一个读者与一个写者竞争访问文件
cobegin

```



```

procedure reader_i
begin          //i=1,2,...
repeat
    P(r);          //保证写者优先，若有写者写第二个之后读者在 r 上排队
    P(read);       //一个读进程与一个写进程在 read 上竞争
    P(rmutex);     //等待修改读者数量 readcount 的互斥信号
    readcount++;   //读者数量加 1
    if(readcount == 1) //第一个读进程需要判断是否有写进程在临界区,若有
                    //读进程等待，若无，阻塞写进程

        P(wmutex);

    V(rmutex);     //释放修改共享变量 readcount 的互斥信号量
    V(read);       //从 read 队列中唤醒一个进程
    V(r);
    读数据;
    P(rmutex);     //等待修改读者数量 readcount 的互斥信号
    readcount--;   //读者数量减 1
    if(readcount == 0) //最后一个离开临界区的读进程需要判断是否有
                    //写进程需要进入临界区，

        V(wmutex); //若有，唤醒一个写进程进临界区
    V(rmutex);     //释放互斥信号量
until false;
end

procedure Writer_j
begin          //j=1,2,...
repeat
    P(wmutex);     //开始对 writecount 共享变量进行互斥访问
    writecount++;  //写者数量加 1
    if(writecount==1) P(read); // 第一个写进程需要判断是否有读进程在临界区，
                    //若有，写进程阻塞，若无，阻塞新的读进程

    V(wmutex);     //释放互斥信号量
    P(rwmutex);    //其他写进程在 rwmutex 上排队

```

```

    写数据;
    P(wmutex);           //写完文件，离开登记
    writecount--;        //写者人数减 1
    if(writecount==0) V(read); //最后一个离开临界区的写进程需要判断是
                                //否有读进程需要进入，若有，
                                //唤醒一个读进程进入临界区
    V(wmutex);           //释放互斥信号量
until false;
end
Coend

```

4. (原书 第 12 题)理发师问题(Barber Problem)

问题描述：理发店理有一位理发师、一把理发椅和 n 把供等候理发的顾客坐的椅子如果没有顾客，理发师便在理发椅上睡觉一个顾客到来时，它必须叫醒理发师如果理发师正在理发时又有顾客来到，则如果有空椅子可坐，就坐下来等待，否则就离开。

【解析】利用解 PV 算法“五步曲”，可解本题如下：

第一步：找进程

- 理发师
- 等待理发的顾客（每个顾客的进程一样）

第二步：找操作

(1). 理发师

- （无顾客）睡觉
- （有顾客）给顾客理发

(2). 顾客

- （理发师在理发，且有空椅子则）等待理发
- （无空椅子）离开

第三步：找关系

- 有顾客等待时，理发师才能理发；
- 理发师空闲时，顾客才能理发。

第四步：设初值

- 用变量 waiting 用来记录等候理发的顾客数，初值均为 0；
- 用信号量 customers 来记录等候理发的顾客数，初值为 0；
- 信号量 barbers 用来记录正在等候顾客的理发师数（假设不止一个理发师），初值为 0；
- 信号量 mutex 用于互斥访问 waiting 变量，初值为 1。

算法如下：

```

int waiting=0;           //等候理发的顾客数
int chairs=n;           //为顾客准备的椅子数
semaphore customers=0,barbers=0,mutex=1;
cobegin
  barber ( )
    begin
      while(TRUE){
        P(customers);      //若无顾客,理发师睡眠
        P(mutex);          //进程互斥
        waiting:=waiting-1; //等候顾客数少一个
        V(barbers);         //理发师去为一个顾客理发
        V(mutex);          //开放临界区
        理发师给顾客理发;
      }
    end
  customer ( )
    begin
      P(mutex);//等待互斥访问 waiting 的信号量
      if(waiting<n) {
        waiting:=waiting+1; //等候顾客数加 1
        V(customers);       //必要的话唤醒理发师
        V(mutex);           //开放临界区
        P(barbers);         //无理发师,顾客坐着养神
        get-haircut();       //一个顾客坐下等理/
      }
    end
end

```

```
else V(mutex);           //人满了，不等了，直接走了  
  
end  
  
coend
```

本章到此就结束了，你觉得本章有什么地方不明白吗？任何问题，欢迎您与我们作者进行交流！



shareOurDreams
梦享团队微信号



weCSdream
梦享团队官方微信公众号



梦享论坛团队
梦享团队新浪微博

第三章 处理机调度和死锁

考点 1 调度的概念和基本准则

温馨提示：本考点考查处理机调度的基本概念和基本准则。请同学们掌握调度的基本概念、三级调度及其联系，调度的时机及其切换过程、调度的基本准则。

一. 选择题部分

1. (原书 第 3 题) () 是指从作业进入系统到作业完成所经过的时间间隔; () 是从作业进入后备队列起, 到被调度程序选中时的时间间隔。
- A. 响应时间 B. 周转时间 C. 运行时间 D. 等待时间

【解析】本题考查作业的周转时间和等待时间。

作业周转时间是指作业进入系统到作业完成所经过的时间间隔。我们常规的理解, 等待时间是指进程处于等待处理机的时间。响应时间则是指从用户提交请求到系统首次产生响应的的时间。作业从后备队列到作业被调度程序选中的时间间隔称为作业等待时间。故而, 答案依次为 B, D。

参考答案: B D

2. (原书 第 5 题) 作业调度是从输入井中处于 () 状态的作业中选取作业调入主存运行。
- A. 运行 B. 收容 C. 输入 D. 就绪

【解析】本题考查作业调度的基本概念。

一个作业进入系统到运行结束, 一般要经历进入、后备、运行和完成四个阶段, 相应地, 作业亦有进入、后备、运行和完成四种状态, 如图 3.1 所示。

(1). 提交状态: 作业的信息从输入设备上预输入到输入井, 此时称为作业处于进入状态。

(2). 后备(收容) 状态: 当作业的全部信息都已输入, 且由操作系统将其存放在输入

井中，此时称作业处于后备状态。系统将所有处于后备状态的作业组成后备作业队列，等待作业调度程序的调度。

(3). **运行状态**：一个后备作业被作业调度程序选中，分配了必要的资源，调入内存运行，称作业处于运行状态。

(4). **完成状态**：当作业正常运行完毕或因发生错误非正常终止时，作业进入这完成状态。

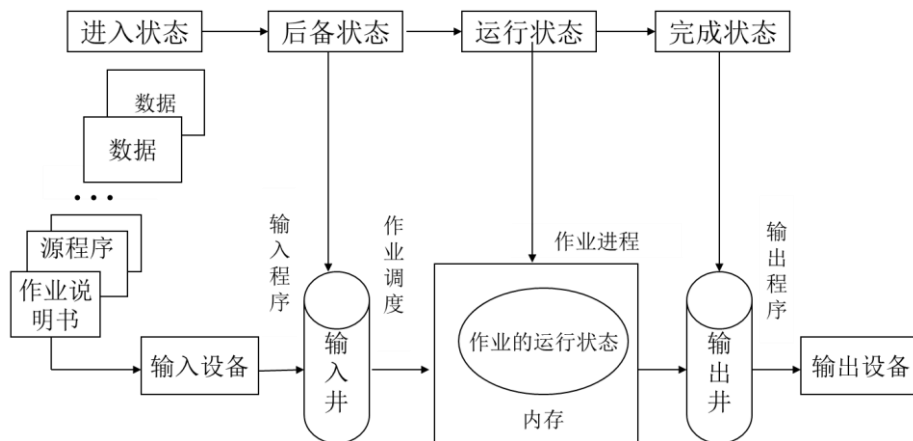


图 3.1 作业的四种状态与调度的关系

故而，本题选择 B 答案。

参考答案： B

3. (原书 第 7 题) 进程调度算法中的 () 属于抢夺式的分配处理器的策略。

- A. 时间片轮转算法
- B. 非抢占式优先数算法
- C. 先来先服务算法
- D. 分级调度算法

【解析】 本题考查进程调度策略，考生应能区分抢占式和非抢占式进程调度算法。

非抢占式的处理器分配策略，简单地理解就是，进程一旦抢占到处理机，除非它执行完或者因为某些原因放弃处理器，否则其他进程不得抢占处理机，中断其执行。抢占式则相反。

时间片轮转算法，占用当前处理机的进程时间片用完之后，必须强行切换，调度新的就绪队列进程执行，属于抢占式的处理器分配策略。故而，选择 A 答案。

【补充】 常见的调度算法中，FCFS 算法只能采用非抢占式调度方式，时间片轮转算法只能采用抢占式调度方式。短作业优先算法和高优先权算法即可采用非抢占式调度方式，也可采用抢占式调度方式。

参考答案： A

4. (原书 第 11 题) 设系统中有 P1、P2、P3 三个进程，并按 P1、P2、P3 的优先次序调度运行，它们的内部计算和 I/O 操作时间如下：

P1：计算 60 ms—I/O 80 ms—计算 20 ms

P2：计算 120 ms—I/O 40ms—计算 40ms

P3：计算 40 ms—I/O 80ms—计算 40ms

设调度程序执行时间忽略不计，完成这三个进程比单道运行节省的时间是（ ）。

- A. 140ms B. 160ms C. 170ms D. 180ms

【解析】三个进程并发执行的运行情况图，如图 3.2 所示。

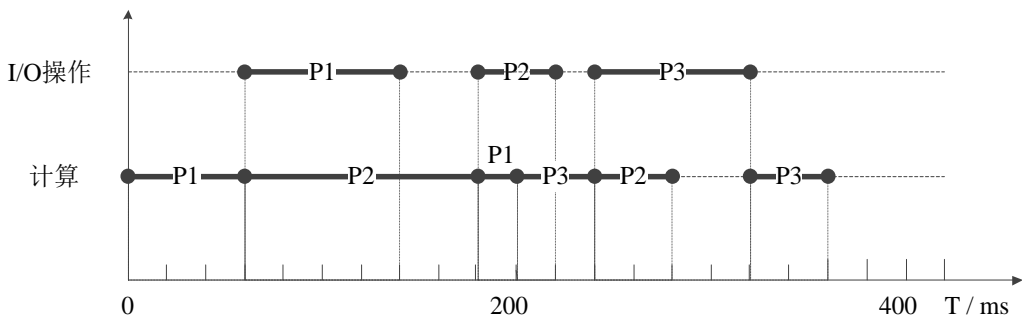


图 3.2 进程 P1、P2、P3 并发执行的情况

由图 3.2 可知，三个进程并发执行，可以在 $T=360\text{ms}$ 时全部完成。若是单道运行，则所需时间为 $T_1=60\text{ms}+80\text{ms}+20\text{ms}+120\text{ms}+40\text{ms}+40\text{ms}+40\text{ms}+80\text{ms}+40\text{ms}=520\text{ms}$ 。故而三个进程并发执行比单道运行节省了 $520\text{ms}-360\text{ms}=160\text{ms}$ 。故而选择 B 答案。

参考答案： B

二. 综合应用题部分

1. (原书 第 2 题) 试说明作业和进程的状态，并说明引起状态转换的典型原因。

【解析】作业的状态有提交状态、后备状态、运行状态和完成状态。

当一个作业还未进入计算机系统内之前，用户正在向系统提交作业时所处的状态叫做提交状态。

送入计算机并由操作系统将其存放在磁盘上以后，系统为这个作业建立一个作业控制块，并把它插入到后备队列中等待调度运行。此时，这个作业所处的状态称为后备状态。

当这个作业被作业调度程序选中，并为它分配了必要的资源，建立了一组相应的进程之后，这个作业就由后备状态变为运行状态。

进程有三种状态：就绪状态、执行状态和阻塞状态。

- 刚建立的进程处于就绪状态；
- 就绪状态的进程被进程调度程序选中，获得处理机运行，此时的状态称为执行状态；
- 对于执行状态的进程，当它使用完分配给它的时间或被更高优先权的进程剥夺处理机后，又回到就绪状态，等待下次调度；
- 进程在执行中若发生了某些事件而暂时无法执行下去，如有输入输出请求并等待输入输出完成，则进入阻塞状态；
- 当引起进程阻塞的事件消失时，如输入输出完成，进程由阻塞状态变为就绪状态，重新获得被调度的资格。
- 当作业正常运行结束或因发生错误而异常结束时，作业进入完成状态。此时，由操作系统将作业控制块从当前作业队列中删除，并收回其所占用的资源。

2. (原书 第 3 题) 某个采用多道程序设计的计算机系统配有输入机和打印机各一台，现有程序 A 和程序 B 并行执行，且程序 A 先开始 50ms。假定程序 A 的执行过程为：计算 50ms，打印 100ms，再计算 50ms，打印 100ms，结束；程序 B 的执行过程为：计算 50ms，输入数据 60ms，再计算 50ms，打印 100ms，结束。当忽略调度和启动外设等所花费的时间时，回答下列问题：

- (1). 用实线画出程序 A 和程序 B 并发执行时各自使用 CPU 与外设的情况图。
- (2). 在程序开始执行直到两道程序都执行结束时，处理器的利用率是_____。
- (3). 程序 B 从开始执行直到结束实际花费的时间是_____。

【解析】

- (1). 程序 A 与程序 B 并发执行时各自使用 CPU 与外设的情况图如图 3.3 所示。

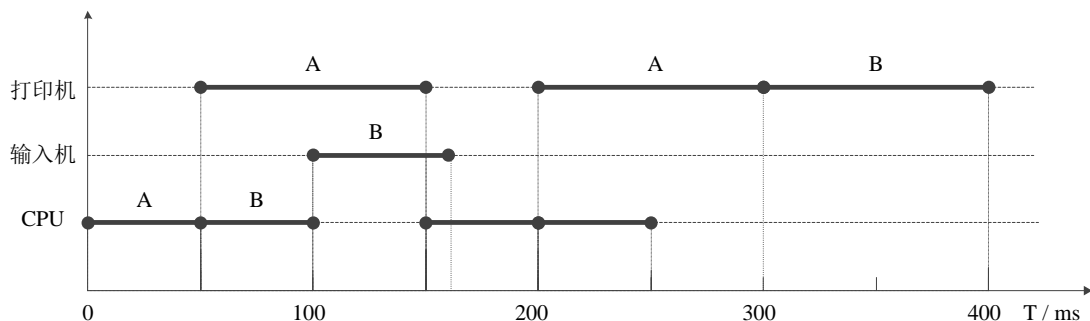


图 3.3 程序 A、B 并发执行时各自使用 CPU 与外设的情况图

(2). 处理机的利用率，我们可以简单理解成，作业占用 CPU 的时间与作业从开始运行到结束的时间之比。本题中，处理器的利用率是 $(50+50+50+50)/400=50\%$ 。

(3). 程序 B 从开始执行直到结束实际花费的时间是 $400-50=350\text{ms}$ 。

考点 2 典型的处理机调度算法

温馨提示：本考点主要考查典型的处理机调度算法，包括先来先服务算法、短作业优先算法、优先级调度算法、高响应比优先算法、时间片轮转算法以及多级反馈队列调度算法。本考点十分重要，请同学一定要掌握这几种处理机调度方法，并会用于解题。

一. 选择题部分

1. (原书 第 1 题) 我们如果为每一个作业只建立一个进程，则为了照顾短作业用户，应采用 ()；为了照顾紧急作业的用户，应采用 ()；为能实现人机交互作用应采用 ()；为了兼顾短作业和长时间等待的作业，应采用 ()；为了使短作业、长作业及交互作业用户都比较满意，应采用 ()；为了使作业的平均周转时间最短，应采用 () 算法。
- | | |
|-----------------|---------------|
| A. FCFS 调度算法 | B. 短作业优先 |
| C. 时间片轮转法 | D. 多级反馈队列调度算法 |
| E. 基于优先权的剥夺调度算法 | F. 高响应比优先 |

【解析】 本题考查进程调度算法，考生应能掌握典型进程调度算法的特点。本题是一个总结题，我们总结如下：

(1). **短作业优先算法**有利于短作业而不利于长作业。可能致使长作业出现“饥饿”现象。短作业优先算法使得平均周转时间最短。

(2). **FCFS 算法**对短进程不公平,当长进程排在就绪队列的前面时必将增加后面短小进程的等待时间,从而增加系统的平均周转时间。**FCFS 算法**有利于 I/O 型作业,不利于 CPU 繁忙型作业。

(3). 交互系统如我们常见的分时系统,采用的是**时间片轮转算法**。时间片足够大时,则该算法退化成 **FCFS 算法**。时间片太小,系统切换进程的开销大。影响时间片长短设置的因素为系统的响应时间、就绪队列的进程个数、系统的处理能力等。

(4). **最高响应比优先算法**是对 **FCFS 算法**与 **SJF** (短作业优先) 算法的一种综合平衡。响应比的计算公式为

$$\text{响应比} = (\text{等待时间} + \text{允许时间}) / \text{运行时间}$$

从公式看,该算法有利于短作业,同时兼顾长作业,只要某长进程等待了足够长时间,总会成为最高响应比者而被调度执行。

(5). **多级反馈队列算法**,如图 3.4 所示,其基本思想如下:

1) 系统中设置多个就绪队列对应不同优先级,第一个队列优先级最高,其余逐次降低。

2) 每个队列进程执行的时间片各不相同,从第一个到第 n 个队列时间片长短依次加倍。

3) 新进程进入先放入第一级队列的末尾,按 **FCFS** 排队等待调度,被选中执行时,若可在一个对应时间片完成,则撤销,否则转入第二级队列末尾,如此下去,最后一级队列使用时间片轮转调度算法。

4) 仅当第 1 到 $i-1$ 个队列均为空时,才会调度第 i 个队列的进程。当 CPU 正处理第 i 个队列中某进程,若又有新进程进入,则新进程抢占 CPU,正执行进程放回第 i 个队列末尾。

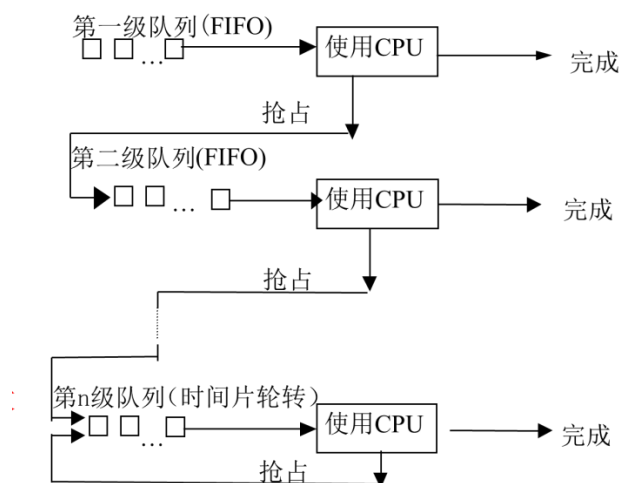


图 3.4 多级反馈队列

多级反馈队列因为以上优点，使得短作业、长作业和交互作业用户都比较满意。

(6). **基于优先权的剥夺调度算法**。对于比较紧急的作业，可赋予作业较高的优先级，并可抢占处理机执行。

故而，本题的答案依次为 B,E,C,F,D,B。

【故事助记】假设医院只有一个医疗窗口，一群人排队按照先来先看病的规则依次看病，这就是 FCFS。但是，有些人就抱怨了，有的人看个病，磨磨唧唧问这问那的，要一个多小时，我就一个小感冒，几分钟就可以了，我应该先看，要不然我等待的时间太长了。

于是，医院想出了一个办法：看病时间短的先看（短作业优先），看病时间比较长的后看。可是，看病时间比较长的人就不愿意了，要是一直有看病时间比自己短的陆陆续续来看病，那要等到什么时候才能轮到我在啊？

医院想了想，有道理啊，于是想出了高响应比优先的办法，有利于看病时间比较短的病人不用等待太长时间就能看病，而且看病时间长的病人在等待了足够长的时间之后，响应比达到最高，也能看病。

可是，还是有问题，大家如果看的都是小病，突然有人推着一个快挂了病人进医院了，你说他都快挂了，你还让他先去排队，等到在他到来之前的所有人都看完病再给他看病吧，不科学。你说让看病时间短的先看病，他是大病，看病时间应该不会短，这也不科学。你说要是按照响应比，要是等待看病的人多，他也得等很长时间，这显然也不科学。

二. 综合应用题部分

1. (原书 第 3 题) 设有三道作业，它们的提交时间及执行时间由表 3.3 给出：

表 3.3 作业 1、2、3 以及它们的提交时间、运行时间

作业号	提交时间	执行时间
1	8.5	2.0
2	9.2	1.6
3	9.4	0.5

试计算在单道程序环境下，采用先来先服务调度算法和最短作业优先调度算法时的平均周转时间（时间单位：小时，以十进制进行计算；要求写出计算过程）。

【解析】

(1). 在单道程序环境下，采用先来先服务调度算法，作业 1、2、3 分别在 8.5、9.2 和 9.4 时刻到达，这三道作业被调度执行的顺序为作业 1→作业 2→作业 3。这 3 道作业的运行情况如表 3.4 所示。

表 3.4 先来先服务调度算法下三道作业的执行情况

作业号	提交时间	执行时间	开始时间	完成时间	周转时间
1	8.5	2.0	8.5	10.5	2.0
2	9.2	1.6	10.5	12.1	2.9
3	9.4	0.5	12.1	12.6	3.2

从表 3.10 可知，这 3 道作业的周转时间分别为 2.0、2.9 和 3.2，它们的平均周转时间 $= (2.0 + 2.9 + 3.2) / 3 = 2.7$ (小时)。

(2). 短作业优先算法在多个作业同时等待被调度执行时，按照作业长短调度最短的作业执行的算法。为什么要强调有多个作业等待调度执行？如果内存中随时都只有一道作业，那么所有算法都将变态成 FCFS 算法了。

在 8.5 时刻，只有作业 1 提交了，所以直接调度作业 1 运行。在 10.5 时刻，作业 1 完成，此时作业 2 和作业 3 均已在内存中。此时，内存中同时逗留了两个作业，必须选择运行时间较短的作业调度运行。作业 2 的执行时间是 1.6，作业 3 的执行时间是 0.5，显然作业 3 比作业 2 短，系统调度作业 3 执行。在 11.0 时刻，作业 3 执行完毕，系统调度作业 2 执行，在 12.6 时刻作业 2 执行完毕。

综上所述,采用最短作业优先调度算法,作业的运行情况如表 3.5 所示。

表 3.5 最短作业优先调度算法下三道作业的执行情况

作业号	提交时间	执行时间	开始时间	完成时间	周转时间
1	8.5	2.0	8.5	10.5	2.0
2	9.2	1.6	11.0	12.6	3.4
3	9.4	0.5	10.5	11.0	1.6

3 道作业的平均周转时间 $= (2.0+3.4+1.6)/3=2.3$ (小时)。

2. (原书 第 4 题)假设有 4 道作业,它们提交的时刻及执行时间由表 3.6 给出,计算在单道程序环境下,采用先来先服务调度算法、最短作业优先算法、最高响应比优先算法的平均周转时间和平均带权周转时间,并指出它们的调度顺序。

表 3.6 4 道作业的提交时间和运行时间

作业	提交时间	运行时间
1	8.0	2.0
2	8.5	0.5
3	9.0	0.1
4	9.5	0.2

【解析】

(1). 作业 1、2、3 和 4 分别在 8.0、8.5、9.0 和 9.5 时刻被提交,按照先来先服务算法,作业的执行顺序与提交顺序一致。所以作业的运行顺序为

作业 1→作业 2→作业 3→作业 4

采用先来先服务调度算法,四道作业的运行情况如表 3.7 所示。

表 3.7 先来先服务调度算法下四道作业的执行情况

作业	提交时间	运行时间	开始时间	完成时间	周转时间	带权周转时间
1	8.0	2.0	8.0	10.0	2.0	1.0
2	8.5	0.5	10.0	10.5	2.0	4.0
3	9.0	0.1	10.5	10.6	1.6	16.0
4	9.5	0.2	10.6	10.8	1.3	6.5

由表 3.12 可知, 作业 1、2、3 和 4 的周转时间是 2.0、2.0、1.6 和 1.3, 平均周转时间 $T = (2.0+2.0+1.6+1.3)/4=1.725$ 。这四道作业的带权周转时间分别为 1.0、4.0、16.0 和 6.5, 平均带权周转时间 $W = (1.0+4.0+16.0+6.5)/4=6.875$ 。

(2). 采用短作业优先算法, 4 道作业的运行情况分析如下:

- ① 在 8.0 时刻, 作业 1 提交即被调度执行, 在 10.0 时刻作业 1 完成。
 - ② 在 10.0 时刻, 作业 2、3、4 都已经提交, 而且作业运行时间分别为 0.5、0.1 和 0.2, 按照短作业优先算法, 调度作业 3 执行, 该作业在 10.1 时刻执行完毕。
 - ③ 在 10.1 时刻, 作业 2 和 4 都在等待执行, 作业 4 的执行时间比作业 2 的执行时间要短, 所以作业 4 被调度执行。
 - ④ 在 10.3 时刻, 作业 4 执行完毕, 作业 2 被调度执行, 该作业在 10.8 时刻执行完毕。
- 综上分析, 四道作业的运行情况如表 3.8 所示。

表 3.8 最短作业优先调度算法下四道作业的执行情况

作业	提交时间	运行时间	开始时间	完成时间	周转时间	带权周转时间
1	8.0	2.0	8.0	10.0	2	1
2	8.5	0.5	10.3	10.8	2.3	4.6
3	9.0	0.1	10.0	10.1	1.1	11
4	9.5	0.2	10.1	10.3	0.8	4

由表 3.13 可知, 四道作业的周转时间分别为 2、2.3、1.1 和 0.8, 它们的平均周转时间 $T = (2.0+2.3+1.1+0.8)/4=1.55$, 作业平均带权周转时间 $W = (1+4.6+11+4)/4=5.15$ 。

(3). 最高响应比优先算法的响应比为

$$\begin{aligned} \text{响应比 } R_p &= \text{作业响应时间} / \text{运行时间} = \text{作业等待时间} + \text{作业运行时间} \\ &= 1 + \text{作业等待时间} / \text{作业运行时间} \end{aligned}$$

我们采用最高响应比算法来分析这 4 道作业的运行情况:

- ① 在 8.0 时刻, 只有作业 1 被提交, 作业 1 可以直接调度运行。
- ② 在 10.0 时刻, 作业 1 执行完毕, 系统中有作业 2、3 和 4, 需要通过计算响应比来决定调度执行哪一个作业。作业 2 的响应比

$$R_p(2) = 1 + \frac{10.0 - 8.5}{0.5} = 4$$

作业 3 的响应比为

$$R_p(3) = 1 + \frac{10.0 - 9.0}{0.1} = 11$$

作业 4 的响应比为

$$Rp(4) = 1 + \frac{10.0 - 9.5}{0.2} = 3.5$$

因而调度作业 3 执行。

③ 在 10.1 时刻, 作业 3 执行完毕, 作业 2 的响应比变成

$$Rp(2) = 1 + \frac{10.1 - 8.5}{0.5} = 4.2$$

作业 4 的响应比为

$$Rp(4) = 1 + \frac{10.1 - 9.5}{0.2} = 4$$

所以调度作业 2 先执行, 最后执行作业 4。

综上所述, 采用最高响应比优先算法, 四道作业的运行情况如表 3.9 所示。

表 3.9 最高响应比优先调度算法下四道作业的执行情况

作业	提交时间	运行时间	开始时间	完成时间	周转时间	带权周转时间
1	8.0	2.0	8.0	10.0	2	1
2	8.5	0.5	10.1	10.6	2.1	4.2
3	9.0	0.1	10.0	10.1	1.1	11
4	9.5	0.2	10.6	10.8	1.3	6.5

由表 3.9 可知, 4 道作业的平均周转时间 $T=(2.0+2.1+1.1+1.3)/4=1.625$, 平均带权周转时间 $W=(1.0+4.2+11+6.5)/4=6.5$ 。

3. (原书 第 7 题) 假设在单处理机上有五个 (A,B,C,D,E) 进程争夺运行, 其运行时间分别为 10、1、2、1、5 (秒), 其优先级分别为 4、1、3、5、2; 在某时刻这五个进程按照 A,B,C,D,E 的顺序同时到达。试回答:

- (1). 给出这些进程分别使用轮转法 (时间片为 2 秒)、非剥夺优先级调度法时的运行进度表。
- (2). 在上述各算法的调度下每个进程的周转时间和等待时间为多少?

【解析】

(1). 按照时间片轮转算法, 五个进程第一轮每个进行都运行一个时间片。但是, 有两个进程的执行时间不够一个时间片, 执行完毕之后可以切换到其他进程。

① 第一轮执行完毕之后, 进程 B、C、D 运行完毕退出系统, 进程 A、E 仍需要分别

再执行 8 秒和 3 秒。

② 第二轮执行时，给进程 A、E 分别执行 1 个时间片。第二轮执行结束之后，进程 A 还需要执行 6 秒，进程 E 还需要执行 1 秒。

③ 第三轮执行时，给进程 A 执行一个时间片，进程 E 只需执行 1 秒即可完成。第三轮结束以后，只有进程 A 尚未执行完毕，仍需要 4 秒的执行时间。

④ 第四轮和第五轮仅有进程 A 在执行，在第五轮执行结束，所有进程都完成了。

综上所述，采用时间片轮转算法时，五个作业的调度运行如图 3.5 所示。

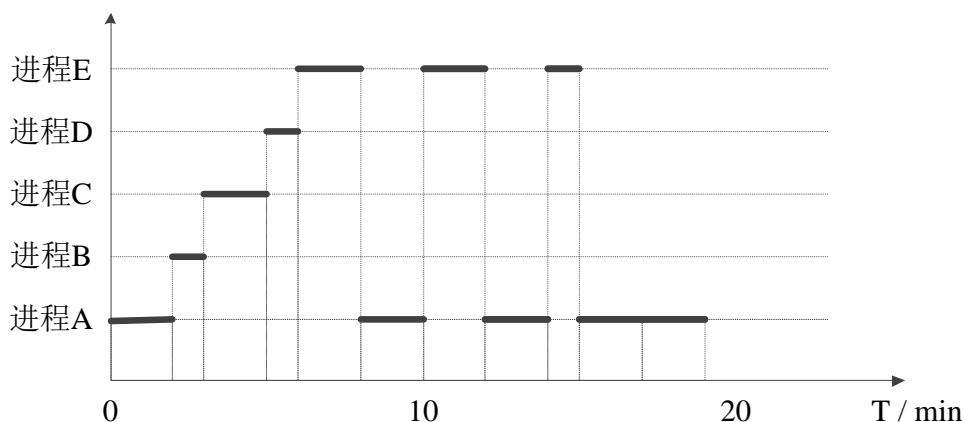


图 3.5 时间片轮转算法下五个作业的调度运行情况

而采用非剥夺优先级调度法时，五个进程的调度情况分析如下：

① 五个进程依次达到，进程 D 的优先级最高，因而先调度进程 D 运行。

② 进程 D 运行完毕之后，进程 A 的优先级比 B、C、E 的优先级高，系统调度进程 A 执行。

③ 在进程 A 运行完毕之后，B、C、E 三个进程 C 的优先级最高，系统调度进程 C 运行。

④ 进程 C 运行完毕之后，B、E 中 E 的优先级较高，系统调度进程 E 运行。E 运行结束之后，进程 B 再调度执行。

综上所述，五个进程的调度运行情况如图 3.6 所示。

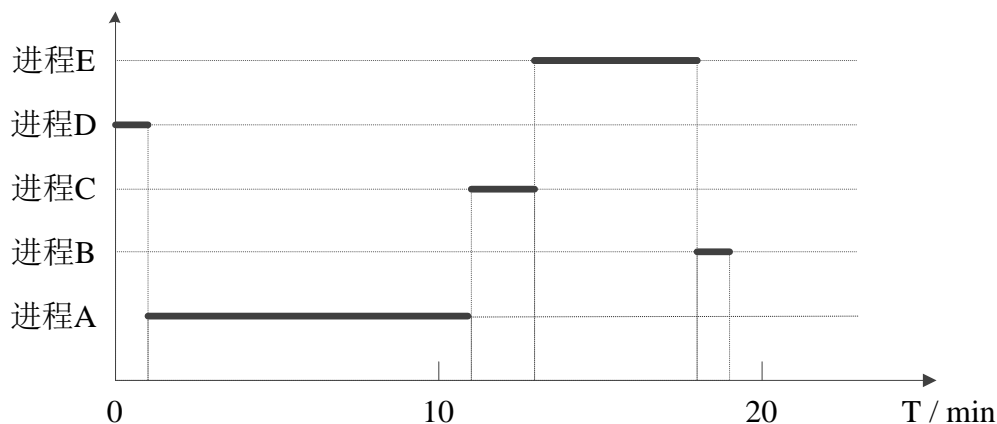


图 3.6 非剥夺优先级调度算法下五个作业的调度运行情况

(2). 根据图 3.6, 可得到 5 个作业在时间片轮转算法下的等待时间和周转时间如表 3.10 所示。

表 3.10 时间片轮转算法下五道作业的调度运行情况

作业	运行时间 (小时)	周转时间 (小时)	等待时间 (小时)
1	10	19	9
2	1	3	2
3	2	5	3
4	1	6	5
5	5	15	10

根据图 3.6, 可得 5 个作业在非剥夺优先级调度法下的周转时间和等待时间如表 3.11 所示。

3.11 非剥夺算法下 5 个作业的周转时间和等待时间

作业	优先级	调度顺序	运行时间	周转时间	等待时间
1	4	2	10	11	1
2	1	5	1	19	18
3	3	3	2	13	11
4	5	1	1	1	0
5	2	4	5	18	13

考点 3 死锁

温馨提示：本考点考查进程间的死锁问题，请同学们掌握死锁的定义、死锁产生的原因，以及死锁预防、避免和检测方法。下一个考点，我们将详细讲解死锁避免中的银行家算法。

一. 选择题部分

1. **【原书 第 2 题】**某系统中有 3 个并发进程，都需要同类资源 4 个，则该系统不会发生死锁的最少资源数是（ ）个。

A. 9 B. 10 C. 11 D. 12

【解析】三个并发进程，每个进程需要同类资源 4 个。一种资源分配最大数量但仍然死锁的情况是，每个进程分配了 3 个该类资源，都只差一个资源即可运行。为了破除这种死锁，需要再增加一个额外的资源，共需要 10 个资源，系统才不会发生死锁。故而，选择 B 答案。

参考答案：B

2. **【原书 第 5 题】**采用资源剥夺法可解除死锁，还可以采用（ ）方法解除死锁。

A. 执行并行操作 B. 撤销进程
C. 拒绝分配新资源 D. 修改信号量

【解析】本题考查解除死锁的办法。

死锁的解除，通常有两种方法，一种是剥夺资源法，另一种是撤销进程。撤销进程，指的是采用强制手段从系统中撤销一个或部分死锁进程，并剥夺这些进程的资源供其他死锁进程使用。而剥夺资源法，是指当法师死锁后，从其他进程中剥夺足够数量的资源给死锁进程，以解除死锁状态的办法。

清华大学出版社出版的操作系统教材，对死锁的预防、避免、检测做了一个详细的总结，如表 3.12、3.13 所示，我们来看看。

表 3.12 死锁的预防、避免、检测比较

方法	资源分配策略	各种可能模式	主要优点	主要缺点
死锁预防	保守的，宁可资源闲置	一次请求所有资源；资源剥夺；资源按序申请	适用于突发式处理的进程；不比剥夺；适用于状态可以保存和	效率低；进程初始化时间延长；剥夺次数过多；多次

【解析】本题考查资源分配图中存在环路与死锁的关系。

利用分配图化简判定死锁的基本法则如下：

- (1). 若图中没有环路，则图是安全状态，即一定不存在死锁。
 - (2). 若图中有环路，而环路中各个资源均为一个单位资源，则图是死锁状态(充分必要条件)。
 - (3). 若图中有环路，但环路中各个资源不全为一个单位数，则不一定死锁。
- 由(2)、(3)可知，资源分配图中有环路，系统可能存在死锁，但也有可能并不存在死锁。故而选择 D 答案。

参考答案：D

二. 综合应用题部分

1. (原书 第 3 题)简述解决死锁问题的三种方法。

【解析】解决死锁问题，有以下三种办法。

- (1). **死锁的防止**。系统按预定的策略为进程分配资源，这些分配策略能使死锁的四个必要条件之一不成立，从而使系统不产生死锁。
- (2). **死锁的避免**。系统动态地测试资源分配情况，仅当能确保系统安全时才给进程分配资源。
- (3). **死锁的检测**。对资源的申请和分配不加限制，只要有剩余的资源就呆把资源分配给申请者，操作系统要定时判断系统是否出现了死锁，当有死锁发生时设法解除死锁。

2. (原书 第 4 题)假定某系统当时的资源分配图如图 3.7 所示：

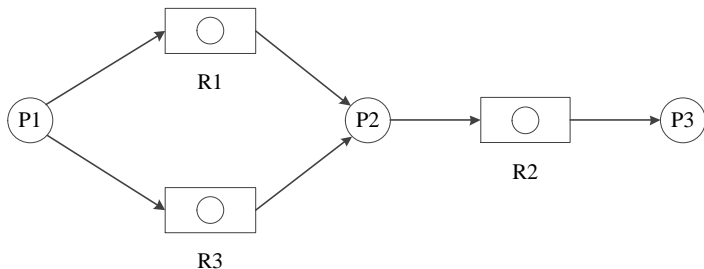


图 3.7 某系统的资源分配图

- (1). 分析当时系统是否存在死锁。
- (2). 若进程 P_3 再申请 R_3 时，系统将发生什么变化，说明原因。

【解析】

(1). 从图 3.7 可知, 因为当时系统的资源分配图中不存在环路, 所以不存在死锁。

(2). 当进程 P_3 申请资源 R_3 后, 资源分配图中引成环路 $P_2 \rightarrow R_2 \rightarrow P_3 \rightarrow R_3 \rightarrow P_2$, 而 R_2, R_3 都是单个资源的类, 该环路无法消除, 所以进程 P_2, P_3 永远处于等待状态, 从而引起死锁。

考点 4 系统安全状态、银行家算法

温馨提示: 本考点主要考查系统的安全状态和银行家算法。银行家算法是本章的一个重要考点, 请同学们务必能够根据进程资源的分配情况以及请求序列, 推导安全序列 (或死锁)。

一. 选择题部分

1. (原书 第 1 题) 用银行家算法避免死锁时, 检测到 () 时才分配资源。
- A. 进程首次申请资源时对资源的最大需求量超过系统现存的资源量
 - B. 进程已占用的资源数与本次申请资源数之和超过对资源的最大需求量
 - C. 进程已占用的资源数与本次申请的资源数之和不超过对资源的最大需求量, 且现存资源能满足尚需的最大资源量
 - D. 进程已占用的资源数与本次申请的资源数之和不超过对资源的最大需求量, 且现存资源能满足本次申请量, 但不能满足尚需的最大资源量

【解析】 我们可以把操作系统看作是银行家, 操作系统管理的资源相当于银行家管理的资金, 进程向操作系统请求分配资源相当于用户向银行家贷款。

为保证资金的安全, 银行家规定:

- (1). 当一个顾客对资金的最大需求量不超过银行家现有的资金时就可接纳该顾客;
- (2). 顾客可以分期贷款, 但贷款的总数不能超过最大需求量;
- (3). 当银行家现有的资金不能满足顾客尚需的贷款数额时, 对顾客的贷款可推迟支付, 但总能使顾客在有限的时间里得到贷款;
- (4). 当顾客得到所需的全部资金后, 一定能在有限的时间里归还所有的资金。

由此可得到一个结论：当进程已占有的资源数与本次申请的资源数之和不超过对资源的最大需求量，而且现有资源能够满足尚需的最大资源量时，可以分配资源。事实上，在本部分的大题里面，我们的解题过程这句话都有体现。

参考答案：C

2. (原书 第 2 题)某系统采用了银行家算法，则下列叙述正确的是 ()。
- A. 系统处于不安全状态时一定会发生死锁
 - B. 系统处于不安全状态时可能会发生死锁
 - C. 系统处于安全状态时可能会发生死锁
 - D. 系统处于安全状态时一定会发生死锁

【解析】采用银行家算法，则系统处于不安全状态不一定发生死锁，死锁一定是不安全状态，而处在安全状态的系统一定不发生死锁。故而，本题选择 B。

参考答案：B

二. 综合应用题部分

1. (原书 第 1 题)某系统有 R1、R2 和 R3 共三种资源，在 T0 时刻 P1、P2、P3 和 P4 这 4 个进程对资源的占用和需求情况如下表 3.14 所示，此时系统的可用资源向量为 (2, 1, 2)。

表 3.14 T0 时刻资源分配情况

进程	最大资源需求量			已分配资源数量		
	R1	R2	R3	R1	R2	R3
P1	3	2	2	1	0	0
P2	6	1	3	4	1	1
P3	3	1	4	2	1	1
P4	4	2	2	0	0	2

- (1). 将系统中各种资源总数和此刻各进程对各资源的需求数目用向量或矩阵表示出来；
- (2). 如果此时 P1 和 P2 均发出资源请求向量 request (1, 0, 1)，为了保证系统的安全性，应该如何分配资源给这两个进程？说明你所采用策略的原因。

【解析】系统中资源总量为 (9, 3, 6)，各进程对资源的需求量为：

$$\begin{bmatrix} 2 & 2 & 2 \\ 2 & 0 & 2 \\ 1 & 0 & 3 \\ 4 & 2 & 0 \end{bmatrix}$$

若此时 P1 发出资源请求 Request1 (1, 0, 1)，按银行家算法进行检查：

$$\text{Request1} (1, 0, 1) \leq \text{Need1}(2,2,2)$$

$$\text{Request1} (1, 0, 1) \leq \text{Available}(2,1,2)$$

试分配并修改相应数据结构，资源分配情况如表 3.15 所示。

表 3.15

资源 进程	Allocation			Need			Available		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P2	2	0	1	1	2	1	1	1	1
P3	4	1	1	2	0	2			
P4	2	1	1	1	0	3			
P1	0	0	2	4	2	0			

再利用安全性算法检查系统是否安全，可用资源 Available(1,1,1)已不能满足任何进程，故系统进入不安全状态，此时系统不能将资源分配给 P1。

若此时 P2 发出资源请求 request2 (1, 0, 1)，按银行家算法进行检查：

$$\text{Request2} (1, 0, 1) \leq \text{need2}(2,0,2)$$

$$\text{Request2} (1, 0, 1) \leq \text{available}(2,1,2)$$

试分配并修改相应数据结构，资源分配情况如表 3.15 所示。

表 3.15

资源 进程	Allocation			Need			Available		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P2	1	0	0	2	2	2	1	1	1
P3	5	1	2	1	0	1			
P4	2	1	1	1	0	3			
P1	0	0	2	4	2	0			

再利用安全性算法检查系统是否安全，可得此时刻的安全性分析情况如表 3.16 所示。

表 3.16 利用银行家算法寻找到的安全序列

资源 进程	Work			Need			Allocation			Work+Allocation			Finish
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3	
P2	1	1	1	1	0	1	5	1	2	6	2	3	true
P3	6	2	3	1	0	3	2	1	1	8	3	4	true
P4	8	3	4	4	2	0	0	0	2	8	3	6	true
P1	8	3	6	2	2	2	1	0	0	9	3	6	true

从上述分析中可以看出，此时存在一个安全序列（P2,P3,P4,P1），故该状态是安全的，可以立即将 P2 所申请的资源分配给它。

2. （原书 第 4 题）假定系统中有四个进程 P1、P2、P3、P4，三种类型的资源 R1、R2、R3，数量分别为 9、3、6，在 T0 时刻的资源分配情况如表 3.17 所示：

表 3.17 T0 时刻资源分配情况

资源 进程	Max			Allocation			Need			Available		
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3
P2	3	2	2	1	0	0	2	2	2	1	1	2
P1	6	1	3	5	1	1	1	0	2			
P3	3	1	4	2	1	1	1	0	3			
P4	4	2	2	0	0	2	4	2	0			

- (1). 检查 T0 时刻的安全性
- (2). 若 P2 发出请求向量 Request2 (1, 0, 1)，系统可否满足？
- (3). 若 P1 发出请求向量 Request1 (1, 0, 1)，系统可否满足？
- (4). 若 P3 发出请求向量 Request3 (0, 0, 1) 系统可否满足？

【解析】

- (1). T0 时刻存在一个安全序列{P2、P1、P3、P4}，所以 T0 时刻是安全的。

表 3.18 T0 时刻的一个安全序列

资源 进程	Work			Need			Allocation			Work+Allocation			Finish
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3	
P2	1	1	2	1	0	2	5	1	1	6	2	3	true
P1	6	2	3	2	2	2	1	0	0	7	2	3	true
P3	7	2	3	1	0	3	2	1	1	9	3	4	true
P4	9	3	4	4	2	0	0	0	2	9	3	6	true

(2). P2 请求资源。

P2 发出请求向量 Request(1,0,1),系统按银行家算法进行检查:

Request(1,0,1) <= Need(1,0,2)

Request(1,0,1) <= Available(1,1,2)

系统先假定可为 P2 分配资源,并修改 Available,Allocation,Need 向量,由此形成的资源变化情况如表 3.19 所示。

表 3.19 P2 申请资源的安全性审查初始资源情况表

资源 进程	Work			Need			Allocation			Work+Allocation		
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	3	3	2	1	0	0	2	2	2	0	1	1
P2	6	1	3	6	1	2	0	0	1			
P3	3	1	4	2	1	1	1	0	3			
P4	4	2	2	0	0	2	4	2	0			

我们再利用安全性算法检查此时系统是否安全,可得表 3.20 所示的安全性分析。

表 3.20 对 P2 申请资源的安全性审查

资源 进程	Work			Need			Allocation			Work+Allocation			Finish
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3	
P2	0	1	1	0	0	1	6	1	2	6	2	3	true
P1	6	2	3	2	2	2	1	0	0	7	2	3	true
P3	7	2	3	1	0	3	2	1	1	9	3	4	true
P4	9	3	4	4	2	0	0	0	2	9	3	6	true

由所进行的安全性检查得知,可以找到一个安全序列{P2,P1,P3,P4},因此,系统是安全的,可以立即将 P2 所申请的资源分配给它。

(3). P1 请求资源。

P1 发出请求向量 $\text{Request}(1,0,1)$, 系统按银行家算法进行检查:

$\text{Request}(1,0,1) \leq \text{Need}(2,2,2)$ 系统按银行家算法进行检查;

$\text{Request}(1,0,1) > \text{Available}(0,1,1)$, 让 P1 等待。

(4). P3 请求资源。

P3 发出请求向量 $\text{Request}(0,0,1)$, 系统按银行家算法进行检查:

$\text{Request}(0,0,1) \leq \text{Need}(1,0,3)$

$\text{Request}(1,0,1) \leq \text{Available}(0,0,1)$

系统先假定可为 P3 分配资源, 并修改有关数据, 如表 3.21 所示。

表 3.21 P3 申请资源的安全性检查

资源 进程	Max			Allocation			Need			Available		
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	3	2	2	1	0	0	2	2	2	0		1
P2	6	1	3	6	1	2	0	0	1	0		
P3	3	1	4	2	1	2	1	0	2			
P4	4	2	2	0	0	2	4	2	0			

3. (原书 第 8 题) 某系统有 R1、R2 和 R3 共 3 种资源, 在 T0 时刻 P1、P2、P3 和 P4 这 4 个进程对资源的占用和需求情况见表 3.22, 此时系统的可用资源向量为(2, 1, 2), 试问:

(1). 将系统中各种资源总数和此刻各进程对各资源的需求数目用向量或矩阵表示出来;

(2). 如果此时 P1 和 P2 均发出资源请求向量(1, 0, 1), 为了保证系统的安全性, 应该如何分配资源给这两个进程? 说明你所采用策略的原因。

(3). 如果(2)中两个请求立即得到满足后, 系统此时是否处于死锁状态?

表 3.22 T0 时刻 4 个进程对资源的占用和需求情况

	最大资源需求量			已分配资源量		
	R1	R2	R3	R1	R2	R3
P1	3	2	2	1	0	0
P2	6	1	3	4	1	1
P3	3	1	4	2	1	1
P4	4	2	2	0	0	2

【解析】

(1). 系统中资源总数是可用资源数与各进程已分配资源数之和, 即

$$(2, 1, 2) + (1, 0, 0) + (4, 1, 1) + (2, 1, 1) + (0, 0, 2) = (9, 3, 6)$$

各进程对各资源的需求量为 Max 与 Allocation 之差, 即

$$\begin{bmatrix} 3 & 2 & 2 \\ 6 & 1 & 3 \\ 3 & 1 & 4 \\ 4 & 2 & 2 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 1 \\ 2 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 2 \\ 2 & 0 & 2 \\ 1 & 0 & 3 \\ 4 & 2 & 0 \end{bmatrix}$$

(2). 若此时 P1 发出资源请求 Request₁ (1, 0, 1), 按银行家算法进行检查:

$$\text{Request}_1 (1, 0, 1) \leq \text{Need}_1 (2, 2, 2)$$

$$\text{Request}_1 (1, 0, 1) \leq \text{Available} (2, 1, 2)$$

试分配并修改相应的数据结构, 资源分配情况如表 3.23。

表 3.23 P1 申请资源后的资源变化情况

进程	Allocation			Need			Available		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	2	0	1	1	2	1	1	1	1
P2	4	1	1	2	0	2			
P3	2	1	1	1	0	3			
P4	0	0	2	4	2	0			

利用安全性检查算法检查, 可知可用资源向量(1, 1, 1)已不能满足任何进程的需求, 若分配给 P1, 系统将进入不安全状态, 因此此时不能将资源分配给 P1。

若此时 P2 发出资源请求 Request₂ (1, 0, 1), 按银行家算法进行检查:

$$\text{Request}_2 (1, 0, 1) \leq \text{Need}_2 (2, 0, 2)$$

$$\text{Request}_2 (1, 0, 1) \leq \text{Available} (2, 1, 2)$$

试分配并修改相应的数据结构, 资源分配情况如表 3.24 所示。

表 3.24 P2 申请资源所引起的资源变化

进程	Allocation			Need			Available		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	1	0	0	2	2	2	1	1	1
P2	5	1	2	1	0	1			
P3	2	1	1	1	0	3			

P4	0	0	2	4	2	0			
----	---	---	---	---	---	---	--	--	--

利用安全性检查算法，可得此时刻的安全性分析情况如表 3.25 所示。

表 3.25 P2 申请资源的安全性审查

	Work			Need			Allocation			Work+Allocation			Finish
	A	B	C	A	B	C	A	B	C	A	B	C	
P2	1	1	1	1	0	1	5	1	2	6	2	3	true
P3	6	2	3	1	0	3	2	1	1	8	3	4	true
P4	8	3	4	4	2	0	0	0	2	8	3	6	true
P1	8	3	6	2	2	2	1	0	0	9	3	6	true

从上面分析可知，存在一个安全序列{P2, P3, P4, P1}，故该状态时安全的，可以立即将 P2 所申请的资源分配给它。

(3) 如果(2)中两个请求立即得到满足后，系统此时并没有立即进入死锁状态，因为此时所有进程没有提出新的资源请求，全部进程都没有因资源请求没有得到满足而进入阻塞状态。只有当进程提出新的资源请求且全部进程(指 P1-P4)都进入阻塞状态时，系统才处于死锁状态。

本章到此就结束了，你觉得本章有什么地方不明白吗？任何问题，欢迎您与我们作者进行交流！



shareOurDreams
梦享团队微信号



weCSdream
梦享团队官方微信公众号



梦享论坛团队
梦享团队新浪微博

第四章 存储器管理

考点 1 内存管理的基本概念

温馨提示：本考点主要考查内存管理的基本概念。

一.选择题部分

1. (原书 第 4 题)静态重定位是在作业的()中进行的,动态重定位是在作业的()中进行的。
- A. 编译过程 B. 装入过程 C. 修改过程 D. 执行过程

【解析】本题考查动态重定位和静态重定位在作业运作过程中发生的时间。

静态重定位由操作系统中的重定位装入程序来完成。用户作业的相对于“0”编址的目标程序,是重定位装入程序的输入。重定位装入程序按照分配区域的起始地址逐一调整目标程序指令中的地址部分。目标程序经过重定位后,不仅进到分配给自己的绝对地址空间中,而且程序指令里的地址部分全部进行了修正,反映出正确的存储位置,从而保证程序的正确运行。

把相对地址空间中的用户作业程序“原封不动”地装入到分配给它的绝对地址空间中去,执行某条指令时,才根据当前程序所在区域,对指令中的地址进行重定位,即指令中地址的转换是在程序执行时动态完成的,故称为地址的“动态重定位”。

静态重定位是在程序运行之前完成地址转换的,动态重定位却是将地址转换的时刻推迟到指令执行时进行。

【补充】静态重定位是在装入时一次集中地把程序指令中所有要转换的地址全部加以转换;而动态重定位则是每执行一条指令时,对其地址加以转换。实行静态重定位,原来的指令地址部分被修改了;实行动态重定位,只是按照所形成的地址去执行这条指令,并不对指令本身做任何修改。

参考答案: B D

2. (原书 第 6 题)地址重定位的结果是得到()。
- A. 源程序 B. 编译程序 C. 目标程序 D. 执行程序

【解析】用户程序处理过程如图 4.1 所示。

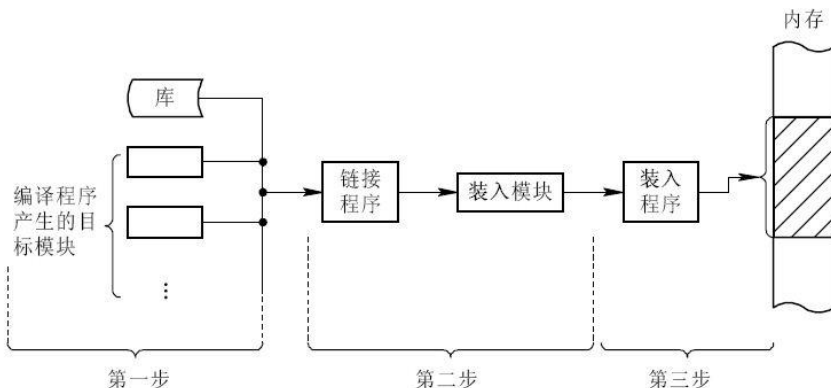


图 4.1 用户程序处理过程

将一个用户源程序变为一个可在内存中执行的程序，通常都要经过以下几个步骤：

- (1). **编译**。由编译程序(Compiler)将用户源代码编译成若干个目标模块(Object Module)；
 - (2). **链接**。由链接程序(Linker)将编译后形成的一组目标模块，以及它们所需要的库函数链接在一起，形成一个完整的装入模块(Load Module)；
 - (3). **装入**。由装入程序(Loader)将装入模块装入内存。
- 地址重定位的结果，是得到执行程序。

参考答案： D

3. (原书 第 8 题)存储管理的目的是()。
- A. 方便用户 B. 提高内存利用率
- C. 方便用户和提高内存利用率 D. 增加内存实际容量

【解析】本题考查存储管理的目的。存储管理的目的为以下 5 点：

- (1). **主存的分配和管理**。当用户需要内存时，系统为之分配相应的存储空间，不需要时，及时回收内存以供其它用户使用。
- (2). **提高主存储器的利用率**。不仅要能使多道程序动态地共享主存，提高主存利用率，最好还能共享主存中某个区域的信息。
- (3). **扩充主存容量**。为用户提供比主存物理空间大得多的地址空间，使用户感觉他的作业是在这样一个大的存储器中运行。

(4). **存储保护**。确保多道程序都在各自分配到的存储区域内操作，互不干扰，防止一道程序破坏其它作业或系统文件的信息。

(5). **方便用户**。

本题选择 C 更加恰当。

参考答案：C

4. (**原书 第 12 题**)()是指目标模块装入内存时一次分配完作业所需的内存空间，不允许在运行过程中再分配内存。

A. 静态分配

B. 动态分配

C. 直接分配

D. 碎片拼接后再分配

【解析】本题考查静态分配的基本概念。动态分配内存与静态分配内存的区别：

(1). 静态内存分配是在编译时完成的，不需要占用 CPU 资源；动态分配内存是在运行时完成的，动态内存的分配与释放需要占用 CPU 资源；

(2). 静态内存分配是在栈上分配的，动态内存是堆上分配的；

(3). 动态内存分配需要指针或引用数据类型的支持，而静态内存分配不需要；

(4). 静态分配内存需要在编译前确定内存块的大小，而动态分配内存不需要编译前确定内存大小，根据运行时环境确定需要的内存块大小，按照需要分配内存即可。可以说，静态内存分配是按计划分配，而动态内存分配是按需分配。

(5). 静态分配内存是把内存的控制权交给了编译器，而动态内存是把内存的控制权交给了程序员。

静态分配，指的是目标模块装入内存时一次分配完工作所需要的内存空间，不允许在执行的过程中再分配内存。静态分配内存适合于编译时就已经可以确定需要占用内存多少的情况，而在编译时不能确定内存需求量时可使用动态分配内存，但静态分配内存的运行效率要比动态分配内存的效率要高。

参考答案：A

5. (**原书 第 16 题**)设有 3 个起始地址都是 0 的目标模块 A、B、C，长度依次为 L、M、N，这 3 个模块按 A、B、C 顺序采用静态连接方式连接在一起后，模块 C 的起始地址变为 ()。

A. $L+M+N$

B. $L+M$

C. $L+M-1$

D. $L+M+1$

【解析】目标模块的起始地址是 0，而且依次按照 A、B、C 的顺序采用静态链接方法链接在一块。模块 A 的起始地址是 0，最高地址为 L-1。模块 B 接在模块 A 之后，起始地址为模块 A 的最高地址 L-1+1，即 L。

同理可知，模块 C 的起始地址为 L+M。故而选择 B 答案。

参考答案：B

二. 综合应用题部分

1. (原书 第 1 题) 什么叫重定位？静态重定位和动态重定位的有什么区别？

【解析】由于程序中的逻辑地址和实际分配后主存储器的绝对地址不一致，为使处理器能按实际地址正确地访问主存，必须进行地址转换。逻辑地址转换成绝对（物理）地址的过程称为重定位，也称为地址转换。

重定位的方式有静态定位和动态定位两种。

(1). **静态重定位**。所谓静态重定位是在装入一个作业的时候，把作业中的指令地址和数据地址全部一次性地转换成绝对地址，由于地址转换是在作业执行一次完成的，装入主存中的作业信息已经是用绝对地址指示的，所以作业在执行过程中，无需再进行地址转换，且不能移动位置。

(2). **动态重定位**。所谓动态重定位是指在装入作业时，不进行地址转换，而是直接把作业装入到分配的主存区域中。在作业执行过程中，每当执行一条指令时由硬件的地址转换机构将指令中的逻辑地址转换成绝对（物理）地址。动态重定位是由软件和硬件相配合来实现的，硬件有一个地址转换机构，该机构有一个基址寄存器和一个地址转换线路组成。基址寄存器中存放作业所占主存的起始地址，所以地址转换公式为：绝对地址=基址寄存器的值+逻辑地址。

考点 2 交换与覆盖

温馨提示：本考点主要考查覆盖与交换，请同学们注意区别这两种方式。本考点稍微了解即可。

一. 选择题部分

1. (原书 第 2 题)在存储管理中,采用覆盖与交换技术的目的是()。

- A. 节省主存空间
- B. 物理上扩充内存容量
- C. 提高 CPU 效率
- D. 实现主存共享

【解析】本题考查在存储管理中引入覆盖和交换技术的目的。

交换与覆盖技术是在多道程序环境下用来扩充内存的两种方法,该两种扩充内存的方法是逻辑上的,并没有真正在物理上扩充内存。记住了,物理上扩充内存,比如添加内存条等,内存真正得到了扩充,这是不同于逻辑扩充的地方。

参考答案: A

2. (原书 第 4 题)在分时系统中,可将进程不需要或暂时不需要的部分移到外存,让出内存空间以调入其他所需数据,称为()。

- A. 覆盖技术
- B. 对换技术
- C. 虚拟技术
- D. 物理扩充

【解析】本题考查交换技术(对换技术)。交换技术将系统中某些进程暂时移动到外存,把外存中某些进程换进内存,占据前者所占用的空间。

一个程序通常由若干功能上独立的程序段组成,在运行时,并不是所有的程序段都同时调入内存执行。因而可以按照程序自身的逻辑结构,让不同时执行的程序段先后共享同一块内存区域,这就是覆盖技术。

覆盖技术先将程序必须的部分代码和数据调入内存,其余部分先放在外存上,当要访问的程序和数据不在内存时,由操作系统负责将其从外存中调入,这就解决了在较小的内存空间中运行较大程序的问题。

参考答案: B

二. 综合应用题部分

1. (原书 第 1 题)简述什么是覆盖?什么是交换?覆盖和交换的区别是什么?

【解析】所谓**覆盖**,是指同一主存区可以被不同的程序段重复使用。通常一个作业由若干个功能上相互独立的程序段组成,作业在一次运行时,也只用到的其中的几段,利用这样一个事实,我们就可以让那些不会同时执行的程序段共用同一个主存区。

所谓**交换**,就是系统根据需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。

覆盖技术要求程序员必须把一个程序划分成不同的程序段，并规定好它们的执行和覆盖顺序，操作系统根据程序员提供的覆盖结构来完成程序之间的覆盖。覆盖主要在同一个作业或同一个进程内进行，而交换主要是在进程或作业之间进行。另外，覆盖只能覆盖那些与覆盖程序段无关的程序段。

考点 3 连续分配管理方式

温馨提示：连续分配方式主要包括单一连续分配、固定分区分配和动态分区分配。单一连续分配和固定分区分配比较简单，请同学们稍作了解。动态分区分配有几种方法，即首次适应算法、最佳适应算法、最坏适应算法和邻近适应算法。有些学校比较喜欢考查这个，请同学们多注意。

一. 选择题部分

1. (原书 第 2 题) 首次适应算法的空闲区是 ()。

- A. 按地址递增顺序连在一起
- B. 始端指针表指向最大空闲区
- C. 按大小递增顺序连在一起
- D. 寻找从最大空闲区开始

【解析】 本题考查首次适应算法的空闲区组织方式。

首次适应算法将空闲分区以地址递增的次序链接。该算法给某一个作业分配内存时，从链首开始顺序查找，直至找到一个大小能满足要求的空闲分区为止，再按作业的大小，从该空闲分区中划出一块内存空间分配给请求者，余下的空闲分区仍留在空闲链中。若从链首直至链尾都不能找到一个能满足要求的分区，则失败返回。

首次适应算法的缺点是有两点：

- (1). 空闲区分布不均匀，小地址的空闲区优先占满，高地址不容易查到；
- (2). 内存利用率不高。

参考答案： A

2. (原书 第 3 题) 在回收内存时可能出现下述四种情况：

- (1). 释放区与插入点前一分区 F1 相邻接，此时应 ()；

- (2). 释放区与插入点的后一分区 F2 相邻接, 此时应 ();
- (3). 释放区不与 F1 和 F2 相邻接, 此时应 ()。
- A. 为回收区建立一分区表项, 填上分区的大下和始址。
- B. 以 F1 分区的表项作为新表项且不做任何改变。
- C. 以 F1 分区的表项作为新表项, 修改新表项的大小。
- D. 以 F2 分区的表项作为新表项, 同时修改新表项的大小和始址。
- E. 以 F2 分区的表项作为新表项, 同时修改新表项的始址。

【解析】本题考查回收内存时, 空闲区的合并方法。动态分区内存回收情况问题, 也是常遇到的一个问题。该问题有以下四种情况, 我们给一个总结。

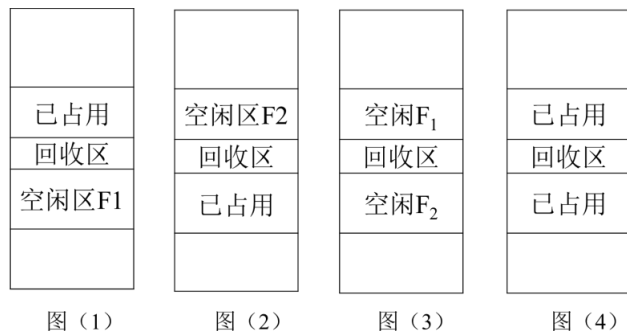


图 4.2

如图 4.2 示, 图 (1) 与图 (2) 分别是回收区的上下邻接分区空闲的情况, 这种情况不改变空闲分区数目。图 (3) 则减少一个空闲分区, 合并了一个更大的空闲分区。图 (4) 则多出来一个空闲分区。

(1). **释放区 (回收区) 与插入点前一个分区 F1 相邻接**, 对应于图 (1), 回收区与空闲分区 F1 合并成更大的分区, 但是首地址仍是 F1 的首地址。故而, 只需要修改表项的大小即可。第 1 小题选择 C。

(2). **释放区与插入点的后一个分区 F2 相邻接**, 对应于图 (2)。此时, 回收区与空闲区 F2 合并成更大的分区, 空闲区首地址该成回收区首地址, 新表项的大小改成空闲区 F2 与回收区的大小之和。第 2 小题选择 D。

(3). **释放区位于空闲区 F1 与空闲区 F2 之间**, 如图 (3) 所示, 需要删除空闲区 F1 表项。并把 F2 表项的大小改成空闲区 F1、回收区、空闲区 F2 三者的大小之和。

(4). **释放区不与 F1 和 F2 相邻接**, 对应于图 (4)。此时, 需要增加一个空闲区表项, 表项的首地址为回收区的首地址, 表项的大小为回收区的大小。第 3 小题选择 A。

参考答案：C D A

3. (原书 第6题) 在存储器管理中,“碎片”是指()。

- A. 存储分配完后所剩空闲区
- B. 没有被使用的存储区
- C. 不能被使用的存储区
- D. 未被使用,而又暂时不能使用的存储区

【解析】本题考查“碎片”的基本概念。碎片是指未被利用,又难以利用的存储区。碎片分为内部碎片和外部碎片两种。

(1). 内部碎片

内部碎片就是已经被分配出去(能明确指出属于哪个进程)却不能被利用的内存空间。

内部碎片是处于区域内部或页面内部的存储块,占有这些区域或页面的进程并不使用这些存储块。进程占有这些存储块时,系统无法利用它们(无法将这些存储块分配给其他进程),直到进程释放它们或进程结束时,系统才有可能利用这些存储块。

(2). 外部碎片

外部碎片指的是还没有被分配出去(不属于任何进程),但由于太小了无法分配给申请内存空间的新进程的内存空闲区域。

外部碎片是出于任何已分配区域或页面外部的空闲存储块。这些存储块的总和可以满足当前申请的长度要求,但是由于它们的地址不连续或其他原因,使得系统无法满足当前申请。

【补充】单道连续分配只有内部碎片,多道固定连续分配既有内部碎片,又有外部碎片。

参考答案：C

4. (原书 第11题) 设内存的分配情况如图4.3所示。若要申请一块40K字节的内存空间,若采用最佳适应算法,则所得到的分区首址为()。

- A. 100K
- B. 190K
- C. 330K
- D. 410K



图 4.3

【解析】我们定义一种空闲块的表述方式：（空闲块的首地址，空闲块的大小）。由图可知，空闲区的大小分别为(100K, 80KB)、(190K, 90KB)、(330K, 60KB)和(410K, 112KB)。显然，最佳适应算法把所有空闲块按照空闲区大小形成队列，并从空闲分区链表中找到第一块能够满足需求的大小的空闲块分配给该作业。因为申请 40KB，故而，分配空闲块（330K, 60KB）最合适。

故而，选择 C 答案。

【注意】内存起始位置不能表述成（100KB, 80KB），这二者是不同的概念。

参考答案：C

5. （原书第 20 题）操作系统为实现多道程序并发，对内存管理可以采用多种方式，其中代价最小的是（ ）。

- | | |
|---------|----------|
| A. 分区管理 | B. 分页管理 |
| C. 分段管理 | D. 段页式管理 |

【解析】本题考查存储管理方式。常见的几种存储管理方式如下：

- (1). 界地址管理方式（一维地址）：分区管理。
- (2). 页式管理方式（一维地址）：分页管理。
- (3). 段式管理方式（二维地址）：分段管理。
- (4). 段页式管理方式（二维地址）：段页式管理。

页式管理、段式管理和段页式管理需要额外的内存空间保存段表和页表。界地址管理方式没有段表和页表，所以相比较代价较小。此外，页式管理、段式管理和段页式管理的地址变换过程比分区管理复杂。

参考答案：A

二. 综合应用题部分

1. (原书 第 1 题) 某操作系统采用可变分区分配存储管理方法，用户区为 512K 且地址为 0，用空闲分区表管理空闲分区。若分配是采用分配空闲区低地址部分的方案，且初始时用户区的 512K 空间空闲，对下述申请序列：申请 300K，申请 100K，释放 300K，申请 150K，申请 30K，申请 40K，申请 60K，释放 30K。回答下列问题：

- (1). 采用首次适应算法，空闲分区中有哪些空块（给出始址、大小）？
- (2). 采用最佳适应算法，空闲分区中有哪些空块（给出始址、大小）？
- (3). 如再申请 100K，针对（1）和（2）各有什么结果？

【解析】

(1). 采用首次适应算法，在完成了题目所给的系列申请及释放内存操作后，内存分配情况如图所示（用阴影表示空闲空间），空闲分区表如图 4.4 和图 4.5 所示。

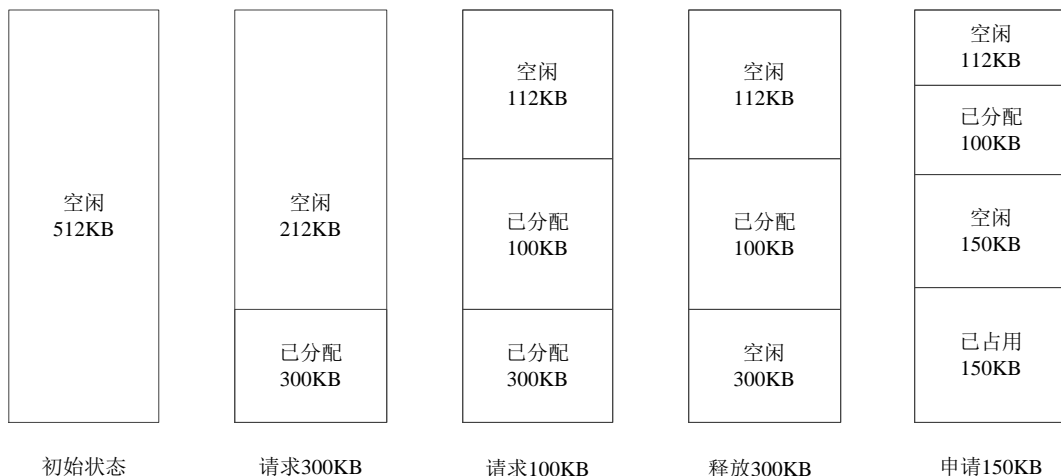


图 4.4 采用首次适应算法的内存分配情况（1）

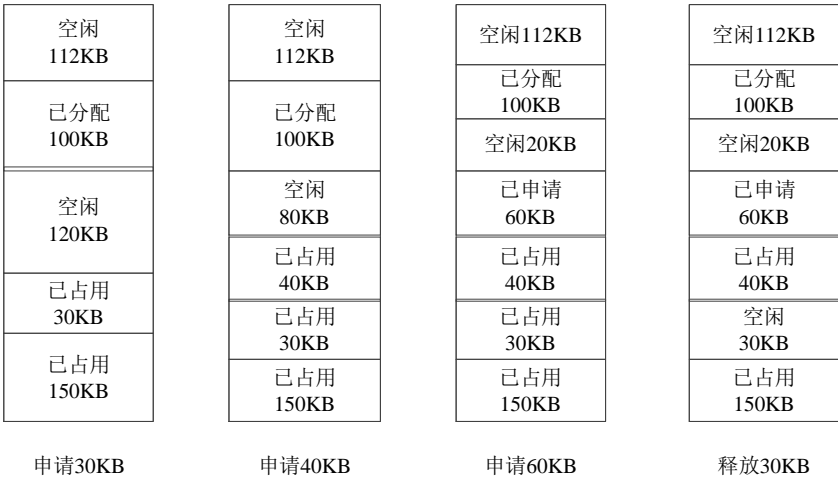


图 4.5 采用首次适应算法的内存分配情况（2）

由图 4.5 可知，空闲分区为（150K,30KB）、（280K,20KB）和（400K,112KB）。

(2). 采用最佳适应算法，完成了题目所给的系列申请及释放内存操作后，内存分配情况如图所示（用阴影表示空闲空间），空闲分区如图 4.6 与图 4.7 所示。

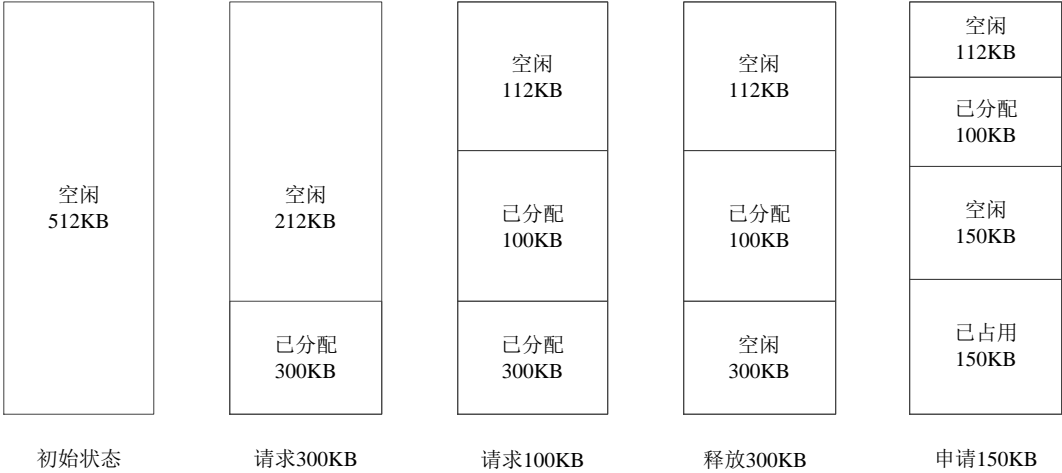


图 4.6 采用最佳适应算法的内存分配情况（1）



图 4.7 采用最佳适应算法的内存分配情况 (2)

由图 4.7 可知，空闲分区为 (210K,90KB)、(400K,30KB) 和 (470K,42KB)。

(3). 再申请 100K 空间，由上述结果可知，采用首次适应算法后剩下的空闲分区能满足这一申请要求；而采用最佳适应算法后剩下的空闲分区不能满足这一申请要求。

2. (原书 第3题) 在系统中采用可变分区存储管理，操作系统占用低地址部分的 126KB，用户区的大小是 386KB，若采用空闲分区表管理空闲分区。若分配时从高地址开始，对于下述的作业申请序列：作业 1 申请 80KB；作业 2 申请 56KB；作业 3 申请 120KB；作业 1 完成；作业 3 完成；作业 4 申请 156KB；作业 5 申请 80KB。试用首次适应法处理上述作业，并回答下面问题。

- (1). 画出作业 1、2、3 进入内存后。内存分布情况。
- (2). 画出作业 1、3 完成后。内存的分布情况。
- (3). 画出作业 4、5 进入内存后。内存分布的情况。

【解析】注意本题采用的是首次适应算法来给作业分配内存，故而有：

(1). 操作系统占用低地址部分的 126KB，用户区的大小是 386KB，分配时从高地址开始，作业 1、作业 2、作业 3 依次进入内存后内存的变化情况如图 4.8 所示。

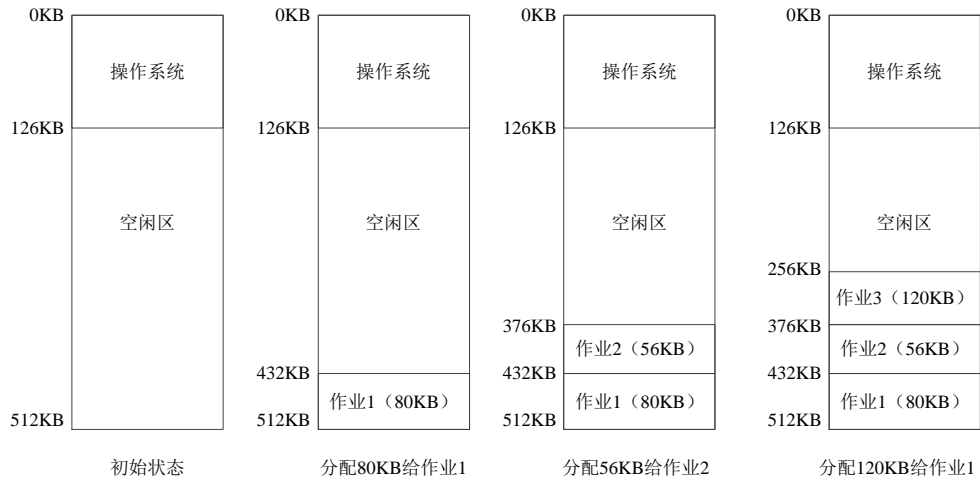


图 4.8 作业 1、2、3 依次进入内存后内存的变化情况

(2). 作业 1、3 完成之后，内存的变化如图 4.9 所示。

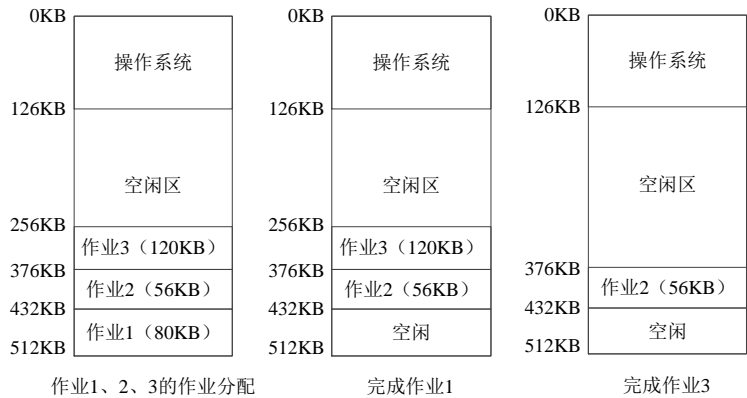


图 4.9 完成作业 1、3 后内存的变化情况

(3). 作业 4、5 进入内存后，内存分配情况如下图 4.10 所示。

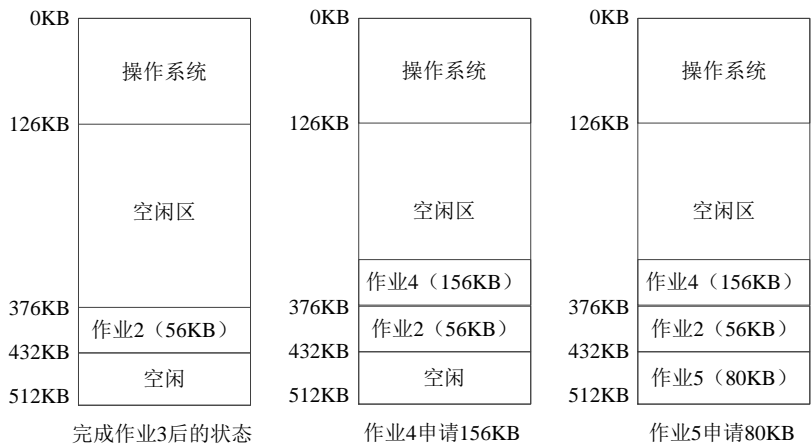


图 4.10 作业 4、5 进入内存后内存的变化情况

考点 4 非连续分配管理方式

温馨提示：本考点主要考查内存非连续分配管理方式，包括基本分页存储管理方式、基本分段存储管理方式和基本段页式存储管理方式。请同学们注意这三种分配方式下的地址变换机构、地址映射方法。

一. 选择题部分

1. (原书 第 1 题)在虚拟存储系统中，若进程在内存中占 3 块（开始时为空），采用先进先出页面淘汰算法，但执行访问页号序列为 1、2、3、4、1、2、5、1、2、3、4、5、6 时，将产生（ ）次缺页中断。
- A. 7 B. 8 C. 9 D. 10

【解析】这里，我们先对虚拟存储系统的页面访问热热身。呵呵...

表 4.1 FIFO 算法页面替换情况

访问序列	1	2	3	4	1	2	5	1	2	3	4	5	6
内存块 1	1	1	1	4	4	4	5	5	5	5	5	5	6
内存块 2		2	2	2	1	1	1	1	1	3	3	3	3
内存块 3			3	3	3	2	2	2	2	2	4	4	4
缺页否?	√	√	√	√	√	√	√			√	√		√

由表 4.1 可知，一共产生了 10 次缺页中断，故而选择 D 答案，热身完毕。

参考答案：D

2. (原书 第 2 题) 分页式存储管理中，地址转换工作是由 () 完成的。

- A. 硬件
- B. 地址转换程序
- C. 用户程序
- D. 装入程序

【解析】在分页存储系统中，进行存储分配时，应为进入主存的每个用户作业建立一张页表，指出逻辑地址中页号与主存中块号的对应关系，页表的长度随作业的大小而定。同时页式存储管理系统包括一张作业表，将这些作业的页表进行登记，每个作业在作业表中有一个登记项。

然后，借助于硬件的地址转换结构，在作业执行过程中按页动态定位。调度程序在选择作业后，从作业表中的登记项中得到被选中作业的页表始址和长度，将其送入硬件设置的页表控制寄存器。地址转换时，只要从页表控制寄存器就可以找到相应的页表，再按照逻辑地址中的页号查页表，得到对应的块号，根据关系式：

$$\text{绝对地址} = \text{块号} \times \text{块长} + \text{单元号}$$

计算出要访问的内存地址单元。

参考答案：A

3. (原书 第 7 题) 段式和页式存储管理的地址结构很类似，但是它们之间有实质上的不同。以下说法中，错误的是 ()。

- A. 页式的逻辑地址是连续的，段式的逻辑地址可以不连续
- B. 页式的地址是一维的，段式的地址是二维的
- C. 分页是操作系统进行的，分段是用户确定的
- D. 页式采用动态重定位方式，段式采用静态重定位方式

【解析】本题考查段式存储管理和页式存储管理的地址结构差异。页式的地址是一维的，段式的地址是二维的。分页由操作系统决定，而分段是由用户决定的。页式和段式都可以采用动态重定位方式。

分页存储器将内存空间分成大小相等的块，与此同时，程序的逻辑地址也分成与块大小相同的页。采用分页式存储管理时，逻辑地址是连续的。用户在编制程序时只须使用顺序地址，而不必考虑如何去分页。

在给作业分配内存时，以块为单位进行分配，一个作业有多少页，装入内存时就给它分配多少块。必须注意的是，分配给作业的主存块是可以不连续的，即作业的信息可按页分散存放在主存的空闲中，这就避免了为得到连续存储空间而进行的移动。

【总结】对此，我们给出一个总结，如表 4.2 所示，请同学们理解该表。

表 4.2

功能 方法	单一连续 分配	分区存储管理方式		页式存储管理方式		段式存储 管理方式	段页式存 储管理
		固定分区	动态分区	静态	动态		
适用环境	单道	多道		多道		多道	多道
虚拟空间	一维	一维		一维		二维	
重定位法	静态	静态、动态		动态		动态	动态
分配方式	静态连续 分区分配	静态、动态连续分区 分配		静态或者动态以页为 单位非连续分配方式		以段为单 位动态非 连续分配	以页为单 位动态非 连续分配
释放	执行完后 全部释放	执行完后 全部释放	分区释放	执行完后 释放	淘汰、执行 完后释放	淘汰、执行 完后释放	淘汰、执行 完后释放
保护	越界保护	越界保护		越界保护、控制权保护		同左	同左
内存扩充	覆盖、交换	覆盖、交换		覆盖、交 换	虚拟内 存	虚拟内 存	虚拟内 存
共享	不能	不能		较难		方便	方便
硬件支持	界地址保 护寄存器	界地址保护寄存器、重定 位机构		地址变换机构、中断机 构、界地址保护机构		段式地址变换机构、界地 址保护寄存器、动态连接 机构	

参考答案： D

4. (原书 第 11 题)【2010 年统考】某计算机采用二级页表的分页存储管理方式，按字节

编制，页大小为 2^{10} 字节，页表项大小为 2 字节，逻辑地址结构为：

页目录号	页号	页内偏移量
------	----	-------

逻辑地址空间大小为 2^{16} 页，则表示整个逻辑地址空间的页目录表中包含表项的个数至少是（ ）。

- A. 64 B. 128 C. 256 D. 512

【解析】依题意，页大小是 2^{10} 字节，故而页内偏移量用 10 位地址表示。每个页表项的大小是 2 字节，每页可以有 2^9 个页表项。而逻辑地址空间的大小是 2^{16} 页，故而需要页目录表 2^7 个，用来记录每个页表的起始地址。故而选择 B 答案。

参考答案：B

5. (原书 第 13 题) 在在页式存储管理中，每个页表的表项实际上是用于实现（ ）。

- A. 访问内存单元 B. 静态重定位
C. 动态重定位 D. 装载程序

【解析】本题考查页表的相关内容。重定位是指被换出的进程再次运行之前必须重新装入内存，而再次进入内存时的存放位置与换出之前通常不同。这就要求程序编址与内存存放位置无关，这种程序称为可重定位程序。动态重定位在进程运行时进行，静态重定位在进程运行前编译时或装入时进行。

页表，每个进程一个，用于记录进程的逻辑页面与内存页框之间的对应关系，根据页号可以找到页框号。如图 4.11 所示。

逻辑页号:	页框号
0	7
1	23
2	12
3	37

图 4.11 页表

页框号是物理地址的高位部分,根据页框号与页内地址可以确定内存物理地址。

参考答案：A

二. 综合应用题部分

1. (原书 第 1 题) 有一页式系统，其页表存放在主存中。

本资料供给考生免费使用，任何机构不得用于盈利！ | 第四章 存储器管理

87

本资料是从《2016 年考研核心考点命题思路解密—计算机操作系统》中抽取的少部分内容，重新排版而成。若有需要，请参考原书！

- (1). 如果对主存的一次存取需要 1.5 微秒, 试问实现一次页面访问的存取时间是多少?
- (2). 如果系统加有快表, 平均命中率为 85%, 当页表现在快表中时, 其查找时间忽略为 0, 试问此时的存取时间为多少?

【解析】若页表存放在主存中, 则要实现一次页面访问需两次访问主存, 一次是访问页表, 确定所存取页面的物理地址, 第二次才根据该地址存取页面数据。

(1). 由于页表存放在主存, 因此 CPU 必须两次访问主存才能获得所需数据, 所以实现一次页面访问的存取时间是 $1.5 \times 2 = 3$ 微秒。

(2). 在系统增加了快表后, 在快表中找到页表项的概率为 85%, 所以实现一次页面访问的存取时间为 $0.85 \times 1.5 + (1 - 0.85) \times 2 \times 1.5 = 1.725$ 微秒。

2. (原书 第 4 题) 在某分页系统中, 测得 CPU 和磁盘的利用率如下, 试指出每种情况下的问题和措施。

- (1). CPU 的利用率为 15%, 磁盘利用率为 95%。
- (2). CPU 的利用率为 88%, 磁盘利用率为 3%。
- (3). CPU 的利用率为 13%, 磁盘利用率为 5%。

【解析】在某分页虚存系统中, 在题中的 CPU 和磁盘的利用率的情况下, 出现的问题和应采取的措施如下:

- (1). 可能已出现了抖动现象, 应减少系统的进程数。
- (2). 系统比较正常, 可考虑适当增加进程数以提高资源利用率。
- (3). CPU 和磁盘的利用率都较低, 必须增加并发进程数。

考点 5 虚拟内存的基本概念

温馨提示: 本考点主要考查虚拟内存的基本概念, 请同学们了解传统存储方式和虚拟存储方式的特征、掌握局部性原理、虚拟内存技术的实现。

一. 选择题部分

1. (原书 第 1 题) 虚拟存储管理策略可以 ()。

- A. 扩大物理内存容量
- B. 扩大物理外存容量
- C. 扩大逻辑内存容量
- D. 扩大逻辑外存容量

【解析】作业全部装入内存方可运行，会出现以下两种情况：

(1). 作业很大，所要求的内存空间超过了内存总容量，作业不能全部装入内存，致使作业无法运行。

(2). 有大量作业要求运行，但由于内存不足难以容纳所有作业，只能将少量作业装入内存让它们先执行，而将其他大量作业留在外存上等待。

这两种情况都会影响内存利用率，因而引入虚拟存储的概念。当进程运行时，先将一部分程序装入内存，另一部分暂留在外存。当要执行的指令不在内存时，由系统自动完成将它们从外存调入内存工作。当没有足够的内存空间时，系统自动选择部分内存空间，将其中原有内容交换到磁盘上，并释放这些空间供其他其他进程使用。

参考答案：C

2. (原书 第4题) 虚拟存储管理系统的基础是程序的 () 理论。

- A. 局部性
- B. 全局性
- C. 动态性
- D. 虚拟性

【解析】虚拟存储器是基于程序的局部性原理实现的。在一段时间内一个程序的执行往往呈现出高度的局部性，具体而言，又分为时间局部性和空间局部性。时间局部性是指：一条指令被执行了，则在不久的将来它可能被再次执行。时间局部性主要有以下几种现象：

- (1). 循环；
- (2). 子程序；
- (3). 栈；
- (4). 用于计数和总计的变量。

空间局部性则是指：若某一个存储单元被使用，则在一定的时间内，与该存储单元相邻的存储单元可能被使用。空间局部性主要有以下几种现象：

- (1). 数组遍历；
- (2). 代码的顺序执行；
- (3). 程序员倾向于将相关的变量定义相互靠近存放。

【补充】程序局部性主要体现在：

(1). 程序在执行时，在大多数情况下仍是顺序执行的。这种特性使得程序的执行在一段时间内被限制在作业的某一局部范围。

(2). 过程调用将会使程序的执行轨迹由一部分内存区域转至另一部分区域。但在大多数情况下, 过程调用的深度都不超过 5。在一段时间内, 程序将会被局限于这些过程的范围内运行。

(3). 程序中存在许多循环结构, 它们可以多次重复执行。

(4). 程序中还包括许多对数据结构的处理, 它们往往都局限于很小的范围内。

参考答案: A

3. (原书 第 8 题) 以下存储管理技术中, 支持虚拟存储器的技术是 ()。

A. 动态分区法

B. 可重定位分区法

C. 请求分页技术

D. 覆盖技术

【解析】请求调页(段)都是比较常见的虚拟存储技术。动态分区技术要求作业一次性全部装入内存, 这显然不是虚拟存储。

【补充】采用页式、段式、段页式管理可以实现虚拟存储器, 但对固定分区、可变分区方式都不能实现虚拟存储器。

实现虚拟存储技术的物质基础是二级存储结构(主存与辅存)和动态的地址转换机构(动态重定位)。固定分区方式没有硬件地址转换机构。

可变分区方式管理主存也不能实现虚拟存储。因为在这种管理方式下, 每次必须将作业完整地调入主存, 并要求连续存放, 这不符合虚拟存储器的基本原理。另外, 虽然可变分区方式有硬件地址转换机构, 但它把绝对地址超出限定范围按出错处理, 而不是产生“缺分区中断”。

虚拟存储器的特征可以归结为以下 16 个字: **虚拟扩充**(只是逻辑上扩充了内存并非真正扩充了主存容量)、**部分装入**(每个作业不是全部一次性地装入内存, 而是分成若干部分。在需要调入内存时, 按部分装入内存)、**离散分配**(装入内存的作业部分不必占有连续的内存空间, 而是“见缝插针”)、**多次对换**(作业运行时, 程序和数据多次在主存和辅存之间对换)。

参考答案: C

4. (原书 第 14 题) 【2011 年统考真题】在虚拟内存管理中, 地址变换机构将逻辑地址变换为物理地址, 形成该逻辑地址的阶段是 ()。

A. 编辑

B. 编译

C. 连接

D. 装载

【解析】作业进入内存的过程如图 4.12 所示。

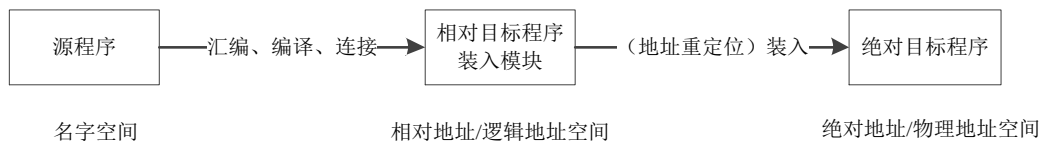


图 4.12 作业装入内存的过程

在源程序中，是通过符号名来访问子程序和数据的，程序中符号名的集合称为“名字空间”。汇编语言源程序经过汇编，或者高级语言程序经过编译，得到目标程序是以“0”作为参考地址的模块。

之后多个目标层序由连接程序连接成一个具有统一地址的装配模块，以便最后装入内存中运行。目标模块中的地址称为相对地址或者逻辑地址。

装配模块虽然具有统一的地址空间，但是仍然以“0”作为参考地址。要把它装入内存执行，还需要确定装入内存的物理地址，并修改程序中与地址相关的代码。这一个过程叫做地址重定位。

地址空间的程序和数据经过地址重定位处理，就变成了 CPU 可以直接执行的绝对地址程序，这些地址空间称为“绝对地址空间”或者物理空间。

参考答案：C

二. 综合应用题部分

1. (原书 第 2 题) 虚拟存储技术的理论基础(局部性原理)是什么？

【解析】程序局部性原理：虚拟存储管理的效率与程序局部性程序有很大关系。根据统计，进程运行时，在一段时间内，其程序的执行往往呈现出高度的局限性，包括时间局部性和空间局部性。

(1). 时间局部性：是指若一条指令被执行，则在不久，它可能再被执行。

(2). 空间局部性：是指一旦一个存储单元被访问，那它附近的单元也将很快被访问。

2. (原书 第 4 题) 有一程序要把 100×100 数组置初值为 0，假定有两个内存块可用来存放数组数据，每块内存可存放 100 个数组元素，数组中的元素按行编址。两内存块初始状态为空，下列程序：

```
[1]  Var A:array[1..100] of array[1..100] of integer;
      for j:=1 to 100 do
```

```

        for i:=1 to 100 do
            A[i,j]:=0;
[2]   Var A:array[1..100] of array[1..100] of integer;
        for i:=1 to 100 do
            for j:=1 to 100 do
                A[i,j]:=0;

```

当采用 LRU 页面调度算法时各会产生多少次缺页中断？

【解析】每页存放 100 个数组元素，按行编址，存放顺序为 $A[1,1] \rightarrow A[1,2] \rightarrow A[1,3] \rightarrow \dots \rightarrow A[1,100] \rightarrow \dots \rightarrow A[100,100]$ 。因 $A[1,1] \rightarrow A[1,100]$ 存放在第 1 页， $A[100,1] \rightarrow A[100,100]$ 存放在第 100 页。

- (1). 产生 10000 次缺页中断；
- (2). 产生 100 次缺页中断。

考点 6 请求分页管理方式

温馨提示：本考点考查请求分页管理方式，请同学们掌握页表机制、缺页中断机制和地址变换机构。

一. 选择题部分

1. (原书 第 1 题) 作业在执行中发生了缺页中断，经操作系统处理后，应让其执行()指令。
 - A. 被中断的前一条
 - B. 被中断的
 - C. 被中断的后一条
 - D. 启动时的第一条

【解析】本题考查缺页中断下指令的执行。缺页中断不同于我们常说的中断，缺页中断是要访问的页面不在内存中，转而去外存调页。当作业执行过程中，发生了缺页中断，操作系统到外存调页，把页面调入内存并作相应的处理之后，应该继续执行被中断的那一条指令。

参考答案： B

2. (原书 第 3 题)假定某采用分页式存储管理的系统中,主存的容量为 1M,被分成 256 块,块号为 0, 1, 2, …… , 255。某作业的地址空间占用 4 页,其页号为 0, 1, 2, 3, 被分配到主存中的第 2, 4, 1, 5 块中。主存地址应该用 (①) 位来表示,作业中的每一页长度为 (②), 逻辑地址中的页内地址应占用 (③) 位来表示,逻辑地址空间至小是 (④) 位,作业中第 2 页在分到的主存块中的起始地址是 (⑤)。

①, ③, ④:

- A. 8 B. 10 C. 12 D. 20

②:

- A. 512 字 B. 1024 字节 C. 2048 字节 D. 4096 字节

⑤:

- A. 2 B. 1024 C. 4096 D. 4095

【解析】主存的大小是 1MB, 即 2^{20} B, 故而需要用 20 位来表示。这 1MB 的内存空间被划分成 256 块, 每块的大小是 4KB, 即 4096 字节。因为页的大小是 4KB, 而页号需要表述 256 个页面, 故而, 页号字段是 8 位, 页内地址是 12 位。

因为作业的地址空间占用 4 页, 所以逻辑空间至小是 2 位。作业中第 2 页对应的主存块号是 1, 而块号为 0 的物理块的地址范围是 0~4096-1, 故而块号 1 物理地址的起始地址是 4096。

参考答案: D D C A C

3. (原书 第 7 题)在一个分页虚存系统中, 设页长 2KB, 某用户程序有 30 页。若该程序的虚页 0、1、2、3、4、5、6、7 已分别装入内存块 4、17、18、20、25、26、30、32 中, 则该程序中的虚地址 0AC5H 和 3AC5H 对应的物理地址分别是 ()。

- A. 4AC5H 和 14AC5H B. 4AC5H 和 20AC5H
C. 8AC5H 和 20AC5H D. 8AC5H 和 102C5H

【解析】本题考查请求分页系统中, 将虚拟地址转换成物理地址的计算方法。

依题意, 因为页长是 2KB, 需要 11 位页内地址。而把 0AC5H 转换成二进制数得到 0000 1010 1100 0101 B, 下划线部分表示页内地址, 而高位部分表示页号。可知页号为 1, 对应物理块号 17。故而, 物理地址是 1000 1010 1100 0101B, 即 8AC5H。

对于地址 3AC5H 也做同样的解析，得到二进制地址 0011 1010 1100 0101B，得到虚拟页号为 7，对应的物理页号是 32。因而，得到的物理地址是 1 0000 0010 1100 0101B，即 102C5H。故而，选择 D 答案。

参考答案：D

4. (原书 第 9 题)【2011 年统考真题】在缺页处理过程中，操作系统执行的操作可能是 ()。

I. 修改页表

II. 磁盘 I/O

III. 分配页框

A. 仅 I、II

B. 仅 II

C. 仅 III

D. I、II 和 III

【解析】本题考查操作系统对缺页的处理办法。采用请求分页存储管理的系统，若进程要访问的页面不在内存，则缺页中断，把磁盘上的内容调入内存中。故而，会有磁盘 I/O。要把外存上内容调入内存，当然要分配页框给该内容。最后，缺失的页面被调入内存，还需要修改页表，比如将页表的存在位改成 1，表示该页在内存中等。

参考答案：D

二. 综合应用题部分

1. (原书 第 1 题)若在一个页式存储管理系统中，页表内容如表 4.3 所示。已知页面大小为 1024 字节，试将逻辑地址 1011, 2148, 3000, 4000, 5012 转化为相应的物理地址。

表 4.3 页表内容

页号	块号
0	2
1	3
2	1
3	6

【解析】本题中，为了描述方便，设页号为 P，页内位移为 W，逻辑地址为 A，页面大小为 L，则：

$$P = \text{int}(A/L), W = A \bmod L$$

(1). 对于逻辑地址 1011, $P = \text{int}(1011/1024) = 0$, $W = 1011 \bmod 1024 = 1011$ ，查页表第 0 页在第 2 块，所以物理地址为 3059。

(2). 对于逻辑地址 2148, $P = \text{int}(2148/1024) = 2$, $W = 2148 \bmod 1024 = 100$ ，查页表第 2 页

在第 1 块，所以物理地址为 1124。

(3). 对于逻辑地址 3000, $P=(3000/1024)=2$, $W=3000 \bmod 1024=952$, 查页表第 2 页在第 1 块, 所以物理地址为 1976。

(4). 对于逻辑地址 4000, $P=(4000/1024)=3$, $W=4000 \bmod 1024=928$, 查页表第 3 页在第 6 块, 所以物理地址为 7072。

(5). 对于逻辑地址 5012, $P=(5012/1024)=4$, $W=5012 \bmod 1024=916$, 因页号超过页表长度, 该逻辑地址非法。

2. **(原书 第 4 题)**某计算机系统主存采用请求分页管理技术, 主存容量为 1MB, 被划分为 256 块, 每块大小为 4KB。假设某个作业共有 5 个页面, 其中 0, 1, 2 三个页面已分别装入到主存 4, 9, 11 三个物理块中, 另外两个页面没有装入主存。该作业的页面变换表 (PMT) 如表 4.4 所示。表中的状态为 0 表示页面已经装入到内存中, 为 1 表示没有装入内存。

表 4.4 作业的页面变换表 PMT

页号	块号	状态
0	4	0
1	9	0
2	11	0
3	—	1
4	—	1

问题:

- (1). 若给定一个逻辑地址为 9016, 其物理地址是多少? 画出地址变换图。
- (2). 若给定一个逻辑地址为 12388, 其物理地址是多少? 地址变换过程中会出现什么问题? 出现问题应如何解决?

【解析】在请求分页的存储管理系统中, 系统是通过查页表来进行地址转换的。对于本题中采用的页面大小为 4KB, 即页内相对地址为 12 位。

(1). 首先从虚拟地址中分离出页号和页内地址。 $[9016/4096]=2$, 所以页号为 2, 页内地址为 824。查页表知道 2 号页对应的物理块号为 11, 即物理地址为: $11*4096=45056$, 再加上页内地址后其真正的物理地址为 45880。

(2). 首先从虚拟地址中分离出页号和页内地址。 $[12388/4096]=3$, 所以页号为 3, 页内地址为 100。查页表知道 3 号页对应的物理块号, 发现其物理块号不在内存, 此时可能发生缺页中断, 从磁盘把对应的页号装入到内存物理块。假设此时物理内存还有足够的空闲块, 则不需要淘汰已经装入的页面。设找到的空闲块号为 7, 则逻辑地址 12385 所对应的物理

地址为： $7 \times 4096 + 100 = 28772$ 。

3. (原书 第 8 题)【2009 年统考真题】请求分页管理系统中，假设某进程的页表内容如下表所示。

页号	页框号	有效位 (存在位)
0	101H	1
1	--	0
2	254H	1

页面大小为 4KB，一次内存的访问时间是 100ns，一次快表 (TLB) 的访问时间是 10ns，处理一次缺页的平均时间为 10^8 ns (已含更新 TLB 和页表的时间)，进程的驻留集大小固定为 2，采用最近最少使用置算法 (LRU) 和局部淘汰策略。假设

- (1). TLB 初始为空；
- (2). 地址转换时先访问 TLB，若 TLB 未命中，再访问页表 (忽略访问页表之后的 TLB 更新时间)；
- (3). 有效位为 0 表示页面不在内存，产生缺页中断，缺页中断处理后，返回到产生缺页中断的指令处重新执行。

设有虚地址访问序列 2362H、1565H、25A5H，请问：

- (1). 依次访问上述三个虚地址，各需多少时间？给出计算过程。
- (2). 基于上述访问序列，虚地址 1565H 的物理地址是多少？请说明理由。

【解析】

(1). 因为页大小为 4KB，所以虚地址中的低 12 位表示页内地址，剩余高位表示页号。则十六进制虚地址的低三位为页内地址，最高位为页号。

1) 2362H：页号为 2，页内地址为 362H。先访问快表 10ns，未命中，再访问内存中的页表 100ns，页表项中的有效位指示该页在内存，根据该页对应的页框号形成物理地址再次访问内存 100ns，共计 $10\text{ns} + 100\text{ns} \times 2 = 210\text{ns}$ 。

2) 1565H：页号为 1，页内地址为 565H。先访问快表 10ns，未命中，再访问内存中的页表 100ns，页表项中的有效位指示该页不在内存，处理缺页 10^8 ns，再次访问快表 10ns 命中，根据该页对应的页框号形成物理地址再次访问内存 100ns，共计 $10\text{ns} + 100\text{ns} + 10^8\text{ns} + 100\text{ns} + 10\text{ns} \approx 10^8\text{ns}$ 。

3) 25A5H：页号为 2，页内地址为 5A5H。由于访问 2362H 时已将页 2 的表项写入 TLB，因此访问快表 10ns，命中，根据该页对应的页框号形成物理地址访问内存 100ns，共

计 $10\text{ns}+100\text{ns}=110\text{ns}$ 。

(2). 虚地址 1565H 的页号为 1, 页内地址为 565H。目前页 0、页 2 在内存, 访问页 1 时发生缺页, 根据 LRU 置换算法和局部淘汰策略, 将页 0 换出页 1 换入, 因此页 1 对应的页框号为 101H, 又块内地址为 565H, 则虚地址 1565H 的物理地址为 101565H。

考点 7 页面置换算法和抖动

温馨提示: 本考点主要考查页面置换算法, 是本章的核心考点, 请同学们掌握最佳置换算法、先进先出页面置换算法、最近最久未使用页面置换算法和时钟页面置换算法, 并能够将这几种算法用于解题。也请大家了解抖动、异常、工作集等基本的概念。

一. 选择题部分

1. (原书 第 1 题) 系统抖动是指 ()。
- A. 使用机器时, 出现屏幕闪烁的现象。
 - B. 刚被调出的帧又立刻被调入所形成的频繁调入调出的现象。
 - C. 系统盘不净, 千万系统不稳定的现象。
 - D. 由于内存分配不当, 偶然造成内存不够的现象。

【解析】 本题考查抖动的基本概念。**抖动(Thrashing)**就是指当内存中已无空闲空间而又发生缺页中断时, 需要从内存中调出一页程序或数据送磁盘的对换区中, 如果算法不适当, 刚被换出的页很快被访问, 需重新调入, 因此需再选一页调出, 而此时被换出的页很快又要被访问, 因而又需将它调入, 如此频繁更换页面, 以致花费大量的时间, 我们称这种现象为“抖动”。故而, 选择 B 答案。

参考答案: B

2. (原书 第 3 题) 下列页面置换算法中, 会产生所谓 Belady 异常现象的是 ()。
- A. 最佳页面置换算法 (OPT)
 - B. 先进先出页面置换算法 (FIFO)
 - C. 最近最久未使用算法 (LRU)
 - D. 时钟页面置换算法 (Clock)

【解析】本题考查 Belady 异常。一般堆栈类算法（如 LRU）都不会出现 Belady 异常。FIFO 算法是我们常见的几个页面置换算法中基于队列实现的算法，非堆栈类算法。FIFO 算法可能会产生 Belady 异常。

【注意】Belady 异常不同于系统抖动，前者指分配给进程的物理块增加缺页中断的次数不减少反而增加的现象；后者指由于页面置换算法选择不当使得系统出现频繁的页面调入调出现象，刚被换出的页面，又被换入内存。

参考答案：B

3. (原书 第 5 题)【2011 年统考】当系统发生抖动 (thrashing) 时，可以采取的有效措施是 ()。

- I. 撤销部分进程 II. 增加磁盘交换区的容量 III. 提高用户进程的优先级
A. 仅 I B. 仅 II C. 仅 III D. I、II

【解析】本题考查抖动的应对办法。在虚存中，页面在内存和外存之间频繁的调度，以至于调度页面所需时间比进程实际运行的时间还多，此时系统效率急剧下降，甚至导致系统崩溃，这种现象称为抖动。

一般应对抖动的措施是：

- (1). **撤销部分进程**。撤销部分进程可以减少所用到的页面数，减少页面的调入调出。
- (2). **增加内存**。增加内存之后，进程可以分配到较大的内存，使得程序在运行的过程中缺页次数减少，调入调出也会相应减少。
- (3). **该用页面置换算法**。产生抖动的原因，主要是因为页面置换算法不合理，分配给进程的物理页面太少。故而，除了多分配给进程物理页面数量之外，就是调整页面置换算法了。

在系统发生抖动时，增加磁盘交换区和提高用户进程优先级都不是解决办法，因为二者与抖动无关。故而，本题选择 A 答案。

参考答案：A

4. (原书 第 7 题)在请求分页管理中，若采用先进先出 (FIFO) 页面置换算法，可能会产生“FIFO 异常”，“FIFO 异常”指的是 ()。

- A. 频繁地出页入页的现象
B. 分配的页面数增加，缺页中断的次数也可能增加
C. 进程交换的信息量过大导致系统工作区不足
D. 分配给进程的内存空间不足使进程无法正常工作

【解析】一般而言，分配给进程的物理块越多，运行时缺页次数应该越少。但是 Belady 在 1969 年发现了一个反例：使用 FIFO 算法时，给进程分配多个物理块时，进程的缺页次数比给进程分配少个物理块时还要多，这种反常的现象叫做 Belady 异常。故而，本题选择 B 答案。

参考答案： B

二. 综合应用题部分

1. (原书 第1题) 设某作业占有 7 个页面，如果在主存中只允许装入 4 个工作页面(即工作集为 4)，作业运行时，实际访问页面的顺序是：1， 2， 3， 6， 4， 7， 3， 2， 1， 4， 7， 5， 6， 5， 2， 1。试用 FIFO、LRU 和 CLOCK 页面置换算法，列出各自的页面淘汰顺序和缺页次数。

【解析】

(1). 采用 FIFO 算法，得到页面替换情况如表 4.5 所示。

表 4.5

访问序列	1	2	3	6	4	7	3	2	1	4	7	5	6	5	2	1
物理页 0	1	1	1	1	4	4	4	4	4	4	4	5	5	5	5	5
物理页 1		2	2	2	2	7	7	7	7	7	7	7	6	6	6	6
物理页 2			3	3	3	3	3	2	2	2	2	2	2	2	2	2
物理页 3				6	6	6	6	6	1	1	1	1	1	1	1	1
缺页否?	√	√	√	√	√	√		√	√			√	√			

由上表可知，共发生 10 次缺页。

(2). 采用 LRU 算法，页面置换情况如表 4.6 所示。

表 4.6

访问序列	1	2	3	6	4	7	3	2	1	4	7	5	6	5	2	1
物理页 0	1	1	1	1	4	4	4	4	1	1	1	1	6	6	6	6
物理页 1		2	2	2	2	7	7	7	7	4	4	4	4	4	2	2
物理页 2			3	3	3	3	3	3	3	3	7	7	7	7	7	1
物理页 3				6	6	6	6	2	2	2	2	5	5	5	5	5
缺页否?	√	√	√	√	√	√		√	√	√	√	√	√		√	√

由上表可知，共发生 14 次缺页。

(3). 采用 CLOCK 算法, 页面置换情况如表 4.7 所示。

表 4.7

访问序列	1	2	3	6	4	7	3	2	1	4	7	5	6	5	2	1
物理页 0	1	1	1	1	4	4	4	4	1	1	1	1	6	6	6	6
物理页 1		2	2	2	2	7	7	7	7	4	4	4	4	4	2	2
物理页 2			3	3	3	3	3	3	3	3	7	7	7	7	7	1
物理页 3				6	6	6	6	2	2	2	2	5	5	5	5	5
缺页否?	√	√	√	√	√	√		√	√	√	√	√	√		√	√

由上表可知, 缺页次数为 14 次。

2. (原书 第 4 题) 考虑下面的页访问串:

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

假定有 4、5、6 个页块, 应用下面的页面替换算法, 计算各回出现多少次缺页中断。注意, 所给定的页块初始均为空, 因此首次访问一页时就会发生缺页中断。

- (1). LRU 页面替换算法。
- (2). FIFO 页面替换算法。
- (3). Optimal 页面替换算法。

【解析】

(1). 采用 LRU 页面替换算法, 当物理块为 4 时, 页面置换情况如表 4.8 所示。

表 4.8 物理块为 4 时, LRU 页面替换算法页面置换情况表

访问序列	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
物理块 0	1	1	1	1	1	1	1	1	1	1	1	1	1	6	6	6	6	6	6	6
物理块 1		2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
物理块 2			3	3	3	3	5	5	5	5	5	3	3	3	3	3	3	3	3	3
物理块 3				4	4	4	4	6	6	6	6	6	7	7	7	7	1	1	1	1
缺页否?	√	√	√	√			√	√				√	√	√			√			

由表 4.8 可知, 采用 LRU 算法, 内存块为 4 时, 共产生 10 次缺页中断。

采用 LRU 算法, 当内存块为 5 时, 页面置换情况如表 4.9 所示。

表 4.9 物理块为 5 时, LRU 页面替换算法页面置换情况表

访问序列	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
物理块 0	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1
物理块 1		2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
物理块 2			3	3	3	3	3	6	6	6	6	6	6	6	6	6	6	6	6	6
物理块 3				4	4	4	4	4	4	4	4	3	3	3	3	3	3	3	3	3
物理块 4							5	5	5	5	5	5	7	7	7	7	7	7	7	7
缺页否?	√	√	√	√			√	√				√	√							

由表 4.9 可知, 采用 LRU 算法, 内存块为 5 时, 共产生 8 次缺页中断。

表 4.10 物理块为 6 时, LRU 页面替换算法页面置换情况表

访问序列	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
物理块 0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
物理块 1		2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
物理块 2			3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
物理块 3				4	4	4	4	4	4	4	4	4	7	7	7	7	7	7	7	7
物理块 4							5	5	5	5	5	5	5	5	5	5	5	5	5	5
物理块 5								6	6	6	6	6	6	6	6	6	6	6	6	6
缺页否?	√	√	√	√			√	√					√							

由表 4.10 可知, 采用 LRU 算法, 内存块为 6 时, 共产生 7 次缺页中断。

(2). 采用 FIFO 页面替换算法, 当物理块为 4 时, 页面置换情况如表 4.11 所示。

表 4.11 物理块为 4 时, FIFO 页面替换算法页面置换情况表

访问序列	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
物理块 0	1	1	1	1	1	1	5	5	5	5	5	3	3	3	3	3	1	1	1	1
物理块 1		2	2	2	2	2	2	6	6	6	6	6	7	7	7	7	7	7	3	3
物理块 2			3	3	3	3	3	3	2	2	2	2	2	6	6	6	6	6	6	6
物理块 3				4	4	4	4	4	4	1	1	1	1	1	1	2	2	2	2	2
缺页否?	√	√	√	√			√	√	√	√		√	√	√		√	√		√	

由表 4.11 可知, 采用 FIFO 算法, 内存块为 4 时, 共产生 14 次缺页中断。

同样的, 采用 FIFO 算法, 当物理块为 5 时, 页面置换情况如表 4.12 所示。

表 4.12 物理块为 5 时, FIFO 页面替换算法页面置换情况表

访问序列	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
物理块 0	1	1	1	1	1	1	1	6	6	6	6	6	6	6	6	6	6	6	6	6
物理块 1		2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1
物理块 2			3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2
物理块 3				4	4	4	4	4	4	4	4	3	3	3	3	3	3	3	3	3
物理块 4							5	5	5	5	5	5	7	7	7	7	7	7	7	7
缺页否?	√	√	√	√			√	√		√	√	√	√							

由表 4.12 可知, 采用 FIFO 算法, 内存块为 5 时, 共产生 10 次缺页中断。

表 4.13 物理块为 6 时, FIFO 页面替换算法页面置换情况表

访问序列	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
物理块 0	1	1	1	1	1	1	1	1	1	1	1	1	7	7	7	7	7	7	7	7
物理块 1		2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1
物理块 2			3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	2
物理块 3				4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	3
物理块 4							5	5	5	5	5	5	5	5	5	5	5	5	5	5
物理块 5								6	6	6	6	6	6	6	6	6	6	6	6	6
缺页否?	√	√	√	√			√	√					√				√	√	√	

由表 4.13 可知, 采用 LRU 算法, 内存块为 6 时, 共产生 10 次缺页中断。

(3). 采用 Optimal 页面替换算法, 页面置换情况如下表 4.14 所示。

表 4.14 物理块为 4 时, Optimal 页面替换算法页面置换情况表

访问序列	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
物理块 0	1	1	1	1	1	1	1	1	1	1	1	1	7	7	7	7	1	1	1	1
物理块 1		2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
物理块 2			3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
物理块 3				4	4	4	5	6	6	6	6	6	6	6	6	6	6	6	6	6
缺页否?	√	√	√	√			√	√					√				√			

由表 4.14 可知, 采用 Optimal 算法, 内存块为 4 时, 共产生 8 次缺页中断。

表 4.15 物理块为 5 时, Optimal 页面替换算法页面置换情况表

访问序列	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
物理块 0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
物理块 1		2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
物理块 2			3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
物理块 3				4	4	4	4	6	6	6	6	6	6	6	6	6	6	6	6	6
物理块 4							5	5	5	5	5	5	7	7	7	7	7	7	7	7
缺页否?	√	√	√	√			√	√					√							

由表 4.15 可知, 采用 Optimal 算法, 内存块为 5 时, 共产生 7 次缺页中断。

表 4.16 物理块为 6 时, Optimal 页面替换算法页面置换情况表

访问序列	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
物理块 0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
物理块 1		2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
物理块 2			3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
物理块 3				4	4	4	4	4	4	4	4	4	7	5	5	5	5	5	5	5
物理块 4							5	5	5	5	5	5	5	6	6	6	6	6	6	6
物理块 5								6	6	6	6	6	6	7	7	7	7	7	7	7
缺页否?	√	√	√	√			√	√					√							

由表 4.16 可知, 采用 Optimal 算法, 内存块为 6 时, 共产生 7 次缺页中断。

考点 8 请求分段管理方式和请求段页式管理方式

温馨提示: 本考点考查请求分页存储管理方式和请求段页式存储管理方式, 请同学们注意这两种存储管理方式下, 逻辑地址到物理地址的转换。

一. 选择题部分

- (原书 第 2 题) 段页式存储管理汲取了页式管理和段式管理的长处, 其实现原理结合了页式和段式管理的基本思想, 即 ()。
A. 用分段方法来分配和管理物理存储空间, 用分页方法来管理用户地址空间。

- B. 用分段方法来分配和管理用户地址空间, 用分页方法来管理物理存储空间。
- C. 用分段方法来分配和管理主存空间, 用分页方法来管理辅存空间。
- D. 用分段方法来分配和管理辅存空间, 用分页方法来管理主存空间。

【解析】段是由用户确定的, 页是由操作系统确定的。操作系统的段是用户可见的, 页则是操作系统能感知的。故而, 利用分段的方法来分配和管理用户地址空间, 而采用分页的方法来管理物理存储空间。因而, 本题选择 B 答案。

参考答案: B

2. (原书 第 3 题) 段页式管理中, 地址映像表是 ()。

- A. 每个作业或进程一张段表, 一张页表
- B. 每个作业或进程的每个段一张段表, 一张页表
- C. 每个作业或进程一张段表, 每个段一张页表
- D. 每个作业一张页表, 每个段一张段表

【解析】在段页式管理中, 每个作业有一张段表, 记录了每一个段的页表起始地址和页表长度, 每个段有多个页表, 记录了逻辑页号与内存块号的对应关系, 如下图 4.13 所示。

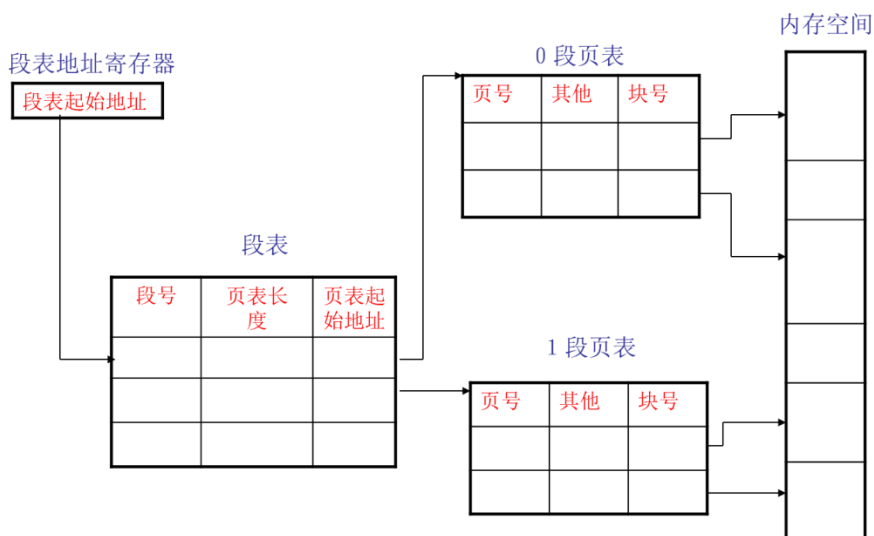


图 4.13

【补充】从上述过程中可知, 若段表、页表存放在内存中, 则为了访问内存的某一条指令或数据, 将需要访问 3 次内存:

第一次, 查找段表获得该段所对应页表的起始地址;

第二次，查找页表获得该页所对应的物理块号，从而形成所需的物理地址；

第三次，根据所得到的物理地址到内存中去访问该地址中的指令或数据。

三次访问内存极大降低了内存的存取速度，可同样采用联想存储器技术提高内存的存取速度。

参考答案：C

3. (原书 第 5 题) 某段表的内容如表 4.17 所示，现执行某条指令 Load 1, 2/154，逻辑地址 2/154（其中段号为 2，段内地址为 154），它对应的物理地址为（ ）。

表 4.17

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

A. 120K+2

B. 480K+154

C. 30K+154

D. 2+480K

【解析】本题考查段式存储管理中将虚拟地址转换成物理地址的计算方法。

暂时先不分析要把逻辑地址 2/154 的内容调到内存的哪儿，我们单看这个逻辑地址，段号为 2，段内地址为 154。很显然，段号为 2 的段，其段首地址为 480K，段长为 20KB。而逻辑地址的段内地址是 154，故而没有访问越界（注意，把逻辑地址转换成物理地址时，要先看看是否越界）。故而该逻辑地址转换到的物理地址为 480K+154，选择 B 答案。

参考答案：B

二. 综合应用题部分

1. (原书 第 3 题) 某段表内容如表 4.18 所示。

表 4.18

段号	段首地址	段长度
0	120K	40K
1	760K	30K

2	480K	20K
3	370K	20K

则一逻辑地址为 (2, 154) 的实际物理地址为多少?

【解析】逻辑地址 (2, 154) 表示段号为 2, 段内偏移为 154。由段表可知, 段号为 2 的段的段首地址为 480K, 段长为 20K, 故而段内偏移合法。154 为单元号, 则实际物理地址为 480K+154。

4. (原书 第 4 题) 某段式存储管理系统的段表如表 4.19 所示。请将逻辑地址 [0, 137]、[1, 9000]、[2, 3600]、[3, 230] 转换成物理地址。

表 4.19

段号	段大小	段起址
0	15 KB	40KB
1	8 KB	80 KB
2	10 KB	100 KB

【解析】本题考查在段式存储管理系统中将逻辑地址转换成物理地址的方法。

(1). 对于逻辑地址 [0, 137], 段号为 0, 小于进程的总段数, 故段号合法。查段表可得该段的段地址为 40KB。段长为 15KB, 段内位移为 137, 故而段内地址合法。因此可得到物理地址为: $40\text{KB}+137\text{B}=40960\text{B}+137\text{B}=41097\text{B}$ 。

(2). 对于逻辑地址 [1, 9000], 段号为 1, 小于进程的总段数, 故段号合法。查段表的项得到该段的段地址为 80KB。该段段长为 8KB, 段内位移 9000 大于段长 8KB=8192B。因此, 产生越界中断。

(3). 对于逻辑地址 [2, 3600], 段号为 2, 小于进程的总段数, 故段号合法, 查段表的项得到该段的段地址为 100KB。段内位移为 3600, 段长为 10KB。段内位移 3600 小于段长 10KB, 故段内地址合法。因此, 可得到物理地址为: $100\text{KB}+3600\text{B}=10240\text{B}+3600\text{B}=10600\text{B}$ 。

(4). 对于逻辑地址 [3, 230], 段号为 3, 段内位移为 230。由于段号 3 大于进程的总段数 2, 故段号不合法, 因此产生越界中断。

本章到此就结束了，你觉得本章有什么地方不明白吗？任何问题，欢迎您与我们作者进行交流！



shareOurDreams
梦享团队微信号



weCSdream
梦享团队官方微信公众号



梦享论坛团队
梦享团队新浪微博

第 5 章 文件管理

考点 1 文件系统的基本概念

温馨提示：本考点考查文件系统的基本概念，包括文件的定义、文件的属性、文件的基本操作三个部分。考查的都是基本内容，请同学们作一个初步的了解。

一. 选择题部分

1. (原书 第 2 题)下面说法正确的是 ()。
- A. 文件系统负责文件存储空间的管理但不能实现文件名到物理地址的转换
 - B. 在多级目录结构中对文件的访问是通过路径名和用户目录名进行的
 - C. 文件可以被划分成大小相等的若干物理块且物理块大小也可任意指定
 - D. 逻辑记录是对文件进行存取操作的基本单位

【解析】文件的主要功能是实现对文件的按名存取，实现了文件名到物理地址的转换，A 答案错误。多级目录文件结构中，对文件的访问使用的是路径名和文件名，而不是路径名和用户目录名，B 答案错误。二级目录文件的查找则是主目录项+用户目录项，如图 5.1 所示。

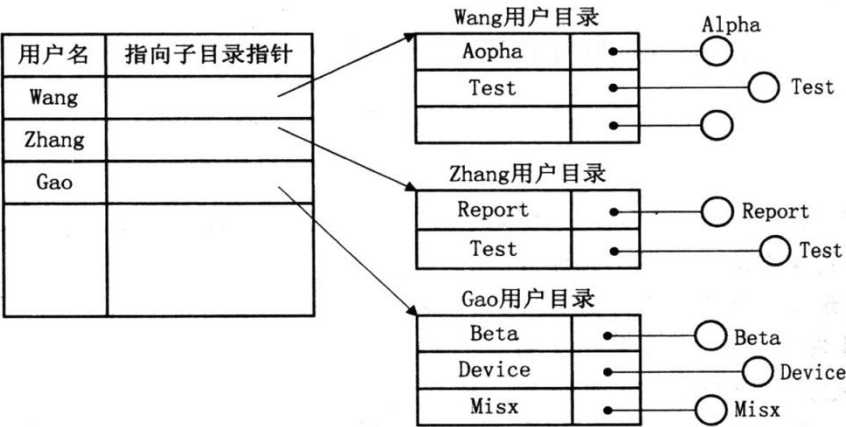


图 5.1 文件二级目录

物理块大小的划分，并不是任意改变的，一般和内存块一般大小，C 答案错误。逻辑记录时文件存取中的基本单位，数据项是文件可使用的最小单位。

参考答案：D

2. (原书 第 5 题) 文件系统中，设立打开文件 (Open) 系统功能调用的基本操作是 ()，关闭文件 (Close) 系统功能调用的基本操作是 ()。

(1)

- A. 把文件信息从辅存读到内存
- B. 把文件的控制管理信息从辅存读到内存
- C. 把文件的 FAT 表信息从辅存读到内存
- D. 把磁盘的超级块从辅存读到内存

(2)

- A. 把文件的最新信息从内存写入磁盘
- B. 把文件当前的控制管理信息从内存写入磁盘
- C. 把位示图从内存写回磁盘
- D. 把超级块的当前信息从内存写回磁盘

【解析】本题考查文件系统中打开文件和关闭文件两个系统调用的基本操作。

文件打开命令的主要功能是：把指定文件的控制信息（在 FCB 中）复制到内存的活动目录表中，以便随后对文件进行的各种操作，可以直接从活动目录表里获得该文件的信息。这样做可以避免每次涉及到文件时，都要与磁盘交往，从而快速进行后续的文件访问工作。

在读/写完毕后，需要执行“关闭文件”操作，以便将该文件的目录信息从主存中撤销。执行“关闭”操作时要检查读到主存中的文件目录或检索表是否被修改过，若被修改过，则应把修改过的文件目录或索引表重新保存号。一个关闭后的文件不能再使用，若要再使用，则必须再次执行“打开”操作。用户提出“关闭”要求时，必须说明关闭哪个文件。

参考答案：B B

3. (原书 第 8 题) 【2009 年统考真题】文件系统中，文件访问控制信息存储的合理位置是 ()。

- A. 文件控制块
- B. 文件分配表
- C. 用户口令表
- D. 系统注册表

【解析】本题考查文件控制块的结构。文件控制块 FCB 中包含以下三类信息：

(1). **基本信息类**。基本信息类主要包括文件名和文件的物理地址。文件名是用于标识一个文件的符号名，在文件系统中，文件都必须具有唯一的名字。文件的物理地址则根据文件的物理结构不同而不同，如连续文件就是文件的起始块号和文件总块数。

(2). **存取控制信息类**。该类主要包括文件的存取权限。

(3). **使用信息类**。该分类包括文件的建立日期、最后一次修改日期、最后一次访问信息、打开文件的进程数等。

故而，本题选择 A 答案。

参考答案： A

4. (原书 第 9 题) 【2012 年统考真题】若一个用户进程通过 read 系统调用读取一个磁盘文件中的数据，则下列关于此过程的叙述中，正确的是 ()。

I. 若该文件的数据不在内存，则该进程进入睡眠等待状态。

II. 请求 read 系统调用会导致 CPU 从用户态切换到核心态。

III. read 系统调用的参数应包含文件的名称

A. 仅 I、II

B. 仅 II

C. 仅 III

D. I、II 和 III

【解析】对于 I，当要访问的数据在磁盘上而为调入内存，产生缺页中断，进程由运行态转入阻塞态。当所请求的数据都调入内存以后，进程被唤醒，继续执行。

对于 II，我们常说 I/O 操作会用到系统调用，同理，read 系统调用通过内陷使得 CPU 从用户态转入核心态执行，获得操作系统提供的服务。

对于 III，read 要求用户提供文件描述符 fd、缓冲区首地址 buff、传送的字节数 n 三个参数，而不使用文件名作为参数。read 通过这三个参数，完成一个功能：从文件描述符 fd 中读入 n 个字节，并把这些字节传送到以 buff 为首地址的缓冲区内。

Open 系统调用用于打开文件，使用参数为文件的路径名和文件名。

参考答案： A

考点 2 文件结构

本部分主要考查文件的结构，包括文件的物理结构和逻辑结构。请同学们主要了解文件的逻辑结构，包括流式文件和记录式文件。其中，记录式文件又包括顺序文件、索引文件、索引顺序文件和直接文件。这几类文件需要掌握。

一. 选择题部分

1. (原书 第 4 题)按文件的物理组织结构可将文件分成()等。
- A. 数据文件，命令文件，文本文件
 - B. 命令文件，库文件，索引文件
 - C. 连续文件，链式文件，索引文件
 - D. 输入文件，输出文件，随机文件

【解析】本题考查文件的物理组织结构分类。文件的物理组织结构可分为连续文件、链式(链接)文件、索引文件三类。为每一个文件分配一组相邻接的盘块，并把逻辑文件中的记录顺序的存储到邻接的各物理盘块中的文件结构称为(连续)顺序文件结构(如图 5.2 所示)。该分配方法具有顺序访问容易，并可实现直接存取，访问速度快等优点，但是要求有连续的存储空间、必须事先知道文件的长度。

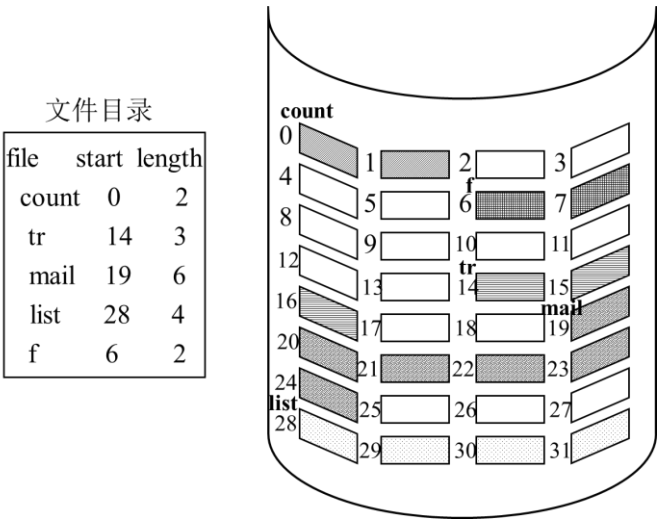


图 5.2 连续(顺序)文件

通过在每个盘块上的链接指针，将同属于一个文件的多个离散的盘块链接成一个链表，形成链接式文件结构，该物理文件称为链接文件。隐式链接如图 5.3 所示。

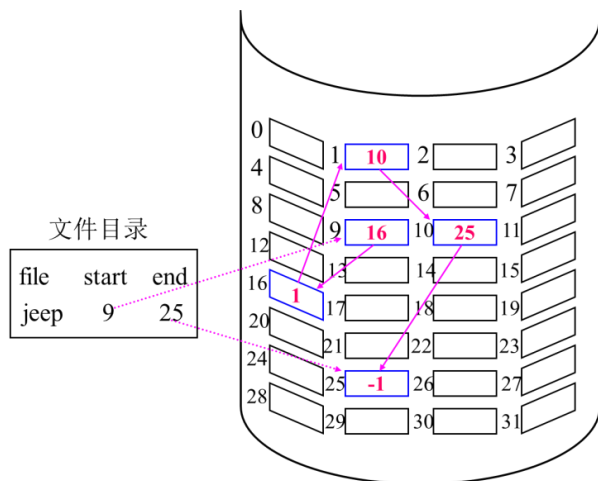


图 5.3 隐式链接文件

隐式链接只适合顺序访问，对随机存取极其低效。隐式链接方法仅通过链接指针实现各离散盘块的链接，只要其中任何一个指针出现问题，都会导致整条链断开，所以可靠性较差。

显示链接则如图 5.4 所示。

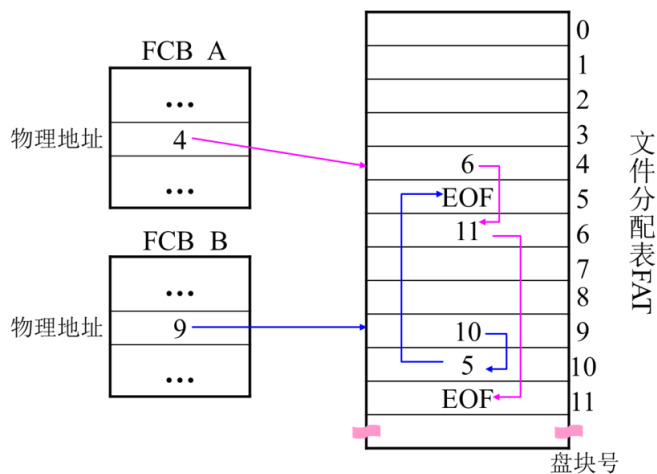


图 5.4 显式链接文件

文件索引分配（如图 5.5 所示）的**基本思想**是：文件打开仅须把该文件所占用盘块编号调入内存即可，故可将每个文件所对应的盘块号集中地存放一个所谓的索引块中，形成一个索引表。

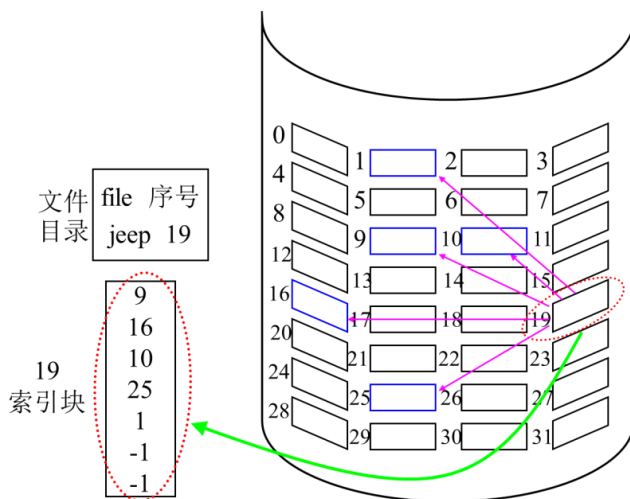


图 5.5 索引文件

值得注意的是，对于小文件，索引分配方式的索引块利用率极低。

参考答案: C

2. (原书第7题)逻辑文件是()的文件组织形式。
- A. 在外部设备上
B. 从用户观点看
C. 虚拟存储
D. 目录

【解析】逻辑结构是从用户视角出发观察到的文件的组织形式，是用户可以直接处理的数据及其结构。物理结构则是文件的存储结构，是文件在外存上的组织方式。文件的物理结构和逻辑结构完全独立。

参考答案: B

3. (原书第8题)下列文件中属于逻辑结构的文件是()。
- A. 连续文件 B. 系统文件
- C. 散列文件 D. 流式文件

【解析】文件从逻辑结构上可分为有结构文件和无结构文件。有结构文件（记录式文件），可分为定长记录和不定长记录两种方式。对于不定长记录，可再分为顺序文件、索引文件和索引顺序文件。无结构文件，由字符流组成，又叫做流式文件。

参考答案: D

3. (原书 第 11 题) 【2009 年统考真题】下列文件物理结构中, 适合随机访问且易于文件

管理操作。文件与文件控制块一一对应，而人们把文件控制块的有序集合称为文件目录，即一个文件控制块就是一个文件目录项。通常，一个文件目录也被看做是一个文件，称为目录文件。

参考答案： C

2. (原书 第 3 题)假定磁盘的大小为 1K，对于 1.2MB 的软盘，FAT 需占用 () 的存储空间；对于 100MB 的硬盘，FAT 需占用 () 的存储空间。(假设为 FAT 12)

(1)

- A. 1KB B. 1.5KB C. 1.8KB D. 2.4KB

(2)

- A. 100KB B. 150KB C. 200KB D. 250KB

【解析】FAT 12 的盘块表项是 12 位。每个盘块的大小是 1KB，对于 1.2MB 的软盘，需要 $1.2\text{MB}/1\text{KB}=1.2\text{K}$ 个表项。故而，FAT 的大小为 $1.2\text{K} \times 1.5\text{B} = 1.8\text{KB}$ 。

对于 100MB 的硬盘，需要 $100\text{MB}/1\text{KB}=100\text{K}$ 个表项，每个表项大小为 1.5B，故而 FAT 的大小应该为 $100\text{K} \times 1.5\text{B}=150\text{KB}$ 。

参考答案： C D

3. (原书 第 6 题)如果允许不同用户的文件可以具有相同的文件名，通常采用 () 来保证按名存取的安全。

- A. 重名翻译机构 B. 建立索引表
C. 建立指针 D. 多级目录结构

【解析】本题考查了多级目录允许文件重名的特点。多级目录结构有层次结构清晰，便于管理和保护，解决重名问题，查找速度快等优点。采用多级目录，不同用户的文件可以有相同的文件名，但是要注意，这些相同的文件名不位于同一个目录下。

参考答案： D

4. (原书 第 13 题)操作系统中对目录管理的主要要求，不包括 () 。

- A. 对文件实现按名存取
B. 节省文件存储空间
C. 提高对目录的检索速度
D. 允许文件重名

【解析】本题考查目录管理的目标。操作系统对文件目录管理的主要要求，即文件目录具有的功能，主要有以下几点：

(1). 实现“按名存取”。用户只需提供文件名就可以对文件进行操作。这是目录挂历最基本的功能。

(2). 提高文件目录检索速度。对于大型文件系统，通常采用三级或三级以上的目录结构，以提高对目录的检索速度和文件系统的性能。

(3). 允许文件同名。不同目录下的文件可以使用相同的名字。【注意】同一目录下仍不能有相同的文件名。

(4). 运行文件共享。

参考答案： B

5. (原书 第 20 题) 文件系统中，把 FCB 分为次部和主部的好处是 ()。

- A. 提高文件的查找速度
- B. 减少 FCB 所占空间
- C. 防止进程修改 FCB 信息
- D. 减少文件 I/O 操作的时间

【解析】为找到一个目录项，平均需要调入盘块 $N/2$ 次，每调入一个盘块即启动磁盘一次，其中 N 为目录文件所占用的盘块数。因为一个 FCB 占用 64B，盘块的大小时 1KB，故一个盘块可以存放 16 个 FCB。故而，文件目录中共有的 3200 个 FCB 需要 200 个磁盘块来存放，平均启动磁盘的次数为 100 次。

参考答案： C

二. 综合应用题部分

1. (原书 第 5 题) 有一个文件系统如图 5.6 所示。图中的框表示目录，圆圈表示普通文件。根目录常驻内存，目录文件组织成链接文件，不设文件控制块，普通文件组织成索引文件。目录表目指示下一级文件名及其磁盘地址（各占 2 个字节，共 4 个字节）。若下一级文件是目录文件，指示其第一个磁盘块地址。若下级文件是普通文件，指示其文件控制块的磁盘地址。每个目录文件磁盘块最后 4 个字节供拉链使用。下级文件在上级目录文件中的次序在图中为从左至右。每个磁盘块有 512 字节，与普通文件的一页等长。

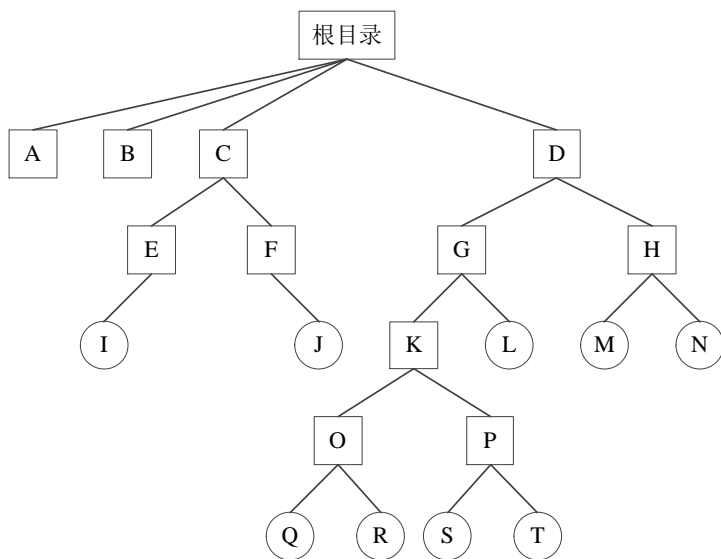


图 5.6 文件系统结构示意图

普通文件的文件控制块组织描述如下：每个磁盘地址占 2 个字节，前 10 个地址直接指示该文件前 10 页的地址；第 11 个地址指示一级索引表地址，一级索引表中每个磁盘地址指示一个文件页地址；第 12 个地址指示二级索引表地址，二级索引表中每个地址指示一个一级索引表地址；第 13 个地址指示三级索引表地址，三级索引表中每个地址指示一个二级索引表地址。问：

- (1) 一个普通文件最多可有多少个文件页？
- (2) 若要读文件 J 中某一页，最多启动磁盘多少次？
- (3) 若要读文件 W 中的某一页，最少启动磁盘多少次？
- (4) 就 (3) 而言，为最大限度减少启动磁盘次数，可采用什么方法？此时，磁盘最多启动多少次？

【解析】

(1). 由题目中所给条件可知，磁盘块大小 512 个字节，每个磁盘地址占 2 个字节，因此一个一级索引可容纳 256 个磁盘地址。同理，一个二级索引表可容纳 256 个一级索引表地址，一个三级索引表可容纳 256 个二级索引表地址。

因为普通文件的文件控制块共 13 个地址项，10 个直接地址，1 个 1 级索引地址，1 个二级索引地址，一个三级索引地址。故而，一个普通文件最多可有的页数为：
 $10+256+256 \times 256+256 \times 256 \times 256=16843018$ 。

(2). 从图 5.11 中可以看出，目录文件 A 和目录文件 D 中目录项都只有 2 个，因此这两

个目录文件都不需要拉链。若要读取文件 J 中某一页，首先从内存的根目录中找到目录文件 A 的磁盘地址，将其读入内存（第 1 次读磁盘）。然后再从目录 A 中找到 D 的磁盘地址，并将其读入内存（第 2 次读磁盘）。从 D 目录中找到文件 J 的文件控制块地址，将其读入内存（第 3 次读磁盘）。在最坏的情况下，要访问页的地址需通过三级索引才能找到这时要三次读取磁盘才能将三级索引读入内存（第 4、5、6 次读磁盘）。最后读入文件 J 的相应页（第 7 次读磁盘）。由此可知，若要读文件 J 中某一页，最多启动磁盘 7 次。

(3). 从图 5.11 中可以看出，目录文件 C 和目录文件 U 中，目录项数目较多。每个盘块可以放 128 个目录项，而最后 4 个字节用作链接，若目录项数超过 127，则目录文件的读入可能需要多次读盘（因目录文件组织成链接文件）。

在最好情况下，所找的目录项都在目录的第一个磁盘块中。若要读文件 W 的某一页，首先从内存的根目录中找到目录文件 C 的地址，将其读入内存（第 1 次读磁盘）。在最好情况下，能从目录 C 的第一个盘块中找到目录文件 I 的地址，将其读入内存（第 2 次读磁盘）。从目录 I 中找到目录文件 P 的地址，将其读入内存（第 3 次读磁盘）。从目录 P 中找到目录文件 U 的地址，将其读入内存（第 4 次读磁盘）。在最好情况下，能从目录 U 的第一个盘块中找到目录文件 W 的文件控制块的地址，将其读入内存（第 5 次读磁盘）。在最好情况下，要访问的页在前 10 页中，这时可以直接读到该页的地址。最后读入文件 W 的相应页（第 6 次读磁盘）。由此可知，若要读文件 W 中的某一页，最少启动磁盘 6 次。

(4). 由于通过文件控制块访问文件所需的读盘次数无法改变，要减少访问磁盘次数，只有通过减少访问目录文件的次数来达到。为最大限度地减少启动磁盘次数，可将文件 W 直接链接在根目录的最左端（其目录项在根目录的前 127 个目录项内）。这样，若要读文件 W 的某页时，首先从内存的根目录中找到文件 W 的文件控制块地址，将文件 W 的文件控制块读入内存（第 1 次读磁盘）。在最好情况下，要访问的页在前 10 页中，这时可以直接读到该页的地址。在最坏的情况下，要访问页的地址需三级索引才能找到，这时要三次访问磁盘（第 2、3、4 次读磁盘）。最后读入文件 W 的相应页（最好情况下，第 2 次读盘；最坏情况下，第 5 次读磁盘）。由此可知，若将文件 W 直接链接在根目录的最左端，要读文件 W 中的某一页，最少启动磁盘 2 次，最多启动磁盘 5 次。

考点 4 文件共享和保护

温馨提示：本考点考查文件的共享和保护。文件共享方面，请同学们掌握基于索引结点和符号链的文件共享方式的特点和区别。文件保护方面，请同学们了解常见的文件保护方式。

一. 选择题部分

1. (原书 第1题) 利用基本文件目录法实现文件共享, 文件系统必须设置一个(), 每个用户都应具有一个 ()。

(1)

- | | |
|-----------|--------|
| A. 系统文件目录 | B. 主目录 |
| C. 基本文件目录 | D. 根目录 |

(2)

- | | |
|-----------|---------|
| A. 用户文件目录 | B. 符号目录 |
| C. 基本文件目录 | D. 当前目录 |

【解析】文件共享是指一个文件被多个用户或程序使用。利用基本文件目录法实现文件共享, 文件系统必须设置一个系统文件目录, 每个用户必须具有一个用户文件目录。

【补充】有三种共享形式:

- (1). 被多个用户使用, 由存取权限控制。
- (2). 被多个程序使用, 但各用自己的读写指针。
- (3). 被多个程序使用, 但共享读写指针。

参考答案: C A

2. (原书 第3题) 允许多个用户同时使用同一个共享文件时, 下列说法错误的是()。
- 允许多个用户同时打开共享文件执行读操作
 - 允许读者和写者同时使用共享文件
 - 不允许读者和写者同时使用共享文件
 - 不允许多个写者同时对共享文件执行写操作

【解析】关于读者写者的问题, 我们在第二章进程同步中讲了很多。允许多个用户同时使用同一个共享文件, 并且允许多个用户对该文件执行读操作。因为, 读操作不会改变

文件的内容，用户不管何时访问文件，读到的内容都是一样的。但是要对文件的写操作进行限制，不允许多个用户同时写文件，或者一些用户在读另一些用户在写的同时进行读写的情况。

设想一下，比如用户 A 和用户 C 要读文件，用户 B 要写文件，若不进行限制，则 A 读完之后，B 则写文件，C 再读文件时，与 A 读到的可能不一样，A 读到的是 B 修改前的文件，C 读到的是 B 修改后的文件。还有可能 A、C 读到的都是 B 修改的半成品，都不是自己想要读取的文件内容。

因而，不允许读者与写者同时使用共享文件，写者与写者之间也不能同时对文件执行写操作，他们得互斥对文件操作文件。

参考答案：B

考点 5 文件系统的实现

温馨提示：本考点考查文件系统的实现，包括目录实现和文件实现。文件实现方面，请同学们着重了解文件的几种实现方式，即连续分配方式、链接分配和索引分配。这一部分比较重要，请同学们多练习。

一. 选择题部分

1. (原书 第 1 题)位示图法可用于()。

- A. 文件目录的查找
- B. 分页式存储管理中主存空闲块的分配和回收
- C. 磁盘空闲盘块的分配和回收
- D. 页式虚拟存储管理中的页面置换

【解析】 本题考查位示图法的作用。位示图利用二进制的一位来表示磁盘中一个盘块的使用情况，当该位的值为“0”时，表示对应的盘块空闲，为“1”则表示已分配。磁盘上的所有盘块都有一个二进制位与之对应，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 $m \times n$ 个数来构成位示图，并使 $m \times n$ 等于磁盘的总块数。

位示图法是外存分配管理算法，不是主存分配算法，该算法是操作系统提供对存储空间进行分配和回收的手段。

参考答案：C

2. (原书 第 4 题)UNIX 系统磁盘存储空间的管理采用 () 的管理方法。

- A. 位示图
- B. 记录的成组操作
- C. 空闲块表
- D. 空闲块成组链接

【解析】本题考查 UNIX 系统磁盘存储空间的管理方法。

UNIX 系统采用空闲块成组连接方法来管理磁盘空间。UNIX 系统把每 100 个空闲块作为一组，每一组的第一个空闲块中登记下一组空闲块的块号和空闲块数，余下不足 100 块的那部分空闲块的块号及块数登记在一个专用块中，登记最后一组块号的那个空闲块其中第 2 个单元填“0”，表示该块中指出的块号是最后一组的块号，空闲块链到此结束。

参考答案：D

3. (原书 第 7 题)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，一个进程要访问某文件中偏移量为 23456789 字节处的数据时，需要经过 ()。

- A. 直接寻址（相当于一级索引）
- B. 一次间址（相当于二级索引）
- C. 二次间址（相当于三级索引）
- D. 三次间址（相当于四级索引）

【解析】由题意可知，一个盘块的大小是 1KB，而一个盘号占 4B。所以，直接寻址可访问磁盘空间的大小是 $1\text{KB}=1024\text{B}$ ，一次间址可访问磁盘空间的大小是 $1\text{KB} \times 1\text{KB}/4\text{B}=256 \times 1024\text{B}=262144\text{B}$ ，二次间址的可访问磁盘空间的大小是 $64\text{MB}=262144 \times 256=67108864$ 。故而，二次间址即可。

当然，本题若是这样分析，计算量就太大了。可以粗略计算一下，23456789 小于 24M，但是大于 256KB，故而二次间址最合适。

参考答案：C

4. (原书 第 10 题)【2013 年统考真题】若某文件系统索引结点(inode)中有直接地址项和间接地址项，则下列选项中，与单个文件长度无关的因素是 ()。

- A. 索引结点的总数
- B. 间接地址索引的级数
- C. 地址项的个数
- D. 文件块大小

【解析】由第 8 题的计算中，我们可以知道，在计算文件的长度时，间接地址索引的级数、地址项的个数、文件块的大小都是需要考虑的因素。而索引结点的个数，则不是与文件长度相关的因素。

索引节点是一个数据结构，它包含了一个文件的文件名、位置、大小、建立或修改时间、访问权限、所属关系等文件控制信息。一个文件系统维护了一个索引节点的数组，每个文件或目录都与索引结点数组中的唯一一个元素对应。

参考答案：A

二. 综合应用题部分

1. (原书 第 1 题)【2011 年统考真题】某文件系统为一级根目录结构，文件的数据一次性写入磁盘，已写入的文件不可修改，但可多次创建新文件。请回答如下问题。

(1). 在连续、链式、索引三种文件的数据块组织方式中，哪种更合适？要求说明理由。为定位文件数据块，需要在 FCB 中设置哪些相关描述字段？

(2). 为快速找到文件，对于 FCB，是集中存储好，还是与对应的文件数据块连续存储好？要求说明理由。

【解析】

(1). 连续方式更合适。因为一次写入不存在插入问题，而且写入文件之后不需要修改（连续的数据块组织方式很适合一次性写入磁盘不再修改的情况）。同时连续存储相对链式和索引省去了指针的空间开销，支持随机查找，查找速度最快。

在连续方式中，为定位文件数据块，需要在 FCB 中设置文件在外存的起始地址（即首个盘块号）及文件的长度（即文件占用的盘块数）。

(2). FCB 集中存储较好。FCB 中存放了关于描述和控制文件的重要信息，是文件目录的重要组成部分。在检索文件时，通常会访问文件的 FCB。如果将 FCB 集中存储，可减少检索文件时访问磁盘的次数，提高文件的访问速度。

2. (原书 第 4 题)某个磁盘上的文件系统，采用混合索引分配方式，FCB 中共有 13 个地址项，第 0~9 个地址项为直接地址，第 10 个地址项为一次间接地址，第 11 个地址项为二次间接地址，第 12 个地址项为三次间接地址。如果每个盘块的大小为 512 字节，若盘块号需要用 3 个字节来描述，而每个盘块最多存放 170 个盘块地址。

(1). 该文件系统允许文件的最大长度是多少？

(2). 将文件的字节偏移量 5000、15000、150000 转换为物理块号和块内偏移量。

(3). 假设某文件的 FCB 已在内存，但其他信息均在外存，为了访问该文件中某个位置的内容，最少需要几次访问磁盘，最多需要几次访问磁盘？

【解析】

(1). 该文件系统的文件结构能支持的索引盘块最多为 $10 \times 170 + 170 \times 170 + 170 \times 170 \times 170 = 4942080$ 块, 可得知该文件系统中一个文件的最大长度可达 $4942080 \times 512\text{B} = 2471040\text{B}$ 。

(2). 对三个字节偏移量分别分析:

1) 5000/512 得到商为 9, 余数为 392, 即字节偏移量 5000 对应的逻辑块号为 9, 块内偏移量为 392。由于 $9 < 10$, 故可直接从该文件的 FCB 的第 9 个地址项处得到物理盘块号, 块内偏移量为 392。

2) 15000/512 得到商为 29, 余数为 152, 即字节偏移量 15 000 对应的逻辑块号为 29, 块内偏移量为 152。由于 $10 < 29 < 10 + 170$, 而 $29 - 10 = 19$, 故可从 FCB 的第 10 个地址项, 即一次间址项中得到一次间址块的地址; 并从一次间址块的第 19 项(即该块的第 57~59 这 3 个字节)中获得对应的物理盘块号, 块内偏移量为 152。

3) 150000/512 得到商为 292, 余数为 496, 即字节偏移量 150000 对应的逻辑块号为 292, 块内偏移量为 496。由于 $10 + 170 < 292 < 10 + 170 + 170 \times 170$, 而 $292 - (10 + 170) = 112$, $112 / 170$ 得到商为 0, 余数为 112, 故可从 FCB 的第 11 个地址项, 即二次间址项中得到二次间址块的地址, 并从二次间址块的第 0 项中获得一个一次间址块的地址, 再从该一次间址块的第 112 项中获得对应的物理盘块号, 块内偏移量为 496。

(3). 由于文件的 FCB 已在内存, 为了访问文件中某个位置的内容, 可通过直接地址直接读取文件盘块, 最少需要 1 次访问磁盘。要使得访问次数最多, 则文件控制块放在三级索引处, 第一次是读三次间址块, 第二次是读二次间地址, 第三次是读一次间地址读到文件控制块, 第四次是读文件盘块, 最多需要 4 次访问磁盘。

3. (原书 第 6 题) 有一计算机系统采用如下图 5.7 所示的位示图(行号、列号都从 0 开始编号)来管理空闲盘块。如果盘块从 0 开始编号, 每个盘块的大小为 1KB。

(1). 现要为文件分配两个盘块, 试具体说明分配过程。

(2). 若要释放磁盘的第 300 块, 应如何处理?

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5																
6																

图 5.7

【解析】

(1). 查位示图，找到两个为 0 的位，第 3 字第 11 位和第 4 字第 2 位；计算出块号 $3*16+11+1=60$ 、 $4*16+2+1=67$ ，然后将 60, 67 分配给文件。

(2). 计算 100 块位示图对应位置，该块的字号是 $[(100-1)/16]=6$ ，位是 $[(100-1)\%16]=3$ ，然后将第 6 字第 3 位置 0。

考点 6 磁盘存储器管理

温馨提示：本考点考查磁盘存储器的管理方式，请同学们注意磁盘调度算法。希望同学们了解磁盘的结构和磁盘调度算法（先来先服务、最短寻道时间优先、电梯算法和循环扫描算法）。

一. 选择题部分

1. (原书 第 2 题)从磁盘读取数据的下列时间中，() 对系统效率的影响最大。
- A. 处理时间 B. 传输时间 C. 延迟时间 D. 寻道时间

【解析】 本题考查影响磁盘读取数据的时间因素。磁盘访问时间由以下三部分组成：

(1). **寻道时间 T_s** ：是指把磁头移动到指定磁道上所经历底时间，是启动磁臂的时间 S 与磁头移动 N 条磁道所花费的时间之和。

(2). **旋转延迟时间 T_r** ：是指指定扇区移动到磁头下面所经历的时间，平均 T_r 为 50—

100ms。

(3). **传输时间 T_t** : 是指向磁盘写入数据所经历的时间, 与每次读/写的字节数 b 和旋转速度有关。

因此总的访问时间 T_a 为上面三项之和:

$$T_a = T_s + 1/2r + b/rN$$

寻道时间和旋转延迟时间占据了访问时间中的大部分, 而传输所占的时间比重是比较小的。

参考答案: D

2. (原书 第 4 题) 会出现饥饿现象的磁盘调度算法是 ()。

- A. FCFS B. SSTF C. SCAN D. CSCAN

【解析】本题考查 SSTF 算法的特点。最短寻道时间优先算法选择这样的进程, 其要求访问的磁道与当前磁头所在的磁道距离最短, 使每次的寻道时间最短, 但这种调度算法不能保证平均寻道时间最短。

只要有新进程的请求到达, 其所要访问的磁道与当前所在磁道的距离较近, 其 I/O 请求必优先满足, 故而可能导致进程发生“饥饿”的现象。

参考答案: B

3. (原书 第 5 题) 对磁盘进行移臂调度的目的是为了缩短 () 时间。

- A. 延迟 B. 寻道 C. 传送 D. 启动

【解析】磁盘移臂, 缩短的是寻道时间。提高磁盘转速, 则可减少旋转延迟。在道密度不变的情况下, 单位时间内能传送的数据也会增多, 使得读数量相等的数据所需传送时间变短。

【补充】磁盘信息记录在磁道上, 多个盘片, 正反两面都用来记录信息, 每面一个磁头, 所有盘面中处于同一磁道号上的所有磁道组成一个柱面。磁道由若干个扇区组成, 每个扇区的大小相当于一个盘块。

物理地址形式为: 磁头号 (盘面号)、磁道号 (柱面号) 和扇区号, 如图 5.8 所示。

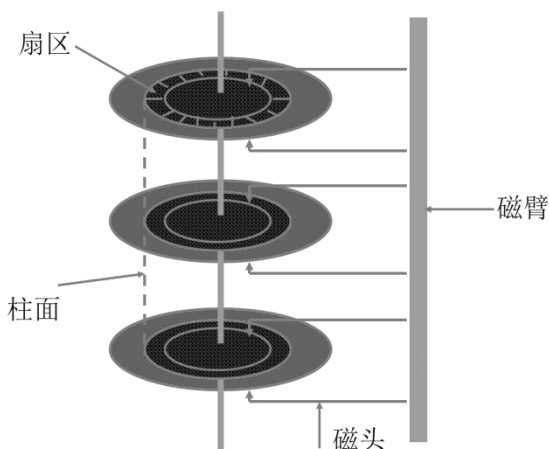


图 5.8 磁盘的物理结构

参考答案: B

4. (原书第10题) 磁盘存取时间包括寻道的时间、定位扇区的时间以及读写数据的时间, 若磁盘的转速提高一倍, 则 ()。
- A. 平均存取时间减少 B. 平均寻道时间减少
C. 存储道密度增加一倍 D. 平均寻道时间增加

【解析】由第五题的补充部分的图可知，磁盘转速提高一倍，并不会改变磁盘的寻道时间，故而 B、D 答案错误。注意区分磁盘的存储密度、道密度和位密度：

- (1). **磁盘的存储密度**: 磁盘单位面积上所能存储的二进制信息量。
- (2). **磁盘道密度**: 沿磁盘半径方向单位长度上的磁道数, 单位为道 / 英寸。
- (3). **磁盘位密度**: 磁道单位长度上记录的二进制代码的位数, 单位是位 / 英寸。

由道密度的定义可知，磁盘转速与存储道密度也没有必然的关系。文件长度不变的情况下，磁盘转速提高一倍，读取文件对应的存储区所需要的时间会减少，磁盘（访问）存取时间由寻道时间、旋转延迟时间和传输时间组成。在旋转延迟时间变短的情况下，会减少传输时间，也会使得平均存取时间变短。

参考答案: A

二. 综合应用题部分

1. (原书第3题)假设一个可移动磁头的磁盘具有 200 个磁道,其编号为 0~199,当前它刚刚结束了 125 道的存取,正在处理 149 道的服务请求,假设系统当前 I/O 请求序列为:

88, 147, 95, 177, 94, 150, 102, 175, 138

试问对以下的磁盘 I/O 调度算法而言, 满足以上请求序列, 磁头将如何移动? 并计算总的磁道移动数。

- (1). 先来先服务算法 (FCFS) ;
- (2). 扫描法 (SCAN) (假设沿磁头移动方向不再有访问请求时, 磁头沿相反方向移动)。

【解析】

- (1). FCFS 算法下磁头的移动轨迹如表 5.2 所示。

表 5.2 FCFS 下磁头的移动情况

当前 149	下一磁道	88	147	95	177	94	150	102	175	138
	移动距离	61	59	52	82	83	56	48	73	37

由表 5.2 可知, FCFS 算法下总的磁道移动数为: $61+59+52+82+83+56+48+73+37=551$ 。

- (2). 我们常说的 SCAN 算法是扫描到最内和最外的磁道的, 哪怕这些磁道没有 I/O 请求。但是本题要求只扫描到有服务请求的最内端和最外端的磁道, 所以我们可以得到 SCAN 算法下磁头的移动轨迹如表 5.3 所示。

表 5.3 SCAN 算法下磁头的移动情况

当前 149	下一磁道	150	175	177	147	138	102	95	94	88
	移动距离	1	25	2	30	9	36	7	1	6

总的磁道移动数为: $1+25+2+30+9+36+7+1+6=117$ 。

2. (原书 第 7 题) 【2010 年统考】假设计算机系统采用 CSCAN (循环扫描) 磁盘调度策略, 使用 2KB 的内存空间记录 16384 个磁盘块的空闲状态。

- (1). 请说明在上述条件下如何进行磁盘块空闲状态管理。

- (2). 设某单面磁盘旋转速度为每分钟 6000 转。每个磁道有 100 个扇区, 相邻磁道间的平均移动时间为 1ms。若在某时刻, 磁头位于 100 号磁道处, 并沿着磁道号大的方向移动 (如下图所示), 磁道号请求队列为 50、90、30、120, 对请求队列中的每个磁道需读取 1 个随机分布的扇区, 则读完这 4 个扇区点共需要多少时间? 要求给出计算过程。

- (3). 如果将磁盘替换为随机访问的 Flash 半导体存储器 (如 U 盘、SSD 等), 是否有比 CSCAN 更有效的磁盘调度策略? 若有, 给出磁盘调度策略的名称并说明理由; 若无, 说明理由。

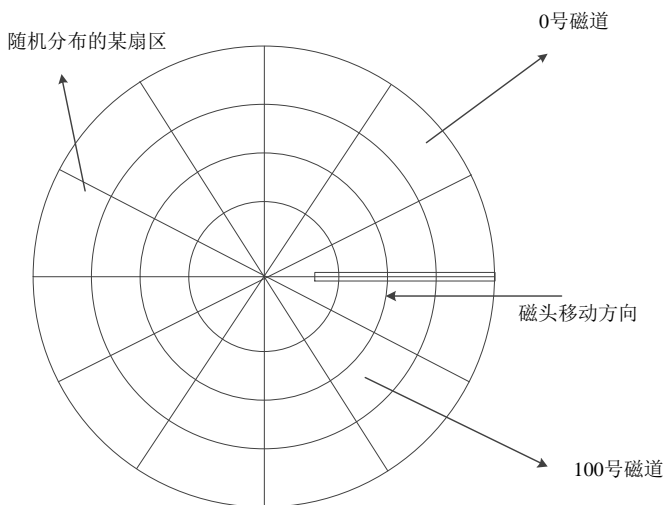


图 5.9

【解析】这个图真不好画，画到快画哭了。

(1). 依题意，可采用位示图法表示磁盘块的空闲状态，每个磁盘块在位示图中用一位 2 进制位表示，用 0 表示磁盘块空闲，用 1 表示磁盘块已分配。16384 个磁盘块共占用 $16384\text{bit}=16384/8\text{B}=2048\text{B}=2\text{KB}$ 。刚好使用 2KB 的内存来记录这些磁盘块的空闲状态，故而恰好能够表述。

(2). 采用 CSCAN 调度算法，磁道的访问次序为 $120 \rightarrow 30 \rightarrow 50 \rightarrow 90$ ，如下图 5.10 所示：

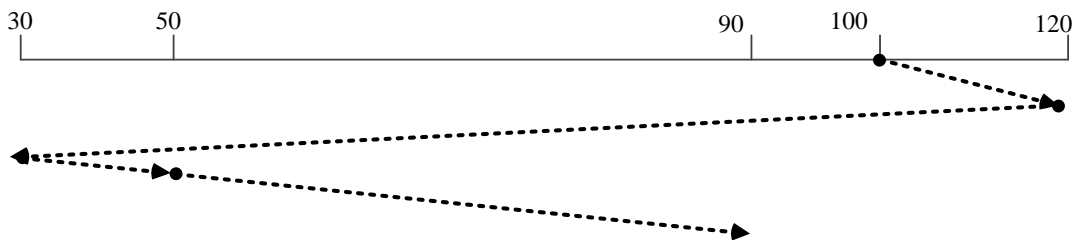


图 5.10 CSCAN 磁盘调度算法磁头移动轨迹

由上图可知，磁头总共移动的磁道总数 = $(120-100) + (120-30) + (50-30) + (90-50) = 20+90+20+40=170$ ，可知总寻道时间 = $170 \times 1\text{ms} = 170\text{ms}$ 。

由题意可知，磁盘的旋转速度为每分钟 6000 转，故而每转需要 $1/6000$ 分钟，即 10ms 。因为请求队列中的每个磁道需读取一个随机分布的扇区，可知平均旋转延迟为磁道旋转半周所需要的时间，故而平均旋转延迟时间为 5ms ，总的旋转延迟时间为 $5\text{ms} \times 4 = 20\text{ms}$ 。

因为读取一个磁道需要 10ms，每个磁道有 100 个扇区，则读取一个扇区需要 0.1ms，总的读取扇区时间为 $0.1\text{ms} \times 4 = 0.4\text{ms}$ ；

综上，磁盘访问总时间为 $170\text{ms} + 20\text{ms} + 0.4\text{ms} = 190.4\text{ms}$ 。

(3). 采用 FCFS（先来先服务）调度策略更高效。因为 Flash 半导体存储器的物理结构不需要考虑寻道时间和旋转延迟时间，可直接按 I/O 请求的先后顺序服务。

本章到此就结束了，你觉得本章有什么地方不明白吗？任何问题，欢迎您与我们作者进行交流！



shareOurDreams
梦享团队微信号



weCSdream
梦享团队官方微信公众号



梦享论坛团队
梦享团队新浪微博

第 6 章 输入输出 (I/O) 管理

考点 1 设备管理的基本概念

温馨提示：本考点主要考查设备管理的基本概念。

一. 选择题部分

1. (原书 第 1 题) 设备管理的主要程序之一是设备分配程序, 当进程请求在主存和外设之间传送信息时, 设备分配程序分配设备的过程通常是 ()。
- A. 先分配设备, 再分配控制器, 最后分配通道
B. 先分配控制器, 再分配设备, 最后分配通道
C. 先分配通道, 再分配设备, 最后分配控制器
D. 先分配通道, 再分配控制器, 最后分配设备

【解析】本题考查设备分配程序分配设备的过程。独占设备的分配程序需要分配以下 3 个部分:

(1). 分配设备: 根据 I/O 请求中的物理设备名, 查找系统设备表 SDT, 找出设备的 DCT, 查看设备状态字。

(2). 分配控制器: 系统把设备分配给请求的 I/O 进程后, 从 DCT 中找到与该设备连接的控制器的 COCT, 从状态控制字中读取控制器是否忙碌。

(3). 分配通道: 从该 COCT 中可找到与该控制器连接的通道的 CHCT, 再根据 CHCT 中的状态信息可知道通道是否忙碌。

只有设备、控制器和通道三者都分配成功, 此次的设备分配成功, 才能进行启动设备传送数据。

参考答案: A

2. (原书 第 4 题) 下列设备不属于系统设备的是 ()。
- A. 鼠标 B. 键盘 C. 扫描仪 D. 磁盘

【解析】本题考查设备的从属关系分类。计算机设备有以下分类方法：

(1). 基于设备的工作特性：

- 外部存储设备（长期保存信息，可随时访问，如磁盘、磁带）。
- 输入/输出设备（字符设备，以单个字符为单位存储、传输信息，如显示器、键盘、打印机等）。

(2). 基于设备的从属关系：

- 系统设备（一般是标准设备）（OS 生成时就配置在系统中的标准设备，如：键盘、鼠标、显示器、终端、磁盘等）。
- 用户设备（一般为非标准设备）（设备的处理程序由用户提供，需另外安装，如：扫描仪、A-D/D-A 转换设备等）。

(3). 基于设备的分配特性：

- 独享设备。使用具有排它性，如低速 I/O 设备。
- 共享设备。可由多个用户程序交替使用，如硬盘。
- 虚拟设备。将独占设备模拟为共享设备，即将慢速的独占设备经软件技术改造成为多个进程可以共享的设备，典型如 SPOOLing 技术。

(4). 基于信息组织和处理的方式：

- 字符设备（信息是以字符为单位来组织和分配的，系统中大部分设备均属此类，如打印机、键盘、显示器等。该类设备特点是速度慢，也称慢速设备）。
- 块设备（信息是以块为单位来组织和分配的，如磁盘、磁带等。该类设备特点是速度快，也称快速设备）。

参考答案： C

考点 2 I/O 控制方式

温馨提示：本考点主要考查 I/O 控制方式，包括程序直接控制、中断驱动方式、DMA 方式和通道方式。I/O 控制方式是本章的重要考点，请同学们务必掌握。

一. 选择题部分

1. （原书 第 4 题）（ ）用作连接大量的低速和中速 I/O 设备。

A. 选择通道

B. 字节多路通道

C. 数组多路通道

D. 以上都不是

【解析】本题考查字节多路通道方式的特点。**选择通道**又称高速通道，在物理上它可以连接多个设备，但是这些设备不能同时工作，在某一段时间内通道只能选择一个设备进行工作。选择通道主要用于连接高速外围设备，如磁盘、磁带等，信息以成组方式高速传输。

字节多路通道适用于连接大量像光电机等字符类中低速设备，这些设备传送一个字符（字节）的时间很短，字符（字节）间的等待时间很长。通道“数据宽度”为单字节，以字节交叉方式轮流为多台设备服务，使效率提高。字节多路通道可有多个子通道，同时执行多个通道程序。

数组多路通道适合于连接多台像磁盘等高速设备，这些设备的传送速率很高，但传送开始前的寻址辅助操作时间很长。通道“数据宽度”为定长块，多台设备以成组交叉方式工作，以充分利用并尽量重叠各台高速设备的辅助操作时间。传送完 K 个字节数据，就重新选择下个设备。数组多路通道可有多个子通道，同时执行多个通道程序。

参考答案: B

2. (原书第9题) I/O 系统硬件结构分为 4 级: ①设备控制器; ②I/O 设备; ③计算机; ④I/O 通道。按级别由高到低的顺序是 ()。

A. ②-④-①-③

B. (3)-(1)-(4)-(2)

C. ②-①-④-③

D. (3)-(4)-(1)-(2)

【解析】本题考查 I/O 系统硬件层次结构。具有通道的 I/O 系统结构如图 6.1 所示。I/O 系统硬件结构可分为 4 级：最低级为 I/O 设备，次低级为设备控制器，次高级为 I/O 通道，最高级为计算机（即主机）。

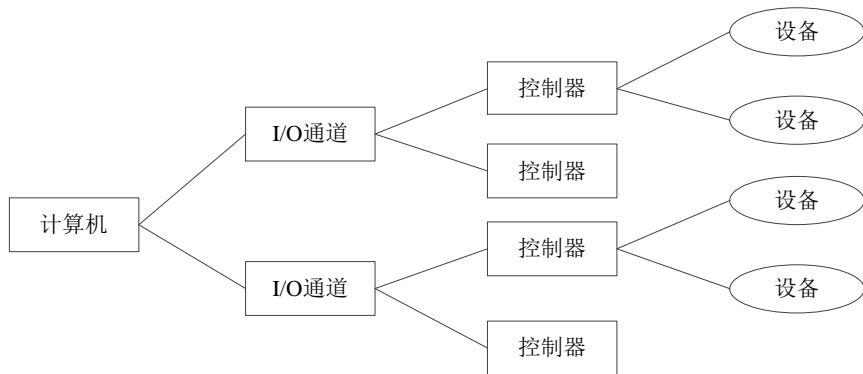


图 6.1 具有通道的 I/O 系统结构

参考答案：D

3. (原书 第 11 题) I/O 软件一般分为 4 个层次：用户层、与设备无关软件层、设备驱动程序、中断处理程序。以下工作中，不是由设备驱动程序完成的是（ ）。
- A. 向设备寄存器写命令
 - B. 检查用户是否有权使用设备
 - C. 将二进制整数转换成 ASCII 码以便打印
 - D. 解释用户的 I/O 请求，并将该请求转化为具体的 I/O 操作

【解析】本题考查 I/O 软件层次结构和设备驱动程序的工作。设备驱动程序的主要功能如下：

- (1). 接收由 I/O 进程发来的命令和参数，并将命令中的抽象要求转换为具体要求。例如，将磁盘块号转换为磁盘的盘面、磁道号及扇区号。
- (2). 检查用户 I/O 请求的合法性，了解 I/O 设备的状态，传递有关参数，设置设备的工作方式。
- (3). 向设备寄存器发出 I/O 命令，如果设备空闲，便立即启动 I/O 设备去完成指定的 I/O 操作；如果设备处于忙碌状态，则将请求者的请求块挂在设备队列上等待。
- (4). 及时响应由控制器或通道发来的中断请求，并根据其中断类型调用相应的中断处理程序进行处理。
- (5). 对于设置有通道的计算机系统，驱动程序还应能够根据用户的 I/O 请求，自动地构成通道程序。

将二进制整数转换成 ASCII 码打印不是设备驱动程序完成的工作。

参考答案：C

4. (原书 第 15 题) 【2012 年统考真题】操作系统的 I/O 子系统通常由四个层次组成，每一层明确定义了与邻近层次的接口。其合理的层次组织排列顺序是（ ）。
- A. 用户级 I/O 软件、设备无关软件、设备驱动程序、中断处理程序
 - B. 用户级 I/O 软件、设备无关软件、中断处理程序、设备驱动程序
 - C. 用户级 I/O 软件、设备驱动程序、设备无关软件、中断处理程序
 - D. 用户级 I/O 软件、中断处理程序、设备无关软件、设备驱动程序

【解析】本题可用第 14 题的答案解之，选择 A 答案。事实上，I/O 系统的逻辑结构图如图 6.2 所示。

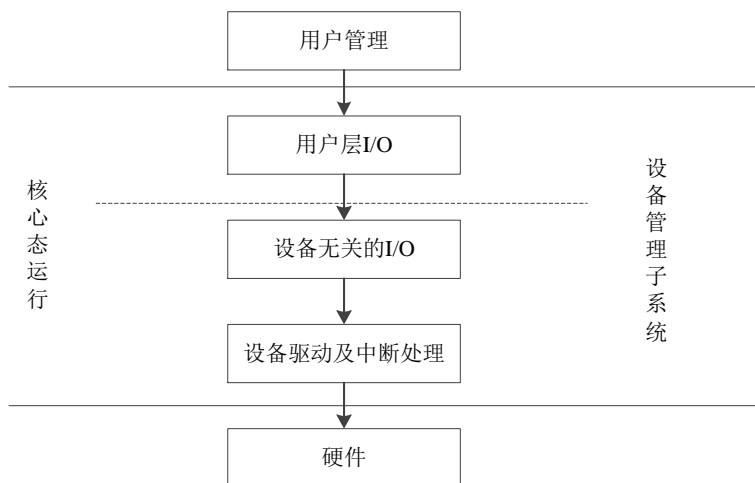


图 6.2 I/O 系统的逻辑结构图

可将 I/O 系统分层为：

(1). **用户层 I/O**。实现与用户交互的接口，用户可直接调用在用户层提供的、与 I/O 操作相关的库函数，对设备进行操作。

(2). **设备无关的 I/O（设备独立性软件）**。该层的基本功能是向上层提供统一的系统调用接口，对下通过设备驱动程序接口调用设备驱动程序。该层的主要任务有两个：一是设备命名及与设备驱动程序的映射，二是对系统中各类设备采用预先设计的、统一的逻辑名称进行命名，所有软件都以逻辑名称访问设备。

(3). **设备驱动程序**。设备驱动程序包含了所有与设备相关的代码，其功能是从与设备无关的软件中接收 I/O 请求并执行。设备驱动程序直接与硬件相关，负责具体实现系统对设备发出的操作指令，驱动 I/O 设备工作。

(4). **中断处理程序**。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，处理完成后恢复被中断进程的现场并返回被中断进程。中断处理层主要完成以下工作：进行进程上下文切换；对处理中断信号源进行测试，读取设备状态和修改进行状态等。

参考答案： A

二. 综合应用题部分

1. (原书 第 4 题) I/O 控制方式有几种？各有什么特点？

【解析】 I/O 控制方式的发展经历了四个阶段：程序查询方式、I/O 中断方式、DMA

方式和 I/O 通道方式。

(1). **程序查询方式**。在早期计算机或现代一些简单的微型计算机系统中, 采用程序查询 I/O 方式。程序查询是一种用程序直接控制 I/O 操作的方式。

CPU 与外设的活动本质上是异步的, 为了实现 CPU 与外设间的信息传送, CPU 必须重复测试外设的状态, 仅当外设是处在准备好的状态时, CPU 才能与外设交换信息。所以, 在程序查询 I/O 方式的接口电路中必须设置一个状态端口, 以使 CPU 通过执行输入指令了解外设的状态。

当采用程序查询传送方式时, 每当程序要使用某一外设进行 I/O 操作时, CPU 要执行一段循环测试程序, 以实现在外设准备好时执行一条输入/输出指令, 进行一个字节或字的数据传送操作。在这种方式下, CPU 的大量时间消耗在等待输入输出的循环检测上, 使 CPU 与外设串行工作, 严重影响了 CPU 和外设的使用效率, 致使整个系统效率很低。

(2). **I/O 中断方式**引入中断技术后, 每当设备完成 I/O 操作时, 便向 CPU 发出中断请求信号, 通知 CPU 外设已准备好, 可以进行数据传送操作。这样, CPU 一旦启动 I/O 设备后便可执行其它程序, 仅在收到 I/O 中断请求时才执行其中断服务程序, 进行 I/O 处理和 I/O 操作。程序中断传送方式改善了 CPU 的利用率, 并使 CPU 与外设并行操作。但 I/O 数据的处理和 I/O 操作的控制都是由 CPU 承担的, 仍然消耗了 CPU 不少时间。

(3). **直接存储器访问 (DMA)** 方式虽然 I/O 中断方式比程序查询方式更有效, 但须注意, 它仍是以字节或字为单位进行输入输出的, 每当完成一个字节或字时, 控制器便要向 CPU 请求一次中断。换言之, 采用 I/O 中断方式时的 CPU, 是以字节或字为单位进行干预的。如果将这种方式用于块设备的 I/O, 显然是低效的。例如, 为了从磁盘中读出 1KB 的数据块, 需要中断 CPU 1K 次。为了进一步减少 CPU 对 I/O 的干预而引入了直接存储器访问 (DMA) 方式。

(4). **I/O 通道方式**。I/O 通道方式时 DMA 方式的发展, 它会进一步较少对 CPU 的干预, 即把对一个数据块的读 (或写) 为单位的干预, 减少为对一组数据块的读 (或写) 即有关的管理和控制为单位的干预。I/O 通道有自己的指令系统, 即通道程序, 可以与 CPU 并行操作, 独立管理外设和实现主存和外设之间的信息传输, 使 CPU 摆脱了繁忙的 I/O 操作。在配置通道的计算机系统中, 不仅能实现 CPU 与通道的并行操作, 而是通道与通道、各通道的外设之间均能实现并行操作, 因而有效地提高了整个系统的使用效率。

考点3 缓冲管理

温馨提示：本考点考查缓冲区及其管理方式，包括单缓冲、双缓冲、缓冲池三种方式。

一. 选择题部分

1. (原书 第1题)缓冲技术用于()。

- A. 提高主机和设备交换信息的速度
- B. 提供主、辅存接口
- C. 提高设备利用率
- D. 扩充相对地址空间

【解析】本题考查缓冲技术的作用。随着计算机技术的发展，外设也在迅速发展，速度在不断提高，但它与 CPU 的速度仍相差甚远。虽然通道技术和中断技术为计算机系统的并行活动提供了强有力的支持，但往往由于通道数量不足而产生“瓶颈”现象，使得 CPU、通道和 I/O 设备之间的并行能力并未得到充分发挥。

引入缓冲技术的优点有：

- (1). 缓和 CPU 与 I/O 设备之间速度不匹配的矛盾，提高了主机与设备交换信息的速度。
- (2). 提高 CPU、通道与 I/O 设备间的并行性。

减少对 CPU 的中断次数，放宽 CPU 对中断响应时间的要求。

参考答案： A

2. (原书 第4题)为了使多个进程能有效地同时处理阵发性的输入和输出，最好使用()结构的缓冲技术。

- A. 多缓冲
- B. SPOOLing
- C. 单缓冲
- D. 双缓冲

【解析】多缓冲为相同类型的 I/O 设备设置两个公共缓冲队列，一个用于输入，另一个用于输出，如图 6.3 所示。当输入设备进行输入时，就取输入缓冲队列指针所指向的缓冲使用，设备用完后归还缓冲。指针则前行指向下一个，整个缓冲队列循环使用。当输出设备进行输出时，就取输出缓冲队列指针所指向的缓冲使用，用完后归还缓冲。

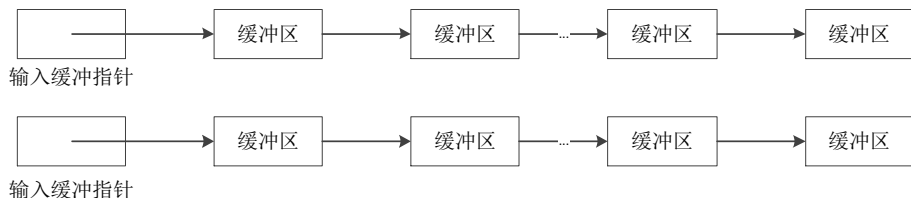


图 6.3 多缓冲下的公共缓冲队列

参考答案: A

3. (原书第6题)【2011年统考真题】某文件占10个磁盘块，现要把该文件磁盘块逐个读入主存缓冲区，并送用户区进行分析。假设一个缓冲区与一个磁盘块大小相同，把一个磁盘块读入缓冲区的时间为 $100\mu s$ ，将缓冲区的数据传送到用户区的时间是 $50\mu s$ ，CPU对一块数据进行分析的时间为 $50\mu s$ 。在单缓冲区和双缓冲区结构下，读入并分析该文件的时间分别是()。
- A. $1500\mu s$ 、 $1000\mu s$ B. $1550\mu s$ 、 $1100\mu s$
- C. $1550\mu s$ 、 $1550\mu s$ D. $2000\mu s$ 、 $2000\mu s$

【解析】单缓冲区即只有一个缓冲区的情况。在块设备输入时，假定从磁盘把一块数据输入到缓冲区的时间为 T ，操作系统将该缓冲区中的数据传送到用户区的时间为 M ，CPU 对这一块数据处理(计算)的时间为 C 。 C 与 T 是可以并行的（如图 6.4 所示），当 $T > C$ 时，系统对每一个数据块的处理时间为 $M+T$ ，反之则为 $M+C$ 。

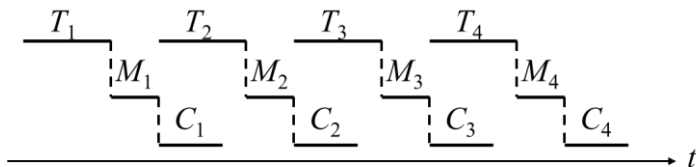


图 6.4

可把系统对每一个数据块的处理时间表示为 $\text{MAX}(C,T)+M$, 得单缓冲区情况下系统对 n 个数据块从输入到处理完成所需要的总时间满足如下计算公式:

$$T_{\text{total}} = T_1 + \sum_{i=1}^{n-1} \max(T_{i+1}, C_i) + \sum_{i=1}^n M_i + C_n$$

可计算得到本题的

$$T_{\text{total}} = 100us + 9 \times 100us + 10 \times 50us + 50us = 1550us$$

当然，也可采用简单的分析方法：当上一个磁盘块从缓冲区读入用户区完成时下一个磁盘块才能开始读入，故而最后一个磁盘块读入用户区完毕时所用时间为 $150\mu s \times 10 = 1500\mu s$ ，加上处理最后一个磁盘块的时间 $50\mu s$ ，得 $1550\mu s$ 。

双缓冲的原理如下图 5.15 所示。在双缓冲的情况下，在设备输入时，先将数据送入第一缓冲区，装满后便转向第二缓冲区。此时操作系统可以从第一缓冲区中移出数据，送给用户进程(见图 6.5)。接着，由 CPU 对数据进行计算。在双缓冲时，情况比较复杂，我们简单分析 C 与 T 的大小关系两种情况。

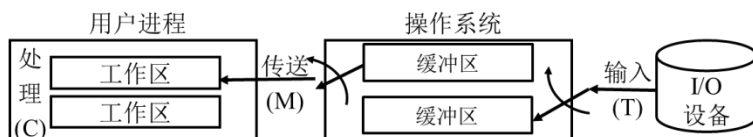


图 6.5 双缓冲

如果 $C < T$ ，可使块设备连续输入，如图 6.6 所示。

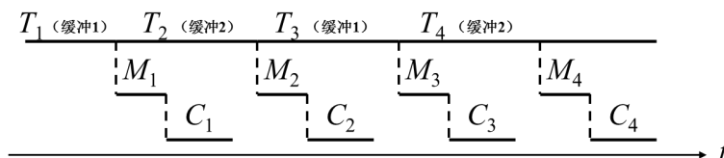
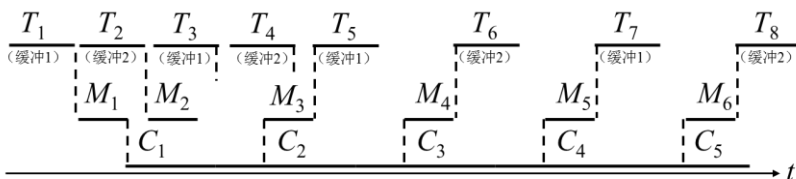


图 6.6

此时，系统处理数量为 n 的数据块所用的总时间为

$$T_{\text{total}} = \sum_{i=1}^n T_i + M_n + C_n$$

如果 $C > T$ ，则可使 CPU 不必等待设备输入。如图 6.7 所示。

图 6.7 双缓冲下 $C > T$ 时 T、M、C 的关系

此时，系统处理数量为 n 的数据块所用的总时间为

$$T_{\text{total}} = \sum_{i=1}^n C_i + M_1 + T_1$$

但是,若用户程序中采用“设备类相对号”的方式来指定自己所需使用的设备,那么系统可以根据设备的分配情况从指定的设备类中找出“好的且尚未分配的”设备来进行分配,并建立“设备类相对号”与“绝对号”的对应关系。这样,用户程序中使用由“设备类相对号”指定的逻辑设备,系统根据“绝对号”来启动实际的物理设备。

参考答案: A

2. (原书 第 5 题)下述各项中, () 不是 SPOOLing 技术的特点。

- A. 提高了 I/O 速度
- B. 将独占设备模拟成共享设备
- C. 采用高速缓存(cache)
- D. 实现了虚拟设备功能

【解析】本题考查 SPOOLing 技术的特点。SPOOLing 技术有以下特点:

(1). **提高了 I/O 速度**。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). **将独占设备改为共享设备**。设备并没有分配给任何进程,在输入井或输出井中,分配给进程的是一个存储区和建立一张 I/O 请求表。这样,便把独占设备改为共享设备。

(3). **实现了虚拟设备功能**。宏观上,多个进程同时使用一台独享设备,而对每一进程而言,都认为自己独占这一设备。当然,该设备只是逻辑上的设备。Spooling 系统实现了将独占设备变换为若干台对应的逻辑设备的功能。

采用高速缓存技术不具备 SPOOLing 技术的特点。

参考答案: C

3. (原书 第 7 题)操作系统 () 采用了以空间换时间的技术。

- A. SPOOLing 技术
- B. 覆盖技术
- C. 通道技术
- D. 虚拟存储技术

【解析】SPOOLing 技术是一种将独占设备改造成共享设备的技术,该技术的引入,是为了满足多进程对独占设备的共享使用。SPOOLing 系统主要包括以下三部分:

(1). **输入井和输出井**。输入井和输出井是在磁盘上开辟出来的两个存储区域。输入井模拟脱机输入时的磁盘,用于收容 I/O 设备输入的数据。输出井模拟脱机输出时的磁盘,用于收容用户程序的输出数据。

(2). **输入缓冲区和输出缓冲区**。输入缓冲区和输出缓冲区是在内存中开辟的两个缓冲区。输入缓冲区用于暂存由输入设备送来的数据,以后再传送到输出井。输出缓冲区用于暂存从输出井送来的数据,以后再传送到输出设备。

(3). **输入进程和输出进程**。输入进程模拟脱机输入时的外围控制机,将用户要求的数

据由输入设备到输入缓冲区，再送到输入井。当 CPU 需要输入设备时，直接从输入井读入内存。输出进程模拟脱机输出时的外围控制机，把用户要求输出的数据，先从内存送到输出井，待输出设备空闲时，再将输出井中的数据，经过输出缓冲区送到输出设备上。

SPOOLing 技术提高了设备的利用率，缩短了进程的周转时间，是一种以空间换取时间的技术。

参考答案： A

4. (原书 第 13 题)【2009 年全国统考】程序员利用系统调用打开 I/O 设备时，通常使用的设备标识是 ()。

- A. 逻辑设备名
- B. 物理设备名
- C. 主设备名
- D. 从设备名

【解析】若系统直接利用物理设备名来给用户程序分配设备，设备利用率低，也不方便用户使用。故而，引入了设备独立性这个概念。**设备独立性**是指，程序员（用户）在编制程序时所使用的设备与实际使用的设备无关。也即，在用户程序中使用逻辑设备名，而不直接使用物理设备名，执行时由系统将逻辑设备名转换为物理设备名。

如何把逻辑设备名转换成物理设备名？具体而言，系统可以根据设备的使用情况，动态地分配给程序某类设备中的任一台物理设备，程序都能正确地执行。

【注意】逻辑设备名，是用户自己指定的设备名（或者设备号），是暂时的，可更改的。而物理设备名，是系统提供的设备标准名称，是永久的，不可更改的。

参考答案： A

本章到此就结束了，你觉得本章有什么地方不明白吗？任何问题，欢迎您与我们作者进行交流！



shareOurDreams
梦享团队微信号



weCSdream
梦享团队官方微信公众号



梦享论坛团队
梦享团队新浪微博

“梦享考研系列”辅导书——答疑解惑

尊敬的读者朋友！

您好！首先，欢迎您使用我梦享团队编写的“梦享考研系列”辅导书，我们衷心地感谢您的支持！

让每一个考研人圆梦，是我们团队的梦想；默默陪伴着您走过每一个寻梦的日日夜夜，做您不离不弃的朋友，是我们的愿望。

宝剑锋从磨砺出，梅花香自苦寒来。每一个甜蜜的果实，都在挥洒汗水之后才能摘取到。对于每一个考生来说，一本绝佳的考研辅导书，就像一把利剑，拥有这把利剑，我们能够百战百胜，所向披靡。我们深知我们团队的能力有限，但我们一直希望，我们这套书，就是大家手中的那一把利剑！为此，我们团队特意向各位考生征求好的教材写法，好的题目解析方法等，希望把大家的精华汇聚到一本书里，为更多的考生提供更好的辅导书。

除了“梦享考研系列”辅导书之外，梦享团队致力于给大家更好的考研服务。为了给广大读者提供一个答疑和交流的平台，梦享团队开通了微信号、微信公众号和新浪微博账号等三种平台。欢迎大家来跟我们交流，一起成长。



shareOurDreams
梦享团队微信号



weCSdream
梦享团队官方微信公众号



梦享论坛团队
梦享团队新浪微博

因为有你，所以有梦享！

梦享团队祝愿每一个考研人梦想成真！

参考文献

- [1] 汤子瀛. 计算机操作系统[M]. 西安: 西安电子科技大学出版社, 2001.
- [2] Tanenbaum A.S. 现代操作系统[M]. 北京: 机械工业出版社, 2009.
- [3] 翔高教育. 计算机学科专业基础综合习题精编[M]. 上海: 复旦大学出版社, 2010.
- [4] 崔巍, 等. 计算机学科专业基础综合辅导讲义[M]. 北京: 原子能出版社, 2011.
- [5] 严蔚敏. 数据结构 (C 语言版) [M]. 北京: 清华大学出版社, 2007.
- [6] 严蔚敏, 等. 数据结构题集 (C 语言版) [M]. 北京: 清华大学出版社, 1999.
- [7] 谢希仁. 计算机网络 (第 5 版) [M]. 北京: 电子工业出版社, 2008.
- [8] 谢希仁. 计算机网络释疑与习题解答[M]. 北京: 电子工业出版社, 2011.
- [9] 唐朔飞. 计算机组成原理 (第 2 版) [M]. 北京: 高等教育出版社, 2008.
- [10] 唐朔飞. 计算机组成原理学习指导与习题解答 (第 2 版) [M]. 北京: 高等教育出版社, 2012.