

操作系统习题集

参考教材：

汤小丹等编着，计算机操作系统（第三版），西安电子科技大学出版社，2007 年版；

何炎祥等编着，计算机操作系统，清华大学出版社，2005 年版；

邹恒明着，计算机的心智操作系统之哲学原理，机械工业出版社，2009 年 4 月。

第一章 操作系统引论

1.1 选择题

1. 下列哪一条是在操作系统设计中引入多道程序技术的好处？ A
A. 使并发执行成为可能 B. 简化操作系统的实现
C. 减少对内存容量的需求 D. 便于实施存储保护
2. Windows XP 属于下列哪一类操作系统？ B
A. 单用户单任务 B. 单用户多任务 C. 多用户 D. 批处理
3. 下列哪一条不是批处理系统的优点？ D
A. 吞吐量大 B. 资源利用率高 C. 系统开销小 D. 响应及时
4. 能及时处理由过程控制反馈的数据并作出响应的操作系统是（ C ）
A、分时系统 B、网络系统 C、实时系统 D、批处理系统
5. UNIX 系统是一个__C__操作系统。
A、单用户 B、单用户多任务 C、多用户多任务 D、多用户单任务
6. 在分时系统中，当用户数一定时，影响响应时间的主要因素是__B__。
A、时间片 B、调度算法 C、存储分配方式 D、作业的大小
7. Windows NT 属于哪一类操作系统？ B
A、单用户单任务 B、单用户多任务 C、单道批处理 D、多用户
8. 多道程序设计技术是指：多道程序可同时进入 A，在 A 的位置 B，为使多道进程并发执行必须为每

- 个用户作业创建进程，批处理系统由 C 创建，而分时系统由 D 创建。 _____

选择答案：

(1) 内存 (2) 系统 (3) 固定 (4) 不定 (5) 进程调度 (6) 中级调度

(7) 作业调度 (8) 输入进程 (9) 系统调用 (10) 命令解释进程

A B C D

答

1 4 7 10

案

1.2 填空题

1. 在手工操作阶段，操作员在进行装卸卡片或磁带等手工操作时，CPU 处于空闲等待，我们称这种现象
为 人机矛盾 。 _____
2. 多道批处理系统的特征为 并发、共享、虚拟和异步 。 _____
3. 批量处理系统的缺点为 周转时间长 ； 缺乏人工干预（人机交互） 。 _____

4. 多道批处理 系统的出现，标志着操作系统的形成。 _____

5. 操作系统的基本类型有 批处理操作系统、分时系统和实时系统 。

6. 分时系统的特征为 多路性、 独立性、 及时性、 交互性 四个基本特征。

7. 以多道程序设计为基础的现代操作系统具有并发性 、 共享性 、 虚拟性 、 异步性 。

8. 计算机系统按用户指定的步骤，为用户一次上机解题所完成的工作的总和称为 作业 。

9. 从资源管理的观点出发，可把操作系统分为 存储管理 、 设备管理、 文件管理 、 处理机管理 和 作 _____

业管理 五大部分。 _____

10. 单道批处理系统是在解决 人机矛盾 和 CPU 与 I/O 设备速度不匹配 的矛盾中发展起来的。 _____

1.3 判断题

1. 分时操作系统必然建立在多道程序技术的基础之上。错
2. 联机批处理解决了作业自动转接，减少了作业建立和手工操作时间。对
3. 交互性是批处理系统的一个特征。错
4. 解决了作业自动转接，减少了作业建立和手工操作时间。对
5. 过载保护是分时系统的一个特征。错
6. 多道程序的引入是为了提高 CPU 的利用率。对
7. 多道程序技术可将一台物理 CPU 虚拟为多台逻辑 CPU。对
8. 在分时系统中，时间片越小，一个作业的总运行时间越短。错

1.4 简答题

1. 研究操作系统的主要观点有那些？

答：（1）资源的观点：研究如何对计算机系统中的各种软、硬件资源进行管理；怎样使计算机系统协调一致

地、有效地为用户服务；如何既发挥计算机系统资源的使用效率、提高计算机系统的服务质量，又确保计算

机系统的安全可靠。

（2）用户观点：操作系统是一个黑盒子，配置了操作系统的计算机与原来真实的物理计算机迥然不同，因为

它提供了用户使用计算机的更方便手段，构造了一台虚拟机，采用的操作命令决定了虚拟机的功能。

（3）进程观点：从进程角度分析操作系统，则所有进程的活动就构成了操作系统的当前行为，在每一个瞬间

都有一棵进程家族树，它展示着操作系统行为主体的一个快照。

（4）模块分层观点：用模块分层观点讨论模块之间的关系或者说讨论如何形成操作系统的架构，如何安排连

结这些程序模块才能构造一个结构简单清晰、逻辑正确、便于分析和实现的操作系统。

2. 什么是操作系统？简述现代操作系统的特征。

答：操作系统是控制和管理计算机系统内各种硬件和软件资源、合理有效地组织计算机系统的工作，为用户

提供一个使用方便可扩展的工作环境，从而起到连接计算机和用户的接口作用。

现代操作系统具有如下特征：

○ 并发（并行）性：指能处理多个同时性活动的的能力。I/O 操作和计算重叠，内存中同时存放几道用户程序，

1

这些都是并发的例子。

○ 共享：指多个计算任务对资源的共同享用。并发活动可能要求共享资源和信息；多个用户共享一个程序的

2

同一个副本，而不是分别向每个用户提供一个副本可以避免重复开发，节省人力资源。

○ 不确定性：指操作系统必须能处理任何一种事件序列，以使各个用户的算题任务正确地完成。

3

3. 操作系统和用户程序之间的关系是什么？

答：操作系统通过虚拟及其界面给用户程序提供各种服务，用户程序在运行过程中不断使用操作系统提供的

服务来完成自己的任务。如用户程序在运行过程中需要读写磁盘，这时就要调用操作系统的服务来完成磁盘

读写操作。

另一方面，用户程序不可能先于操作系统启动之前启动，因此每次启动一个用户程序，都相当于操作系统将

控制转移给用户程序；而在用户程序执行完毕后，又将控制还回给操作系统。从这个角度看，操作系统是主

程序，用户程序是子程序，操作系统在其生命周期内不断地调用各种应用程序。

因此操作系统和各种应用程序可以看作是互相调用，从而形成一个非常复杂的动态关系。

4. 推动操作系统进化的原因有哪些？

答：推动操作系统进化的根本原因是：硬件成本的不断下降；计算机功能和复杂性的不断变化。成本降低意

味着同样的价格可以买到更为先进的计算机；而复杂性的提高自然需要操作系统能力的提高。

另外，操作系统和攻击者之间的博弈也是影响操作系统发展的一个重要因素。

5. 试简述操作系统的发展历史。

答：操作系统大致经历了如下几个重要阶段：

第一阶段：状态机操作系统（1940 年以前）。此时计算机尚处于萌芽状态，操作系统运行在英国人贝巴斯想

象的自动机上。驱动这一阶段操作系统的动力是个人英雄主义。

第二阶段：单一操作员、单一控制端操作系统（20 世纪 40 年代）。这种单一操作员、单一控制端操作系

统（SOSC，single operator，single console）的操作系统是刚出现计算机时人们能想到的最直观的控制计算

机的方式，以美国宾西法尼亚大学的 ENIAC 计算机为代表。操作系统提供一个标准命令供用户使用，满足用户基

本的人机交互需求。（串行执行）

第三阶段：批处理操作系统（20 世纪 50 年代）。由于人操作缓慢，致使 SOSC 的运行效率低下，即人机矛盾，

为了改变这种状况，出现了批处理操作系统，以 IBM 的 1401 和 7094 为代表。用户首先将自己的程序编在卡

片或纸带上，交给计算机管理员，管理员在收到一定数量的用户程序后，将卡片及纸带上的程序和数据通过

IBM 1401 机器读入，并写在磁带上。然后，计算机操作系统将这盘磁带加载到 IBM 7094 上，一个个地运行

用户程序，运行的结果写到另一个磁盘上。所有程序运行完毕后，将存有结果的磁盘连接到 IBM 1401 上输

出。因此批处理操作系统就是由批处理监视器和原来的操作系统库函数组成的。（串行执行）

第四阶段：多道批处理操作系统（20 世纪 60 年代）。批处理阶段不能很好地解决高速设备（CPU）等待低速

设备（I/O）的问题。此阶段的主要目标是让 CPU 和 I/O 重叠运行，以 IBM 的 OS 360 为代表。同一时间可以

运行多个程序（宏观上），但控制计算机的人还是一个，即用户将自己的程序交给计算机管理员，再由管理

员负责加载运行。（并发执行）

第五阶段：分时/实时操作系统（20 世纪 70 年代）。分时操作系统是为了解决人机交互问题而出现的，以 MULTICS 和 UNIX 为代表。用户重新回到了机器的前面，通过 RS232 与主机进行通信，管理自己的程序；主机给每个用户分配一定的时间片，轮流地为各个用户服务。

实时操作系统是为了解决对计算机相应时间有严格要求的临界系统或应用而产生的，以 VxWorks 和 EMC 的

DART 为代表。

第六阶段：现代操作系统（1980 年以后）。是工作站和个人计算机出现的结果，代表性的有 DOS，

Windows，

Unix 和 Linux 等。这时以信息安全、网络为主要特征。

6. 现代几个典型操作系统所属的类型？

答：操作系统是由于需要而产生的，它随着计算机技术本身及其计算机应用的日益发展而逐渐发展和不断完善

善。它的功能由弱到强，现已成为计算机系统的核心组成。经历了手工操作、早期批处理阶段、执行系统阶

段、多道程序系统阶段、分时系统、实时系统、通用操作系统。进入 80 年代，硬件技术飞速发展以及微

机的出现和发展，操作系统有了进一步发展，如单用户操作系统、网络操作系统、分布式操作系统及智能化

操作系统。

单用户、单任务的操作系统，以 DOS 操作系统为代表，继 CP/M 操作系统之后，还出现了 C-DOS、M-DOS、

TRS-DOS、S-DOS 和 MS-DOS 等磁盘操作系统。还包括 Windows 3.1/3.2/95/98 等版本。

多用户多道作业和分时系统，其典型代表有 UNIX、XENIX、OS/2 以及 Windows NT 及其后来版本的操作系

统。

1.5 综合题

1. 假设有一个支持多道程序设计的计算机系统，其中每个作业都有完全相同的属性。对一个作业，在一段

计算周期 T 中，一半的时间用于 I/O，另一半时间用于处理器操作。每个作业总共运行 N 段计算周期。有

个定义如下:

吞吐量 (Throughput) = 在一时间段 T 中完成的平均作业数;

计算当有 1, 2 或 4 个作业并发执行时的周期、吞吐量和处理器使用率, 假设时间段 T 按一下任一种方式分

(1) I/O 在前半段, 处理器运行在后半段;

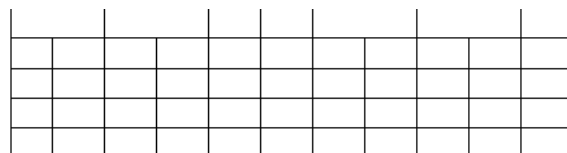
答：(1) I/O 在前半段，处理器运行于后半段

完成时刻 运行时间 利用率

12 N



B I/O CPU I/O CPU ... CPU I/O CPU I/O CPU N+1/2 N/2 N+1/2

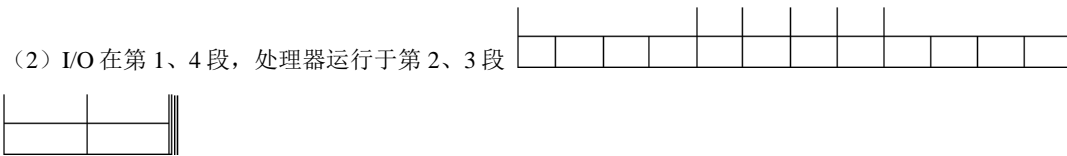


 $2N$

C I/O CPU ... I/O CPU $2N$ $N/2$

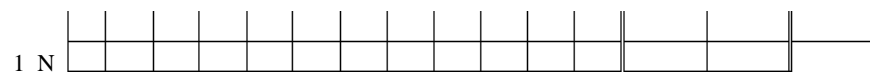
$2N+1/2$

D I/O CPU ... CPU I/O CPU $2N+1/2$ $N/2$



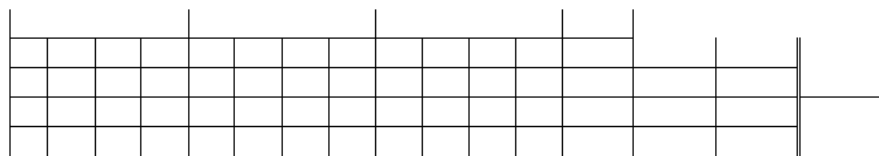
1 N

A I/O CPU CPU I/O I/O CPU ... I/O I/O CPU CPU I/O N N 50%



A I/O CPU CPU I/O I/O CPU CPU I/O I/O CPU CPU I/O N N/2 N

B I/O CPU CPU I/O I/O CPU CPU I/O I/O CPU CPU I/O $N+1/2$ $N/2$ $N+1/2$



A 1 2 3

B I/O CPU CPU I/O I/O CPU CPU ... $2N-3/4$ $N/2$

$2N$

C I/O CPU CPU I/O I/O CPU ... $2N-1/2$ $N/2$

D I/O CPU CPU I/O I/O ... $2N$ $N/2$

$2N+1/2$

I/O CPU CPU I/O I/O ... $2N+1/2$ $N/2$

当同时运行 2 个作业时，系统吞吐量和 CPU 利用率显著增加，表明系统被充分利用。但当作业增加到 4 个

时，吞吐量和 CPU 使用率变化不大，但平均周期却增加一倍，表明系统负荷过重，作业处理时间明显增长。

2. 某计算机用 Cache、内存和磁盘来实现虚拟内存。如果某数据在 Cache 中，访问它需要 t (ns)；如果在

A

内存但不在 Cache 中，则需要 t (ns) 的时间将其装入 Cache 然后开始访问；如果不在内存中，需要 t (ns)

BC

$n-1m-1$

将其读入内存，然后用 t (ns) 读入 Cache。如果 Cache 命中率为 n ，内存命中率为 m ，则平均访问

B_____

nm

时间是多少？

答：根据题目中的数据，平均访问时间为：

$n-11m-111$

$\times t + \times \times (t+t) + \times \times (t+t+t)$

= _____

AABABC

nnmmnm

11

$t+t+t$

= _____

ABC

nmn

3. 操作系统的未来发展趋势是怎样的？

答：随着计算机的不断普及，操作系统的功能会变得越来越复杂。在这种趋势下，操作系统的发展面临着两个不同的方向选择：一是微内核，二是大而全的全方位发展。微内核操作系统虽然有不少人在研究，但在工

业界获得认可的并不多。对工业界来说，操作系统是向着多功能、全方位方向发展的。另外，随着人们对信

息安全重视程度的不断提高，如何构建可靠、可用和安全的操作系统将成为一个十分重要的课题。从 Unix 的

1400 行代码到 Windows XP 的 4000 万行代码，这种系统的爆炸性增长给系统的可靠、可用和安全性带来的安

全隐患，在短时期内是很难解决的。

综上所述，操作系统的发展趋势很难预测。

4. 操作系统的主要特征是什么？

答：操作系统的主要特征是并发性、共享性、虚拟性和不确定性。

○ 并发性：并发性是指两个或多个事件在同一时间间隔内发生。在多道程序环境下，并发性是指在一段

1

时间内，宏观上有多个程序在同时运行，但在单处理机系统中，每一时刻却仅能有一道程序执行，故微观上

这些程序只能是分时地交替执行。倘若在计算机系统中有多个处理机，则这些可以并发执行的程序便可被分

配到多个处理机上，实现并行执行，即利用每个处理机来处理一个可并发执行的程序，这样，多个程序便可

同时执行。两个或多个事件在同一时刻发生称为并行。在操作系统中存在着许多并发或并行的活动。

○ 共享性：共享是指系统中的资源可供内存中多个并发执行的程序共同使用。由于资源属性的不同，对

2

资源共享的方式也不同，目前主要有以下两种资源共享方式互斥共享方式和同时访问方式。并发和共享是操

作系统的两个最基本的特征，它们又互为对方存在的条件。一方面，资源共享是以程序的并发执行为条件

的，若系统不允许程序并发执行，自然不存在资源共享问题；另一方面，若系统不能对资源共享实施有效管理

理，协调好多个程序对共享资源的访问，也必然影响到程序并发执行的程度，甚至根本无法并发执行。

○ 虚拟性：是指将一个物理实体映射为若干个逻辑实体。前者是客观存在的，后者是虚构的，是一种感

3

觉性的存在，即主观上的一种想象。

○ 不确定性：在多道程序环境下，允许多个程序并发执行，但只有程序在获得所需的资源后方能执行。

4

在单处理机环境下，由于系统中只有一个处理机，因而每次只允许一个程序执行，其余程序只能等待。内存

中的每个程序在何时能获得处理机运行，何时又因提出某种资源请求而暂停，以及程序以怎样的速度向前推

进，每道程序总共需多少时间才能完成，等等，都是不可预知的。因此，在操作系统中，存在着不确定性。

4. 简述 Windows 系列操作系统的发展历史。

答：Windows 系列操作系统是由微软公司从 1985 年起开发的一系列视窗操作系统产品，包括个人（家用）、商用和嵌入式 3 条产品线（图 1.4）。个人操作系统包括 Windows Me、Windows 95/98，及更早期的版本 Windows 1.x、2.x、3.x 等，主要在 IBM 个人机系列上运行。商用操作系统是 Windows 2000 和其前身版

本 Windows NT，主要在服务器、工作站等上运行，也可以在 IBM 个人系列机上运行。嵌入式操作系统有 Windows CE 和手机用操作系统 stinger 等。Windows XP 将使家用和商用两条产品线合二为一。截止至 20 世

纪末，全世界运行各种 Windows 版本的计算机有两亿台左右。微软公司从 1983 年开始研制 Windows 操作系

统。当时，IBM PC 进入市场已有两年，微软公司开发的磁盘操作系统 DOS 和编程语言 BASIC 随 IBM PC 捆

绑销售，取得了很大的成功。Windows 操作系统最初的研制目标是在 DOS 的基础上提供一个多任务的图形用

户界面。不过，第一个取得成功的图形用户界面系统并不是 Windows，而是 Windows 的模仿对象——苹果公

司于 1984 年推出的 Mac OS（运行于苹果公司的 Macintosh 个人计算机上），Macintosh 机及其上的操作系

统当时已风靡美国多年，是 IBM PC 和 DOS 操作系统在当时市场上的主要竞争对手。当年苹果公司曾对 PC 机

和 Windows 操作系统不屑一顾，并大力抨击微软公司抄袭 Mac OS 的外观和灵感。但苹果机和 Mac OS 是封

闭式体系（硬件接口不公开、系统源代码不公开等），而 IBM PC 和 MS-DOS 是开放式体系（硬件接口公开、允许并支持第三方厂家做兼容机、公开操作系统源代码等）。这个关键的区别使得 IBM PC 后来者居

上，销量超过了苹果机，并使得在 IBM PC 上运行的 Windows 操作系统的普及率超过了 Mac OS，成为个人计

算机市场占主导地位的系统。

5. 简述 UNIX 系列操作系统的发展历史。

答：“UNIX”这个名字是取“Multics”的反义，其诞生背景与特点一如其名。Multics 项目

（MULTiplexed Information and Computing Service）由贝尔（电话）实验室（Bell（Telephone）

Laboratories,

简称 BTL)、通用电气公司 (General Electric) 和麻省理工学院联合开发, 旨在建立一个能够同时支持数千

个用户的分时系统, 该项目因目标过于庞大而失败, 于 1969 年撤销。退出 Multics 项目后, 1969 年中期, 贝

尔实验室的雇员 Thompson 开始在公司的一台闲置的只有 4KB 内存的 PDP-7 计算机上开发一个“太空漫游”

游戏程序。由于 PDP-7 缺少程序开发环境, 为了方便这个游戏程序的开发, Thompson 和公司的另一名雇员

Ritchie 一起用 GE-645 汇编语言 (以前曾用于 Multics 开发) 开发 PDP-7 上的操作环境。最初是一个简单的

文件系统, 很快又添加了一个进程子系统、一个命令解释器和一些实用工具程序。他们将这个系统命名为 UNIX。此后, 随着贝尔实验室的工作环境需要, 他们将 UNIX 移植到 PDP-11 上, 并逐渐增加了新的功能。很快, UNIX 开始在贝尔实验室内部流行, 许多人都投入到它的开发中来。1971 年,《UNIX 程序员手册》第 1 版出版, 这之后直到 1989 年, 贝尔实验室又相继发行了 10 个版本的 UNIX 和相应的手册。1973 年

Ritchie 用 C 语言重写了 UNIX (第 4 版), 这使得 UNIX 的可移植性大大增强, 这是 UNIX 迈向成功之路的

关键一步。1973 年 10 月, Thompson 和 Ritchie 在 ACM (Association for Computing Machinery, 计算机协会)

的 SOSP (Symposium Operating Systems Principles, 操作系统原理讨论会) 会议上发表了首篇 UNIX 论文, 这是 UNIX 首次在贝尔实验室以外亮相。

UNIX 的第一次移植是由 Wollongong 大学于 1976 年将其移植到 Interdata 机上。其它几次较早的移植包括: 1978 年, 微软公司与 SCO 公司合作将 UNIX 移植到 Intel 8086 上, 即 XENIX 系统 (最早的 UNIX 商业

变种之一); 1978 年, DEC 公司将 UNIX 移植到 VAX 上, 即 UNIX/32V3 (BSD 的前身)。

UNIX 的不断发展导致许多计算机公司开始发行自己机器上的 UNIX 增值商业版本。UNIX 的第一个商业变种是 1977 年 Interactive Systems 公司的 IS/1 (PDP-11)。20 世纪 80 年代著名的商业变种有 SUN 公司的 Sun

OS、微软公司与 SCO 公司的 XENIX 等。

20 世纪 70 年代中期到 80 年代中期, UNIX 的迅速发展, 众多大学和公司的参与, 使得 UNIX 的变种迅速增多。这些变种主要围绕 3 条主线: 由贝尔实验室发布的 UNIX 研究版 (First Edition UNIX 到 Tenth Edition

UNIX, 或称 V1 到 V10, 以后不再发行新版; 由加利福尼亚州大学伯克利分校发布的 BSD (Berkeley Software

Distribution) 和由贝尔实验室发布的 UNIX System III 和 System V。

到 20 世纪 80 年代, UNIX 已在从微型机到巨型机等众多不同机型上运行。作为通用操作系统, 当时 UNIX 的主要竞争对手是各计算机厂商的专有系统, 如 IBM 的 OS 360/370 系列等。

20 世纪 80 年代后期, UNIX 已经出现了很多变种, 变种增多导致了程序的不兼容性和不可移植 (同一应用程序在不同 UNIX 变种上不能直接运行)。因此, 迫切需要对 UNIX 进行标准化。这就导致了两大阵营

的出现。

1987 年, 在统一市场的浪潮中, AT&T 宣布与 SUN 公司合作, 将 System V 和 Sun OS 统一为一个系统。

其余厂商十分关注这项开发, 认为他们的市场处于威胁之下, 于是联合开发新的开放系统。他们的新机构

Open Software Foundation（开放软件基金会，简称 OSF）于 1988 年成立。作为回应，AT&T 和 SUN 公司联

盟亦于 1988 年形成了 UNIX International（UNIX 国际，简称 UI）。这场“UNIX 战争”将系统厂商划分成 UI 和 OSF 两大阵营。UI 推出了 SVR4，而 OSF 则推出了 OSF/1。虽然两者都是 UNIX，但他们在系统构架、

命令操作以及管理方式上都有所不同。两者在市场上展开了激烈的竞争。

两种 UNIX 系统并存，却又不能相互兼容，这对用户非常不利，因而直接影响了 UNIX 对用户的吸引力。随着 Microsoft 公司的迅速崛起，并以惊人的速度由传统的 PC 机市场向工作站和网络市场扩张，迫使 UI

和 OSF 两大阵营不得不相互让步、握手言和，从而共同制定了应用程序接口 API（Application Program Interface）标准技术规范，并联合开发共同开放软件环境 COSE（Common Open Software Environment）。

6. 简述 Linux 操作系统的发展历史。

答：Linux 最初是由芬兰赫尔辛基大学计算机系大学生 Linus Torvalds，在从 1990 年底到 1991 年的几个月中，为了自己的操作系统课程学习和后来上网使用而陆续编写的，在他自己买的 Intel 386 PC 机上，利用

Tanenbaum 教授自行设计的微型 UNIX 操作系统 Minix 作为开发平台。据 Linus 说刚开始的时候他根本没有想

到要编写一个操作系统内核，更没想到这一举动会在计算机界产生如此重大的影响。最开始是一个进程切换

器，然后是自己上网需要而自行编写的终端仿真程序，再后来是为他从网上下载文件而自行编写的硬盘驱

动程序和文件系统。这时候他发现自己已经实现了一个几乎完整的操作系统内核，出于对这个内核的信心和

美好的奉献与发展愿望，Linus 希望这个内核能够免费扩散使用，但出于谨慎，他并没有在 Minix 新闻组中公

布它，而只是于 1991 年底在赫尔辛基大学的一台 FTP 服务器上发了一则消息，说用户可以下载 Linux 的公开

版本（基于 Intel 386 体系结构）和源代码。从此以后，奇迹发生了。

1993 年，Linux 的第一个产品——Linux 1.0 版问世的时候，是按完全自由版权进行扩散的。它要求所有的源代码必须公开，而且任何人均不得从 Linux 交易中获利。然而半年以后，Linus 开始意识到这种纯粹的自

由软件理想对于 Linux 的扩散和发展来说，实际上是一种障碍而不是一股推动力，因为它限制了 Linux 以磁盘

复制或者 CD-ROM 等媒体形式进行扩散的可能，也限制了一些商业公司参与 Linux 进一步开发并提供技术支

持的良好愿望。于是 Linus 决定转向 GPL 版权，这一版权除了规定有自由软件的各项许可权之外，还允许用

户出售自己的程序复制品。这一版权上的转变后来证明对 Linux 的进一步发展确实极为重要。从此以后，便

有多家技术力量雄厚又善于市场运作的商业软件公司加入了原先完全由业余爱好者和网络黑客所参与的这场

自由软件运动，开发出了多种 Linux 的发布版本，增加了更易于用户使用的图形界面和众多的软件开发工具，极大地拓展了 Linux 的全球用户基础。Linus 本人也认为：“使 Linux 成为 GPL 的一员是我一生中做过

的最漂亮的一件事。”一些软件公司，如 Red Hat、InfoMagic 等也不失时机地推出了自己的以 Linux 为核心

的操作系统版本，这大大推动了 Linux 的商品化。在一些大的计算机公司的支持下，Linux 还被移植到 Alpha、Power PC、MIPS 及 SPARC 等为处理机的系统上。

随着 Linux 用户基础的不断扩大，性能的不断提高、功能的不断增加，各种平台版本的不断涌现，以及越来越多商业软件公司的加盟，Linux 已经在不断地向高端发展，开始进入越来越多的公司和企业计算领域。Linux 被许多公司和 Internet 服务提供商用于 Internet 网页服务器或电子邮件服务器，并已开始在很多企

业计算领域中大显身手。1998 年下半年，由于 Linux 本身的优越性，使得它成为传媒关注的焦点，进而出现

了当时的“Linux 热”：首先是各大数据库厂商（Oracle、Informix、Sybase 等）；继而是其它各大软硬件厂

商（IBM、Intel、Netscape、Corel、Adeptec、SUN 公司等），纷纷宣布支持甚至投资 Linux（支持是指该厂

商自己的软硬件产品支持 Linux，即可以在 Linux 下运行，最典型的是推出 xxx for Linux 版或推出预装 Linux

的机器等）。即使像 SUN 和 HP 这样的公司，尽管它们的操作系统产品与 Linux 会产生利益冲突，也大力支

持 Linux，从而达到促进其硬件产品销售的目的。

7. 自由软件的含义是什么？

答：自由软件的自由（free）有两个含义：第一是免费，第二是自由。免费是指自由软件可免费提供给任何用户使用，即便是用于商业目的，并且自由软件的所有源程序代码也是公开的，可免费得到。自由是指它

的源代码不仅公开而且可以自由修改，无论修改的目的是使自由软件更加完善，还是在对自由软件进行修改

的基础上开发上层软件。总之，可以对它做自己喜欢做的任何事情，除了一两件不能做的事之外（如不能宣

称这个系统是你自己开发的）。自由软件的出现给人们带来很大的好处：首先，免费可给用户节省相当的费

用；其次，公开源码可吸引尽可能多的开发者参与软件的查错与改进；在开发协调人的控制下，自由软件新

版本的公布、反馈、更新等过程是完全开放的。

第二章 进程管理

2.1 选择题

1. 进程在发出 I/O 请求后，可能导致下列哪种进程状态演变？D

- A. 就绪 → 执行 B. 执行 → 就绪
- C. 阻塞 → 执行 D. 执行 → 阻塞

2. “临界区”是指：C

- A. 一组临界资源的集合 B. 可共享的一块内存区
- C. 访问临界资源的一段代码 D. 请求访问临界资源的代码

3. 使用一个信号量协调 5 个进程对 3 个同类临界资源的访问，下列哪个信号量值不应该出现？D

- A. 3 B. 0 C. -1 D. -3

3 4 6 9 7 8 11 10

完成一个共同的任务，则诸进程是以 A 方

案 ——

式运行的。对临界资源的访问时采用 B 方 ——

式，对于相互合作的进程采用 C 方式以协调各进程执行的 D。 ——

选择答案：

(1) 共享 (2) 独立 (3) 互斥 (4) 同步 (5) 次序 (6) 次数 (7) 异步

18. 一个数据表格 (Dtab)，在同一时间只允许一个写者去写，容许

A B C D

答

4 3 7 2

RN 个读者同时去读。每个读者读前必须在登记表 (Tab) 上登记，退出

案

时则要删除相应的登记项。对以下流程填入正确语句实现同步操作。

(注：Tab=Ω 表示登记表为空，即没有读者或写者正在操作 Dtab。)

var mutex, wmutex, count : semaphore := 1, 1, RN //语义?

begin

parbegin

reader : begin

repeat

if tab=Ω then -----tab 的初值是多少呢？目前教学中《数据结构》均由 Pascal 语言改成了 C 语言。若用 Pascal 写信号量，学生理解起来比较困难，考试易出差错。

P(wmutex)

Add entry

V(mutex)

perform reading dtab operation;

P(mutex)

delete entry

if Tab=Ω then □C

V(count);

until false

End

writer : begin

repeat

perform writing dtab operation

until false

end

parend

end

选择答案：

(1) p(mutex) (2) p(wmutex) (3) p(count)

(4) v(mutex) (5) v(wmutex) (6) v(count)

19. 在分时系统中用户登陆成功，系统要为该终端用

A B C D E F

答

1 6 5 4 2 5

户建立 A，并把它插入到就绪队列。正在执行的进程

案 ——

请求读磁盘数据，若数据不在输入缓冲区中，则调用

B 将自己投入到相应的 C。 ——

选择答案：

(1) 输入进程 (2) 子进程 (3) 终端解释进程 (4) 激活原语 (5) 唤醒原语 (6) 阻塞原语 (7)

阻塞队列 (8) 就绪队列 (9) 运行队列

20. 在含有线程的系统中，引入线程的目的是为了提高系统的

A B C

答

3 6 7

A，节省只有进程系统的 B。线程是一个 C 单位，线程切换基本不涉及

案 ——

D 的再分配。 —

选择答案：

(1) 吞吐量 (2) 交互能力 (3) 并发能力 (4) 执行 (5) 资源 (6) CPU (7) 占有资源 (8)

时间开销 (9) 空间开销 (10) 时空开销

21. 生产者与消费者分别利用计数信号量 empty，full 并借助单缓冲

A B C D

答

3 10 4 5

buffer 进行数据传输：

案

var empty,full: semaphore:=1,0;

begin

parbegin

producer: begin

repeat

produce an item in nextp:

□ ;

A

buffer:=nextp:

until false:

end

consumer: begin

repeat

```

nextc: =buffer:
consume the item in nextc:
until false
end
parend
end
选择答案:

```

(1) wait(full) (2)wait(empty) (3)signal(full) (4)signal(empty)

22. 利用消息缓冲通信机制进行通信，a 为发送进程的发送区首地

A B C D

答

2 3 1 4

址，b 为接收进程的接收区首地址，mq、mutex、sm 分别为接收进
案

程消息队列的队首地址、互斥信号量和同步信号量，对以下发送原
语和接收原语实现正确的同步操作：

procedure send (receiver,a) // receiver 为接收者，a 为发送区首址

begin

getbuf(a.size ,i)

i.sender:= a.sender;

i.size:= a.size;

i.text:= a.text;

i.next:= 0;

getid (PCBset, receiver, j);

☐ ;

A

insert (j.mq, i)

☐ ;

B

signal (j.sm);

end

procedure receive()

begin

j:= internal name;

☐ ;

C

Wait(j.mutex);

☐ ;

D

b.sender:= i.sender;

b.size:= i.size;

b.text:= i.text;

end

选择答案：

(1) wait(j.mutex) (2)wait (j.sm) (3)signal(j.mutex) (4)signal(j.sm)

23. 进程 get、copy、put 分别对缓冲区 buffer1、buffer2 进

A B C D

答

1 3 2 i:=delete(j.sm)

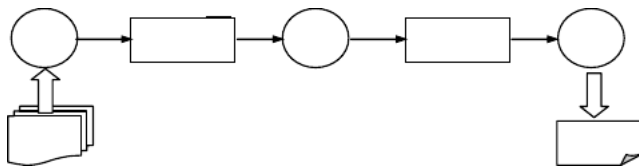
行操作。get 把产生的数据送 buffer1；copy 把 buffer1 的数

案

据复制到 buffer2 中；put 把 buffer2 中的数据取出来打印。

请选择相关的 wait、signal 语句完善给出的流程：

流程中 s1 是 get 的私用信号量；s2、s3 是 copy 的私用信号量；s4 是 put 的私用信号量。



var s1, s2, s3, s4:semaphore := A ;

get copy put

buffer1 buffer2

buffer1, buffer2 : item;

begin

parbegin

get : begin

卡片

打印机

repeat

produce an item in x;

wait(s1) ;

buffer1:= x ; ☐

B ;

until false ;

end

copy : begin

repeat

wait(s2) ; ☐

C ;

copy buffer1 to buffer2; ☐

D ;

E ;

until false ;

end

put : begin

repeat ☐


```
F ;
y := buffer2;
signal(s3) ;
print data in y ;
until false ;
end
parent
end
```

选择答案:

A: ① 1,1,1,1 ② 0,1,0,1 ③ 1,0,1,0 ④ 1,1,0,0

B、C、D、E、F:

① wait(s1) ② wait(s2) ③ wait(s3) ④ wait(s4)

⑤ signal(s1) ⑥ signal(s2) ⑦ signal(s3) ⑧ signal(s4)

24. 用户级线程与内核线程之间的关系存在多对一、

A B C D E F

答

3 6 3 8 5 4

一对一、多对多三种模型。其中：多对一模型是指多

案

个用户级线程映射到一个内核线程。在这种模型中用

--	--

户级线程在内核之上支持，并在用户空间通过 A 实现；对线程的创建、管理、和调度 B 内核支持；

--

任何一个用户级线程执行了引起阻塞的系统调用，则该 C 阻塞；开发人员可创建任意多的线程，系统

的并发度（并发性能）D。一对一模型是指每个用户级线程映射到一个内核线程，在这种模型中线程的

--

并发能力 E，能 F 运行在多处理器系统上。多对多模型是指多路复用了许多用户级线程到同样数量

--

或更小数量的内核线程上，开发人员可创建 G 的用户级线程，内核线程可在多处理器系统上并行运

行。

选择答案:

A: ① 内核 ② 线程库 ③ 原语 ④ 系统调用

B: ① 需要 ② 不需要

C、D、E、F、G:

① 线程 ② 程序 ③ 进程 ④ 任意数量 ⑤ 提高

⑥ 并发 ⑦ 不变 ⑧ 并行 ⑨ 有限数量

A B C D E F G

答

1 1 2 7 5 8 4

案

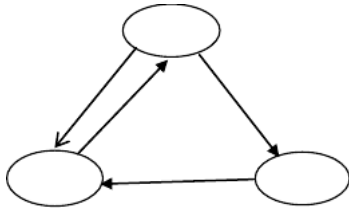
2.2 填空题

1. Sa、Sb、Sc 是已定义信号量，初值分别为 1、0、0；x、y、z 是公共变量。下面三个并发程序段执行

结束后, $x=19$, $y=10$, $z=28$ 。 _____

```
prA()
{ P(Sc);
  z=x+y;
  V(Sb);
}
prB()
{ y=18;
  P(Sb); V(Sc);
  P(Sb);
  y=z-y;
  V(Sa);
}
prC()
{ P(Sa);
  x=10;
  V(Sb);
  P(Sa);
  x=z-9;
}
```

2. 从物理意义上讲, 信号量的值大于 0 时, 其值表示 有效资源的数量 。 _____
3. 管程的三个组成部分为 局部于管程的共享变量说明 、 对该数据结构进行操作的一组过程、 对局部于管程的数据设置初始值的语句 。 _____
4. 系统感知进程存在的唯一标识为 进程控制块 (PCB) 。 _____
5. 从结构上看每个进程由 用户程序、 用户数据、 系统栈、 进程控制块 组成。 _____
6. 一段时间内仅允许一个进程访问的资源称为 临界资源 。 _____
7. 在操作系统控制下的多个程序的执行顺序和每个程序的执行时间是不确定的, 这种现象称为操作系统的 异步性 。 _____
8. 进程的动态特征是指 进程是动态创建、动态执行和动态消亡的 。 _____
9. 设有 n 个进程共享一个临界区, 若最多允许 m 个进程 ($m < n$) 同时进入临界区, 则所采用的信号量的初值应为 m , 信号量值的变化范围为 $[m-n, m]$ 。 _____
10. 在 Unix V 系统中, PCB 大致可分为 进程表项 和 U 区 两部分。 _____
11. 并发进程之间存在着 同步 和 互斥 两种关系。 _____
12. 把执行不能分割的过程称为 原语 。 _____
13. 进程调度的两种基本方式为 非剥夺调度 和 可剥夺调度 。 _____
14. 有 2 个同类临界资源, 被 5 个并发进程访问, 使用信号量机制实现互斥, 则初值为 2 , 信号量值的变化范围为 $[-3, 2]$ 。 _____
15. 进程三种基本状态之间有四种基本变迁关系: 1、2、3、4, 如图所示。对下列给出的每个事件, 在括号 “ () ” 中分别填上对应的变迁关系。例如: 若事件只引起了变迁关系 “2”, 则只在括号 “ () ” 中填入 2; 若事件引起了变迁关系 “2”, 随后一定会再产生变迁关系 “1”, 则在 “ () ” 中填入 2、1。



运行

3

2

1

4

就绪

阻塞

☐ 设备驱动进程启动了通道程序（ 3 ）。

1

☐ 进程发出了读键盘指令（ 3 ）。

2

☐ 接受进程取消息时，发现消息队列中无消息（ 3 ）。

3

☐ 运行进程时间片用完（ 2 ）。

4

☐ 高优先级进程剥夺运行进程的 CPU（ 2，1）。

5

☐ 输入设备输入完成中断，且中断时没有其他进程运行（ 4，1）。

6

☐ 某资源的信号量 S 的值为负时，运行进程执行了 Signal（S）操作（ 4 ）。

7

☐ 批处理系统中，后备队列有新作业到来（ 1 ）。

8

☐ 在就绪队列不空时，运行进程完成（ 1 ）。

9

☐ 磁盘驱动进程把读入的数据传送给用户（ 4，1 ）。

10

2.3 判断题

1. 原语仅可在管态下执行。对

2. 原语可在目态下执行。错

3. 进程就是作业。错

4. 所谓并行是指两个或两个以上的事件在同一时刻发生。对

5. 处理机从目态转变为管态是通过置程序状态字来实现的。错

6. 所谓并发是指两个或两个以上的事件在同一时刻发生。错

7. 操作系统的不确定性是指同一程序使用相同的输入、在相同的环境下，•经过多次运行却可能获得完全不同的结果。错

8. 处理机从管态转变为目态是通过置程序状态字来实现的。对

9. 广义指令必须在管态下执行。对
10. 在采用顾客/服务员模型的系统中，服务员必须在管态下运行。错
11. 在消息缓冲通信中，消息队列属于临界资源。对
12. 访管中断是由于程序执行特权指令引起的。对
13. 信号量的值不一定表示某类可用资源的数量。对
14. 在 UNIX 系统中所有进程都可在核心态和用户态两种状态下运行。错
15. 并发指的是在操作系统本身存在着许多同时的或并行的活动。错
16. 在 UNIX 系统中，所有进程都是利用系统调用 fork 创建的。错
17. 我们可以通过设置状态字，将 CPU 设置为内核态或用户态。对
18. 所谓的用户态、内核态实际上是处理器的一种状态，而不是程序的状态。对

2.4 简答题

1. 对比“进程”和“线程”的异同。

答：相同点：

- (a) 二者都具有 ID，一组寄存器，状态，优先级以及所要遵循的调度策略。
- (b) 每个进程都有一个进程控制块，线程也拥有一个线程控制块。
- (c) 线程和子进程共享父进程中的资源；线程和子进程独立于它们的父进程，竞争使用处理器资源；线程和子进程的创建者可以在线程和子进程上实行某些控制，比如，创建者可以取消、挂起、继续和修改线程和子进程的优先级；线程和子进程可以改变其属性并创建新的资源。

不同点：

- (a) 线程是进程的一部分，一个没有线程的进程是可以被看作单线程的，如果一个进程内拥有多个进程，进程的执行过程不是一条线（线程）的，而是多条线（线程）共同完成的。
- (b) 启动一个线程所花费的空间远远小于启动一个进程所花费的空间，而且，线程间彼此切换所需的时间也远远小于进程间切换所需要的时间。
- (c) 系统在运行的时候会为每个进程分配不同的内存区域，但是不会为线程分配内存（线程所使用的资源是它所属的进程的资源），线程组只能共享资源。对不同进程来说，它们具有独立的数据空间，要进行数据的传递只能通过通信的方式进行，这种方式不仅费时，而且很不方便。而一个线程的数据可以直接为其他线程所用，这不仅快捷，而且方便。
- (d) 与进程的控制表 PCB 相似，线程也有自己的控制表 TCB，但是 TCB 中所保存的线程状态比 PCB 表中少多了。
- (e) 进程是系统所有资源分配时候的一个基本单位，拥有一个完整的虚拟空间地址，并不依赖线程而独立存在。

2. 为什么要引入信号量集机制。信号量机制可以被用于何种场合？

答：1965 年，Dijkstra 在讨论并发进程时指出信号量机制是一种卓有成效的进程同步机制。由于操作系统可以被看成一个并发进程集，如果提供一种能支持他们之间合作的可靠机制，用户就能很容易地使用它们。因此它现已被广泛地应用于各种类型的操作系统中。

信号量的主要含义和用途如下：

- 信号量的含义。信号量是一个用来实现同步的整型或记录型变量，除了初始化外，对它只能执行 wait 和 signal 这两种原子操作。
- 信号量的物理意义。一个信号量 S 通常对应于一类临界资源。
- 用信号量实现互斥。为了实现进程对临界资源的互斥访问，需为每类临界资源设置一初值为 1 的互斥

信号量 mutex。

○ 用信号量实现前趋关系。为实现前趋关系 $P \rightarrow P$ ，可为它们设置一个初值为 0 的信号量 S。

4

ij

3. 原语与广义指令的主要区别。

答：所谓原语，是由若干条指令所组成都是一个指令序列，用来实现某个特定的操作功能。这个指令序列的执行是连续的，具有不可分割性，在执行时也不可间断，直到该指令序列执行结束。原语是操作系统核心的一个组成部分。原语必须在管态下执行，并且常驻内存。

广义指令是作为机器指令的扩充而提供的，以便增加机器的功能。它是通过执行相应的程序模块来实现的，使计算机成为功能强大的虚拟计算机。现代计算机 CPU 的指令系统中都有一条称为“访管”的指令，用户特别是编程人员可以利用这条指令来访问操作性并向它提出要求。

4. 管态和目态有何区别？如何区分二态？

答：大多数计算机系统将 CPU 执行状态分为管态和目态。CPU 的状态属于程序状态字 PSW 的一位。

CPU 交替执行操作系统程序 and 用户程序。

管态又叫特权态，系统态或核心态。CPU 在管态下可以执行指令系统的全集。通常，操作系统在管态下运行。

目态又叫常态或用户态。机器处于目态时，程序只能执行非特权指令。用户程序只能在目态下运行，如果用户程序在目态下执行特权指令，硬件将发生中断，由操作系统获得控制，特权指令执行被禁止，这样可以防止用户程序有意或无意的破坏系统。

从目态转换为管态的唯一途径是中断。

从管态到目态可以通过修改程序状态字来实现，这将伴随着由操作系统程序到用户程序的转换。

5. 什么是并发？什么是并行？

答：并发与并行是两个既相似而又不相同的概念：并发性，又称共行性，是指能处理多个同时性活动的的能力，如下左图所示；并行是指同时发生的两个并发事件，具有并发的含义，如下右图所示。

并发则不一定并行，也亦是说并发事件之间不一定要同一时刻发生。所有的并发处理都有排队等候，唤醒，执行至少三个这样的步骤。所以并发肯定是宏观概念，在微观上他们都是序列被处理的，只不过资源不会在某一个上被阻塞(一般是通过时间片轮转)，所以在宏观上看多个几乎同时到达的请求同时在被处理。如果是同一时刻到达的请求也会根据优先级的不同，而先后进入队列排队等候执行。

6. 进程的三种基本状态是什么？它们之间相互转换的主要原因是什么？

答：进程的三种基本状态是：

○ 就绪状态 (Ready)，存在于处理机调度队列中的那些进程，它们已经准备就绪，一旦得到 CPU，就立

1

即可以运行，这些进程所处的状态为就绪状态（有多个进程处于此状态）。

○ 运行状态(Running)，当进程由调度/分派程序分派后，得到 CPU 控制权，它的程序正在运行，该进程

2

所处的状态为运行状态（在系统中，总只有一个进程处于此状态）。

○ 阻塞状态 (blocked)，若一个进程正在等待某个事件的发生（如等待 I/O 的完成），而暂停执行，这时，

3

即使给它 CPU 时间，它也无法执行，则称该进程处于阻塞状态。

进程状态转换的主要原因有：

运行 \nrightarrow 阻塞：等待某事件的发生（如请求磁盘或键盘等执行 I/O 操作、等待同步信号、等待消息等）。

阻塞 \nrightarrow 就绪：等待的事件已经发生（如 I/O 完成，消息已到达）

运行 \nrightarrow 就绪：在可剥夺调度方式中，更高优先级的进程到达；在时间片轮转调度方式中，进程运行的时间片到。

7. 何谓临界资源？使用临界资源的原则是什么？使用临界资源的诸进程间如何实现进程同步。

答：一次仅允许一个进程使用的资源称为临界资源。内存变量、指针、数组等等也是临界资源。

入临界区的准则：

(1)每次至多有一个进程处于临界区；

(2)当有若干个进程欲进入临界区时，应在有限的时间内使其进入；

(3)进程在临界区内仅逗留有限的时间。

可以采取如下策略实现临界资源的访问同步：

(1) 锁机制

(2) 信号量机制

(3) 管程

(4) 消息机制

8. 何谓管程，管程是由哪几部分组成？说明引入管程的必要性。

答：Hansan 为管程所下的定义：“一个管程定义了一个数据结构和能为并发进程所执行（在该数据结构上）的一组操作，这组操作能同步进程和改变管程中的数据”。

由上述定义可知，管程由四部分组成：

○ 管程内部的共享变量；

1

○ 管程内部的条件变量；

2

○ 管程内部并行执行的进程；

3

○ 对局部于管程内部的共享数据设置初始值的语句。

4

局部于管程的数据结构，只能被局部于管程的过程所访问，任何管程之外的过程都不能访问它；反之，局部于管程的过程也只能访问管程内的数据结构。由此可见，管程相当于围墙，它把共享变量和对它进行操作的若干个过程围了起来，所有进程要访问临界资源时，都必须经过管程才能进入，而管程每次只允许一个进程进入管程，从而实现了进程的互斥。

引入管程的必要性：管程的基本思想是将共享变量和对它们的操作集中在一个模块中，操作系统或并发程序就由这样的模块构成。这样模块之间联系清晰，便于维护和修改，易于保证正确性。

9. 对相关临界区的管理有哪些要求？

答：为了使并发进程能正确地执行，对若干进程共享某一变量（资源）的相关临界区应满足以下四点要求：

①空闲让进。当无进程处于临界区时，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，有效地利用临界资源。

②以忙则等待。当已有进程进入临界区时，其它试图进入临界区的进程必须等待，以保证对临界资源的互斥访问。

③有限等待。对要求访问临界资源的进程，应保证在有限的时间内能进入自己的临界区，以免陷入“死等”状态。

○ 让权等待。当进程不能进入自己的临界区时，应立即释放处理机，以免进程陷入“忙等”状态。

4

10. 进程产生的主要原因有哪些？Windows XP 在启动时会有哪些进程产生？

答：进程产生的主要原因有：系统初始化；正在执行进程创立程序（子进程）；用户请求创建新进程。

在一个系统初始化时，将有许多系统正常运行时必不可少的进程产生。如 Windows 初始化时自动产生对话管理（SMSS）、登录管理（WINLOGON）、安全管理（LSASS），Windows 子系统（CSRSS）、Windows

壳（explore）等系统进程。

11. 进程消亡的主要原因有哪些？

答：进程消亡的主要原因有：进程运行完成而退出（寿终）；进程因错误而自行退出（自杀）；进程被其他进程强行“杀死”（他杀）；进程因异常而强行终结（处决）。

12. 进程创建的主要步骤是什么？

答：进程创建的主要步骤是：分配进程控制块；初始化机器寄存器；初始化页表；将程序代码从磁盘读入内存；将处理器状态设置为“用户态”；跳转到程序的起始地址（即设置程序计数器）。

注：最后 2 步由硬件作为一个步骤一起完成。

13. 消息队列和管道有何区别？

答：管道是一个线性字节数组，发送线程在一端写入信息，接受线程则在另一端读取信息。管道所占的空间可以在内存，也可以在磁盘。而消息队列是一列具有头和尾的消息排列，仅存在于内存。

基于此，消息队列与管道的区别在于：

☐ 消息队列无需固定的读写进程，任何进程都可以读写；

1

☐ 消息队列可以支持多个进程，多个进程可以读写消息队列，即多对多，而不是管道的点对点；

2

☐ 消息队列只在内存中实现；

3

☐ 消息队列并不是只在 UNIX 和类 UNIX 操作系统实现，几乎所有的主流操作系统都支持消息队列。

4

2.5 综合题

1. 某宾馆门前有一个出租汽车停车位，假设宾馆每一位顾客出门都要乘坐出租车，并且对顾客约定：如果有其它顾客在此停车位等车则在旁等待；否则在此等车；此停车位有车则乘坐。对出租车作出约定：如果此停车位已停有车，则等待此停车位空闲；否则停到此停车位等待顾客；有顾客则搭载顾客离开。试用 P、V 原语编写程序描述顾客与出租车的行为。

答：设置信号量 empty1, empty2, vehicle_ready, cust_ready，其初值分别为 1,1,0,0。顾客与出租车的行为描述如下：

出租车：repeat

P(empty1);

进入停车位，等待顾客；

V(vehicle_ready);

P(cust_ready);

顾客上车后，开走；

V(empty1);

forever

顾客：repeat

P(empty2);

进入停车位，等待出租车；

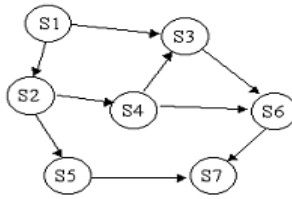
V(cust_ready);

P(vehicle_ready);

坐上出租车；

V(empty2);

forever



2. 使用 P、V 原语实现图示的前趋关系。

答：设置信号量 s12, s13, s25, s24, s43, s36, s46, s67, s57，初值均为 0。各个进程的描述分别如下：

S1: ...

V(s12);

V(s13);

S2: P(s12);

...

V(s24);

V(s25);

S3: P(s13);

P(s43);

...

V(s36);

S4: P(s24);

...

V(s43);

V(s46);

S5: P(s25);

...

V(s57);

S6: P(s36);

P(s46);

...

V(s67);

S7: P(s57);

P(s67);

...

3. 现有四个进程 R, R, W, W，它们共享可以存放一个数的缓冲区。进程 R•每次把从键盘上读入

12121

的一个数存到该缓冲区中，供进程 W 打印输出；进程 R•每次从磁盘上读一个数存放到该缓冲区中，供

12

W 打印输出。当一个进程把数存放到缓冲区后，•在该数还没有被打印输出之前不准任何进程再向缓冲

2

区中存数。•当一个进程已把缓冲区中的数打印输出后，在缓冲区中还没有存入一个新的数之前不准任何进程再从缓冲区中取数打印。•用 P V 操作来协调它们的工作。

解：设置信号量：empty, full1 和 full2，其初值分别为 1, 0, 0。

进程描述如下：

R: repeat

1


```

    准备数据;
    P(empty);
    放置数据到 buff;
    V(full1);
    until forever;

```

W: repeat

```

1
    P(full1);
    放置数据到 buff;
    V(empty);
    消费数据;
    until forever;

```

R: repeat

```

2
    准备数据;
    P(empty);
    放置数据到 buff;
    V(full2);
    until forever;

```

W: repeat

```

2
    P(full2);
    放置数据到 buff;
    V(empty);
    消费数据;
    until forever;

```

4. 设有一数据区，有若干进程要去读或写它。各进程要遵循下列原则：

写是互斥的。当一进程正在写时，其它进程既不能读也不能写。

读可同时进行。只要没有进程正在写，则任何进程都可读。

请用 P、V 操作写出读写过程的同步算法。

答：算法描述如下：

var w, s, mutex: semaphore; // w 表示有效资源的个数（写者信号灯），初值为 1，表示一次仅允许一个写者进程进入。S 和 mutex 分别是写者和读者互斥信号灯。

RC: integer; //读者计数器，进入临界区中的读者数目。

begin w:=1; s:=1; mutex:=1; RC:=0;

Cobegin

Reader:

Repeat

P(w); //登记

P(mutex);

if RC=0 then p(s);

RC:=RC+1;

V(mutex);

V(w);

```

    Reading;
    P(mutex); //退出
    RC:=RC-1;
    if RC=0 then v(s);
    V(mutex);
until FALSE
Writer:
    Repeat
        P(w);
    P(s);
Writing;
    V(s);
    V(w);
until FALSE
coend

```

5. 读者优先的读者-写者问题。除了上述两个规则外，还增加读者优先的规定，即当有读者和写者同时等待时，则首先满足读者。其算法描述如下：

```

var s, mutex: semaphore;
    RC: integer; //读者计数器，进入临界区中的读者数目。
begin s:=1; mutex:=1; RC:=0;
cobegin
Reader:
    Repeat
        P(mutex);
        if RC=0 then p(s);
        RC:=RC+1;
        V(mutex);
        Reading;
        P(mutex);
        RC:=RC-1;
        if RC=0 then v(s);
        V(mutex);
    until FALSE
Writer:
    Repeat
        P(s);
Writing;
        V(s);
    until FALSE
coend

```

6. 写者优先的读者-写者问题。

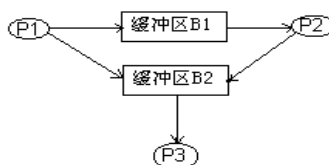
Var m, w, s, mutex, k: semaphore; //m 表示读者信号灯（一次读），s 表示写者信号灯（一次写），mutex 表示读者互斥信号灯，k 表示写者互斥信号灯。

RC, WC: integer;

```

begin m:=w:=s:=mutex:=k:=1; RC:=WC:=0;
Cobegin
Reader:
  Repeat
P(m);
P(w);
P(mutex);
if RC=0 then p(s);
RC:=RC+1;
V(mutex);
V(w);
V(m);
Reading;
P(mutex);
RC:=RC-1;
IF RC=0 then v(s);
V(mutex);
  until FALSE;
Writer:
  Repeat
P(k);
if WC=0 then p(w);
WC:=WC+1;
V(k);
P(s);
Writing;
V(s);
P(k);
WC:=WC-1;
if WC=0 then v(w);
V(k);
  until FALSE
Coend

```



7. 假设有一如图所示的工作模型，具有三个并发进程 P1、P2 和 P3，两个单缓冲 B1 和 B2。进程 P1 负责不断从输入设备读数据，若读入的数据为正数，则直接送入 B2，否则应先将数据送入 B1，经 P2 取出加工后再送入 B2，P3 从 B2 中取信息输出。请用信号量和 P、V 操作描述进程 P1、P2、P3 实现同步的算法。

答：设置信号量 empty1，empty2，其初值为 1；

信号量 full1, full2, 其初值为 0。

各个进程的描述如下:

P1:

repeat

读入一个数据 data;

if(data > 0) {

P(empty2);

放数据 data 到 buffer2 中;

V (full2);

}

else {

P(empty1);

放数据 data 到 buffer1 中;

V (full1);

}

until forever

P2:

repeat

P(full1);

从 buffer1 中取出数据 data;

然后处理该数据 data 成 DATA;

P(empty2);

将数据 DATA 放入 buffer2 中;

V (full2);

V (empty1);

until forever

P3:

repeat

P(full2);

从 buffer2 中取出数据 data;

V (full2);

打印数据 data;

until forever

8. 在天津大学与南开大学之间有一条弯曲的小路, 这条路上每次每个方向上只允许一辆自行车通过。

但其中有一个小的安全岛 M, 同时允许两辆自行车停留, 可供两辆自行车已从两端进入小路的情况下错车使用。如图所示。

下面的算法可以使来往的自行车均可顺利通过。其中使用了 4 个信号量, T 代表天大路口资源, S 代表南开路口资源, L 代表从天大到安全岛一段路的资源, K 代表从南开到安全岛一段路的资源。程序如下, 请在空白位置处填写适当的 PV 操作语句, 每处空白可能包含若干个 P, V 操作语句。

begin

t:=1;s:=1;l:=1;k:=1;

cobegin

从天大到南开的进程

begin

```

    _____(1)____p(t); p(l);____
    通过 L 路段;
    进入安全岛 M;
    _____(2)____v(l); v(t); p(s); p(k);____
    通过 K 路段
    _____(3)____v(k);v(s);____
end
从南开到天大的进程
begin
    略, 与“从天大到南开的进程”相反。
end
coend
end.

```

9. 五个哲学家在一块儿思考问题并一起用餐。用餐时, 它们公用一个由 5 把椅子围成的圆桌。每把椅子归某个哲学家使用。桌子中间是一些“永远也吃不完”的饭菜。桌子上还放有 5 个盘子和 5 支筷子。当哲学家们思考问题时, 它们互不干扰。一个哲学家需要用餐了, 他就进入餐厅, 走到餐桌边找到一把空椅子就座, 然后就试图去拿相邻的两支筷子。当然, 他不能去拿已经握在邻近椅子上同事手上的筷子, 也不能去拿位于其左、右同事位置之外的筷子。当一个需用餐的哲学家拿到邻近的两支筷子后, 他就开始用餐而不放下。当他用餐完毕, 就把手中的两支筷子放回原处再去思考他的问题。因此, 这些哲学家们的生活是一种单调的重复动作, 即这个问题可以概括为:

```

repeat
think,
eat
forever。

```

答:

○ 用一个信号量表示一支筷子。一个哲学家试图去拿一支筷子是通过在哪个信号量上执行一个 P 操

1

作来表示的, 放下一支筷子则是在相应的信号量上执行一个 V 操作来描述的。因此共享变量是筷子, 相应的程序为:

```

program diningphilosophers;
    var chopstick: array[0..4] of semaphore; (*binary*)
        i: integer;
    procedure philosopher(I:integer);
        begin repeat
think;
p(chopstick[i]);
p(chopstick[(i+1) mod 5]);
eat;
v(chopstick[i]);
v(chopstick[(i+1) mod 5]);
        forever
        end;
    begin (* main program *)
        for i:=1 to 4 do chopstick[i]:=1;

```

```

cobegin
  philosopher(0); philosopher(1); philosopher(2); philosopher(3); philosopher(4);
coend
end.

```

注：该算法简单，能够互斥，但有可能产生死锁（每个人都拿到了左边的筷子，又试图去拿右边的筷子时）。

○ Program diningphilosophers;

2

```

Monitor chopstickmonitor;
Var chopstick: array[0..4] of integer;
oktoeat: array[0..4] of condition;
i: integer;
procedure takechopstick(i:integer);
begin
if chopstick[i]<>2 then wait(oktoeat);
chopstick[(i+1) mod 5]:=chopstick[(i+1) mod 5]-1;
chopstick[(i-1) mod 5]:=chopstick[(i-1) mod 5]-1;
end;
procedur releasechopstick(i:integer);
begin
chopstick[(i+1) mod 5]:=chopstick[(i+1) mod 5]+1;
chopstick[(i-1) mod 5]:=chopstick[(i-1) mod 5]+1;
if chopstick[(i+1) mod 5]=2 then singal(oktoeat[(i+1) mod 5]);
if chopstick[(i-1) mod 5]=2 then singal(oktoeat[(i-1) mod 5]);
end;
begin (* monitor 赋初值部分*)
  for i:=1 to 4 do chopstick[i]:=2
end;
procedure philosopher(i:integer);
begin repeat
think;
takechopstick(i);
eat;
releasechopstick(i);
forever
end;
begin (* main program *)
cobegin
  philosopher(0); philosopher(1); philosopher(2); philosopher(3); philosopher(4);
coend
end

```

注：这种使用管程方法解决了死锁，同时又是互斥执行。

○ 下面给出一种使用 P、V 操作实现的办法，它不仅保证了安全性，而且也不会发生死锁和饥

3

饿现象。相应的程序描述如下：

```
program diningphilosophers;
  var chopstick: array[0..4] of semaphore; (*binary*)
      room: semaphore;
      i:integer;
  Procedure philosopher(i:integer);
  begin repeat
    think;
    P(room);
    P(chopstick[(i+1) mod 5]);
    eat;
    V(chopstick[i]);
    V(chopstick[(i+1) mod 5]);
    V(room);
  forever
  End;
begin (*main program *)
  for i:=0 to 4 do chopstick[i]:=1;
  room:=4;
cobegin
  philosopher(0); philosopher(1); philosopher(2); philosopher(3); philosopher(4);
coend
End.
```

除了增加一个信号量 **room** 外，这种解决方法与第一中解决方法类似。安全性同前面一样得到了

保证。也不会产生死锁，因为 **room** 保证了至多有 4 个哲学家试图（同时）去存取 **chopstick**。

而且根

据“鸽巢原理”（pigeon-hole priciple），在圆圈中的 4 个哲学家之间分配 5 支筷子的任何尝试将导致

至少一个哲学家会分得二支筷子，有关信号量 **room** 的不变式是：**room+(p(room)和 v(room)之间的**

进程个数)=4。

下面通过一系列的引理来证明该方法也不会出现饥饿现象。

引理 1 若进程 **P** 执行了 **P(chopstick[i])**，则最终它将完成该 **P** 操作的执行。

i

证明：若 **P_i** 仅因为 **chopstick[i]=0** 而没有完成 **P** 操作，这隐含 **P** 正在用餐（因为 **P** 这时右手

i-1i-1

边的那支筷子，在 **eat** 之前被 **P** 拿走了）。最终 **P** 用完餐就会马上执行 **V(chopstick[i])**而允许 **P** 继

i-1i-1i

续。

引理 2 若 **P** 正没完没了地等待在 **chopstick[i+1]**上，则 **P** 正没完没了地等待在 **chopstick[i+2]**

ii+1

上。

证明：仅考虑 P 和 P “竞争” 信号量 chopstick[i+1]的情况。若 P 在 think 出终止，则 chopstick[i+1]

ii+1i+1

就不可能阻塞 P。类似地，P 和 P 不可能同时都阻塞在同一信号量 chopstick[i+1]上（考虑该信号

iii+1

量的不变式）。因此，如果 P 被阻塞在 chopstick[i+1]上，而且假定 P 决不会“唤醒”这个信号量，

ii+1

那么剩下的唯一可能是 P 被无限期地阻塞在信号量 chopstick[i+2]上。

i+1

引理 3 若 P 执行 P(chopstick[i+1])，最终它将完成该操作（和 eat）的执行。

i

证明：通过连续四次运用引理 2，我们便得知，若 P 没完没了地等待在 chopstick[i+1]上，则 P 就没完没

ii+1

了地等待在 chopstick[i+j+1]上，其中 j=1,2,3,4，但这与信号量 room 的不变式相矛盾。

第三章 处理机调度与死锁

3.1 选择题

1. 操作系统中的高级调度是指：A
A. 作业调度 B. 进程调度 C. 进程交换调度 D. 线程调度
2. 作业经过下面哪一个过程进入“后备”状态？A
A. 作业创建 B. 作业调度 C. 进程调度 D. 作业终止
3. 要求进程一次性申请所需的全部资源，是破坏了死锁必要条件中的哪一条？B
A. 互斥 B. 请求与保持 C. 不剥夺 D. 循环等待
4. 使用“银行家算法”决定是否给进程分配资源，这种策略属于：B
A. 预防死锁 B. 避免死锁 C. 检测死锁 D. 解除死锁
5. 对资源编号，要求进程按照序号顺序申请资源，是破坏了死锁必要条件中的哪一条？D
A. 互斥 B. 请求与保持 C. 不剥夺 D. 循环等待
6. 通过破坏死锁必要条件之一来防止死锁产生，这种策略属于：A
A. 预防死锁 B. 避免死锁 C. 检测死锁 D. 解除死锁
7. 当某个作业被作业调度程序选中，进入内存开始运行时，作业的状态为:D
A、提交状态 B、完成状态
C、执行状态 D、后备状态
8. 死锁定理用于:A
A、预防死锁 B、解除死锁
C、避免死锁 D、检测死锁
9. 进入输入井的作业其状态处于（ D ）
A、提交状态 B、完成状态
C、执行状态 D、后备状态
10. 作业由后备状态转变为执行状态是通过以下那个调度程序实现的（ A ）
A、作业调度 B、进程调度
C、中级调度 D、驱臂调度

11. 以下那种调度算法不可能是剥夺式的 (A)
 A、先来先服务 B、最短 CPU 执行期优先
 C、最高优先权 D、轮转法
12. 在 UNIX 系统中, 用来实现进程换入换出的是 (B)
 A、0 进程 B、1 进程 C、kill 系统调用 D、作业调度进程
13. 可以破坏环路等待条件的策略是 (C)
 A、资源抢占 B、独享分配 C、按序分配 D、共享分配
14. 在操作系统中用户进程本身启动的唯一状态转换是__B__。
 A、调度 B、阻塞 C、时间片到 D、唤醒
15. 把资源按类型排序编号, 并要求进程严格按序申请资源, 这种方法摒弃了下述哪一个 (D) 死锁发生条件?
 A、互斥条件 B、部分分配条件 C、不剥夺条件 D、环路等待条件
16. 以下哪种调度算法不可能是剥夺方式的? (A)
 A、先来先服务 B、最短 CPU 执行期优先 C、最高优先权 D、轮转法
17. 作业调度无工作可做时处于 A 状态, 当后备队列有新作业录入时, 输入进程要 B 作业调度。进——
 程调度作为 C 执行, 通常采用两种调度方法, 批处理系统常采用 D 方式, 分时系统采用 E 方式。在——

内存和外存对换区之间完成‘页面对换’或‘分段对换’功能的进程称之为 F 调度。 ——

选择答案:

- (1) 激活 (2) 唤醒 (3) 就绪 (4) 挂起 (5) 阻塞 (6) 非抢占
 (7) 进程 (8) 原语 (9) 低级 (10) 高级 (11) 中级 (12) 抢占

--	--	--	--	--	--	--	--

A B C D E F

答

--	--	--	--	--	--	--	--

4 2 8 6 12 11

18. CPU 的状态可分为用户态和 A, CPU 状态由现

案 ——

行的 B 来描述。在用户态下运行时, CPU 执行特权指——

令将产生 C, 中断处理程序将 D 该程序的执行。 ——

选择答案:

- (1) 运行态 (2) 目态 (3) 系统态 (4) 通道寄存器 (5) 指令寄存器
 (6) 程序状态字 (7) I/O 中断 (8) 访管中断 (9) 程序中断 (10) 终止

- (11) 暂停 (12) 继续

19. 现有 3 个同时到达的作业 J1、J2 和 J3, 它们的执行时间分别为

A B C D

答

3 6 8 11

T1、T2 和 T3, 且 $T1 < T2 < T3$ 。系统按单道方式运行且采用短作业优

案

先算法, 则平均周转时间是__C__。

- A、 $T1+T2+T3$ B、 $(T1+T2+T3)/3$ C、 $(3T1+2T2+T3)/3$ D、 $(T1+2T2+3T3)/3$

3.2 填空题

1. 常用的多道处理系统的作业调度算法有 先来先服务调度算法、 基于优先级的作业调度、分时与 _____
优先级结合的作业调度、 综合考虑资源要求的调度策略、 轮转法 。

2. 产生死锁的原因 系统资源不足、进程推进顺序非法 。 _____

3. 一个作业从提交开始到完成，往往要经历 长程调度 、 短程调度 和中级调度三级调度。 _____

4. 常用的单道批处理作业调度有 短作业优先、最高响应比（优先） 和 先来先服务 。

5. 解决死锁问题常用的三种方法是 死锁的预防 、 死锁的避免 和 死锁的检测与解除 。 _____

3.3 判断题

1. 多用户实时操作系统一定采用剥夺调度方式。对
2. 进程发出 I/O 请求后将被阻塞，直至 I/O 操作完成。对
3. 死锁危害很大，操作系统要绝对防止死锁的发生。错
4. 不安全状态是死锁状态。错
5. 处于死锁的系统中，没有进程可再运行。错
6. 最短 CPU 执行期优先算法一定是剥夺式的。 错
7. 多级反馈队列属于非剥夺式调度。 错
8. 最短 CPU 执行期优先算法(SCBF--Shortest CPU Burst First)一定是剥夺式的。错
9. 一系统处于死锁状态则一定是不安全状态。对
10. 作业 A 处于运行状态，作业 A 的进程一定处于执行状态。错
11. 如果死锁的四个必要条件同时成立，则系统一定产生死锁。错
12. 在分时系统中，时间片越小，一个作业的总运行时间越短。错
13. 预防死锁可通过屏弃“互斥条件”实现。错
14. 进程被创建后处于就绪状态。对
15. 作业 A 处于运行状态，但作业 A 的进程 B 可能处于阻塞状态。对
16. 当进程提出资源请求得不到满足时，系统必定发生死锁。错
17. 当进程调度采用以下方案时，判断各语句的对错，对者在□中打“√”，错者在□中打“×”。
 - 1) 先来先服务调度：
 - (1) 到达时间指进入内存时间。错
 - (2) 进程获得 CPU 一直运行到完成或等待某事件才让出 CPU。对
 - (3) 有利于 I/O 忙的工作。对
 - 2) 短进程优先调度：
 - (1) 用户满意度好。对对
 - (2) 吞吐量好。对
 - (3) 长进程运行机会少。对
 - 3) 多级反馈队列调度（就绪队列 1、就绪队列 2、……，优先级依次下降）：
 - (1)各就绪对列的优先级依次下降，他们分得的时间片依次增加。对
 - (2)某就绪队列给予较大的时间片，是因为该队列的进程优先级高。错
 - (3)只有就绪队列 1 为空时，才去调度就绪对列 2 中的进程。对
 - (4)进入高优先级就绪队列的进程不能抢占低优先级对列进程的 CPU。错
 - (5)长进程可能出现在各个就绪对列中。对
 - (6)为了保证响应时间，就绪对列 1 的时间片设置必须使得大部分终端命令在几个时间

片内完成。错

3.4 简答题

1. 进程调度的时机有哪些？

答：进程调度的时机主要有以下几种：○ 正在执行的进程执行完毕或因发生某事件而不能再继续执行；

1

○ 执行中的进程因提出 I/O 请求而暂停执行；○ 在进程通信或同步过程中执行了某种原语操作如 P 操

23

作、阻塞、挂起原语等；○ 在可剥夺式调度中，有比当前进程优先权更高的进程进入就绪队列；○ 在时

45

间片轮转法中，时间片完。

通常系统是按先来先服务或优先权形式来组织调度队列。

2. 何为死锁？产生死锁的原因和必要条件是什么？

答：死锁是指多个进程的永久性阻塞现象，产生的原因主要有 2 个：进程间竞争资源；进程推进顺序非法。

产生死锁的四个必要条件：

(1) 互斥条件：一个资源每次只能被一个进程使用。

(2) 请求与保持条件：一个进程因请求资源而阻塞时，对已获得的资源保持不放。

(3) 不剥夺条件：进程已获得的资源，在未使用完之前，不能强行剥夺。

(4) 循环等待条件：若干进程之间形成一种头尾相接的循环等待资源关系。

这四个条件是死锁的必要条件，只要系统发生死锁，这些条件必然成立，而只要上述条件之一不满足，就不会发生死锁。

3. 死锁排除的方法有哪些？

答：○ 撤消陷于死锁的全部进程；

1

○ 逐个撤消陷于死锁的进程，直到死锁不存在；

2

○ 从陷于死锁的进程中逐个强迫放弃所占用的资源，直至死锁消失；

3

○ 从另外一些进程那里强行剥夺足够数量的资源分配给死锁进程，以解除死锁状态。

4

4. Windows NT 利用多线程可以更好地实现多任务。简略回答：什么是多任务？Windows NT 实现这种技术的方法是什么？

答：Windows NT 是 Microsoft 推出的面向工作站、网络服务器和大型计算机的网络操作系统，也可做 PC 操作系统。它与通信服务紧密集成，提供文件和打印服务，能运行客户机 / 服务器应用程序，内置了 Internet / Intranet 功能，是企业组网的标准平台。

多任务操作系统就是能同时运行多个任务的操作系统。Windows NT 是一种抢先式的多任务系统。所谓抢先式多任务系统就是 CPU 时间被分为一个个时间片，调度器将时间片分给当前优先级最高的线程 (thread，一旦有更高优先级的线程就绪，就马上抢先正在运行的线程，如果有多个优先级相同的线程，调度器把它们排队，先执行队首的线程，执行过的则放到队尾。抢先多任务功能是由操作系统内部的调度器(scheduler)实现的。

基于优先级的进程、线程在 Win32 API 中，将进程分成四类：空闲进程、常规进程、高级进程和实时进程。1)空闲进程：这类进程的线程只有系统处于空闲状态时才运行，如屏幕保护线程。2)常规进程：一般进程都属于该类型。3)高级进程：指执行紧急任务的进程，如弹出 Windows 的任务列表窗口。4)实时

进程：这类进程的线程可以抢占所有其它类型的线程，包括操作系统线程，例如，一个实时线程的执行可能会导致磁盘缓冲刷新或鼠标操作无效。Win32 API 中，线程有 32 级优先级，其中 1-15 级是可变优先级线程，16-31 级是实时线程(0 级为系统保留)，0 级最低，31 级最高，每个线程有一个基本优先级，它可以由程序改变。线程被调度时的优先级称为调度优先级，对于可变优先级线程，操作系统可以动态调整线程的调度优先级，调整范围最低为其基本优先级，最高为 15 级；而实时线程的调度优先级是不能被系统改变的。

5. 引起进程调度的因素有哪些？

答：引起进程调度的因素主要有：

(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) **进程在执行中调用阻塞原语将自己阻塞起来进入睡眠等待状态。**

(3) 进程在执行中调用了 P 原语操作，从而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) 进程在执行中提出 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程序时，可看做系统进程执行完毕，从而调度选择一新的用户进程执行。

(7) 在 CPU 执行方式是可剥夺时，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。

6. 为什么说多级反馈队列调度算法能较好地满足各类用户的需要？

答：对终端型作业用户，由于作业大都属于交互性，通常比较短小，系统只要使这些作业在第一队列所规定的时间片内完成，便能使用户满意。

对短批处理作业用户，他们的作业通常在第二、三队列各执行一次即可完成，其周转时间仍然较短。对长批处理作业用户，他们的作业在各个队列中依此运行，用户不必担心其作业长期得不到处理。

7. 实时调度与非实时调度的主要区别是什么？

答：(1)实时调度所调度的任务有完成时限，而非实时调度没有。从而，实时调度算法的正确与否不仅与算法的逻辑有关，也与调度算法调度的时限有关。

(2)实时调度要求较快的进程或线程切换时间，而非实时调度的进程或线程的切换时间较长。

(3)非实时调度强调资源利用率(批处理系统)或用户共享处理机(分时系统)，实时调度则主要强调在规定时限范围内完成对相应设备的控制。

(4)实时调度为抢先式调度，而非实时调度则很少采用抢先式调度。

8. 分时系统中有作业调度的概念吗？如果没有，为什么？

答：在分时系统中，一般不存在作业调度，而只有线程调度、进程调度和交换调度。这是因为在分时系统中，为了缩短响应时间，作业不是建立在外存，而是直接建立在内存中。在分时系统中，一旦用户和系统的交互开始，用户马上要进行控制。因此，分时系统中没有作业提交状态和后备状态。分时系统的输入信息经过终端缓冲区为系统直接接收，或立即处理，或经交换调度暂存外存中。

9. 某一系统分配资源的策略是：当进程提出申请资源时，只要系统有资源总是分配给它，系统无资源时让其等待。任一进程总是先释放已占有的资源后再申请新的资源，且每次申请一个资源，系统中的进程得到资源后总能在有限的时间内归还。证明该系统不会发生死锁。

证明：死锁的四个必要条件分别是：互斥、请求与保持、循环等待、非剥夺条件。

根据题目中的描述，任一进程总是先释放已占有的资源后再申请新的资源，且每次申请一个资源，系统中的进程得到资源后总能在有限的时间内归还。因此，四个必要条件中的“请求与保持条

件”被破坏，所以死锁不会发生。

10. 处理器调度的总体目标是什么？

答：处理器调度的总体目标是要达到极小化平均响应时间、极大化系统吞吐率、保持系统各个功能部件均处于繁忙状态和提供某种貌似公平的机制。

极小化平均响应时间就是要极小化用户发出命令和看到结果之间所花费的时间；极大化系统吞吐率就是在单位时间内完成尽可能多的程序；保持系统各个功能部件均处于繁忙状态就是要让 CPU 和输入输出设备均处于忙碌状态；而公平是任何系统都应该努力达到的目标。

对于不同的系统，在调度目标上是有细微的区别的。批处理系统追求系统吞吐率和 CPU 的利用率，不太重视系统响应时间；而交互式系统对系统响应时间要求严格。

11. 何谓优先级倒挂？如何解决优先级倒挂问题？

答：优先级倒挂是指一个低优先级任务持有一个被高优先级认为所需要的共享资源。这样高优先级任务因缺乏资源而处于阻塞状态，一直到低优先级任务释放资源为止。优先级倒挂可能造成系统故障，或激发表先定义的纠正措施，如美国的火星探路器 Mars Pathfinder 就是因为优先级倒挂而出现故障。

优先级倒挂问题在 20 世纪 70 年代就已经被发现，但一直未找到一个可以预测其发生的方法，只能通过一些手段来防止其发生。主要有：使用中断禁止；优先级上限；优先级继承。

3.5 综合题

进程 占用情况 请求情况

r r r r r r

123123

P 1 2 1

1

							P 2 1

1. 某系统有三类非剥夺性资源，其中 r 类有 2

2

1

P 2 2 1

个、r 类有 2 个、r 类有 4 个；当前三个进程（P、

3

231

P、P）对资源的占用和请求情况如右表：

23

①画出当前资源分配图；

②通过化简资源分配图判断是否发生死锁。

答：①当前资源分配图为：

②对于进程 P 来说，尚需要一个 r 就可以运行完毕，而此时系统的空闲资源为（1，0，0），如果将此资

11

源分配给 P 的话，那么等 P 运行完毕后，就可以释放已占有的其他 3 个资源（r 类 1 个，r 类 2 个，记

1113

做（1，0，2）），因此剩余资源就为（2，0，2）=（1，0，0）+（1，0，2）。而 P 需要的资源数为

3

（2，0，1），所以 P 能顺利执行完毕。而 P 执行完后，能释放 r 类资源 2 个，系统的空闲资源为（2，

332

2，2）=（2，0，2）+（0，2，0），能满足进程 P 的需求。

2

综上所述, 存在一个安全序列 P, P, P。所以系统不会死锁。

132

2. 设进程调度算法为: 按统计规律输入输出多的进程高优先, 使用 CPU 时间多的低优先, 在同一优先级上按先来先服务原理调度。

(1) 设计就绪队列

(2) 画出此算法的进程状态转换图

答: (1) 就绪队列如下:

(2) 状态转换图如下:

3. 在银行家算法中, 若出现下述的资源分配情况:

Allocation 数组	Need 数组	Available 向量
P0 0 0 3 2	0 0 1 2	1 6 2 2
P1 1 0 0 0	1 7 5 0	
P2 1 3 5 4	2 3 5 6	
P3 0 3 3 2	0 6 5 2	
P4 0 0 1 4	0 6 5 6	

试问该状态是否安全? 若安全给出一安全序列, 若此时进程 P2 提出请求 Request(1, 2, 2, 2), 请问系统能否将资源分配给它, 为什么?

答: 目前是安全状态, 存在一个安全序列 P0, P1, P3, P2, P4。

若此时进程 P2 提出请求 Request(1, 2, 2, 2), 系统不能将资源分配给它。这是因为若系统将资源分配给它后, 系统尚可用资源为 (0, 4, 0, 0), 不能满则任何进程的需求, 使系统处理不安全状态。

第四章 存储器管理

4.1 选择题

1. 可变分区存储管理中用链表记录分区使用情况, 为应用最差适应法(WF)分配空闲分区, 链表中应该按照下列哪种方法排列? D

A. 按分区起始地址递增排列 B. 按分区起始地址递减排列

C. 按分区大小递增排列 D. 按分区大小递减排列

2. 关于段页式存储管理系统中的页表数, 下面哪种说法比较准确? C

A. 整个系统有一个 B. 整个系统有多个

C. 每个进程有一个 D. 每个进程有多个

3. 可变分区存储管理中用链表记录分区使用情况, 为应用最先适应法(FR)分配空闲分区, 链表中应该按照下列哪种方法排列? A

A. 按分区起始地址递增排列 B. 按分区起始地址递减排列

C. 按分区大小递增排列 D. 按分区大小递减排列

4. 在可变分区存储管理中, 可能存在 (B)

A、内零头 B、外零头 C、A,B 均可能 D、A,B 均不可能

5. 分页存储管理系统中引入“快表”, 是为了: B

A. 保存最近访问的数据 B. 保存最近用过的页表项

C. 保存最近用过的物理地址 D. 保存最近用过的虚拟地址

6. 以下哪个叙述正确? D

A、使用静态重定位的系统, 用户的作业可不要求分配连续的存储空间。

B、使用静态重定位的系统, 作业可在内存中移动。

C、使用静态重定位的系统, 有可能为用户提供一个比内存大的多的地址空间。

- D、使用静态重定位的系统，无需增加硬件地址变换机构。
7. 以下那种存储管理不可用于多道程序系统中？ B
- A、固定分区存储管理 B、单一连续区存储管理
C、可变分区存储管理 D、段式存储管理
8. 以下哪种存储管理可使用静态重定位？ A
- A、固定分区存储管理 B、页式存储管理
C、可重定位分区存储管理 D、段式存储管理
9. 以下哪种存储管理会产生内零头？ A
- A、固定分区存储管理 B、可变分区存储管理
C、可重定位分区存储管理 D、段式存储管理
10. 以下哪个关于纯分页存储管理的叙述不正确？ C
- A、此种存储管理会产生内零头
B、此种存储管理要求作业一次全部调入内存
C、此种存储管理会产生外零头
D、此种存储管理不要求作业分配连续的存储区
11. 以下那个叙述不正确（ C ）
- A、使用动态重定位的系统，用户的作业可不要求分配连续的存储空间。
B、使用动态重定位的系统，作业可在内存中移动。
C、使用动态重定位的系统有可能为用户提供一个比内存大的多的地址空间。
D、使用动态重定位的系统有可能为用户提供一个比内存大的多的存储空间。
12. 以下哪种存储管理会产生内零头？ A
- A、固定分区存储管理 B、可变分区存储管理
C、可重定位分区存储管理 D、段式存储管理
13. 目标程序存在于（ B ）
- A、名空间 B、逻辑地址空间
C、储存空间 D、物理地址空间
14. 以下哪种存储管理必须使用动态重定位（ D ）
- A、固定分区存储管理 B、单一连续区存储管理
C、可变分区存储管理 D、段式存储管理
15. 通常以下哪种分区分配算法产生的外零头最小（ B ）
- A、首次适应 B、最佳适应
C、最坏适应 D、下次适应
16. 在段页式存储管理系统中，当访问主存中的一条指令或数据时（ C ）
- A、需访问两次主存 B、需访问一次主存
C、至少访问三次主存 D、至少访问两次主存
17. 在页式存储管理方案中，进行主存分配的单位是（ B ）
- A、段 B、块 C、作业 D、不一定
18. 在请求分页存储管理的页面置换策略中，会产生贝莱迪异态的算法是（ A ）
- A、FIFO B、最佳置换 C、LRU D、最坏适应
19. 动态重定位是在（ B ）进行的重定位。
- A、作业执行前 B、作业执行过程中 C、作业装入过程中 D、A，B，C均不对
20. 可以实现虚拟存储器的方案是___D___。
- A、固定分区方式 B、可变分区方式 C、纯分页方式 D、请求页式
21. 程序访问的局部性原理决定应使用__C__。

A、中断 B、DMA C、高速缓存 D、虚拟存储器

22. 可变式分区管理中存在一些小而无用的分区, 称做__A__。

A、外零头 B、内零头 C、页表零头 D、页内零头

23. 操作系统中的工作集模型与__C__有关。

A、合并存储区中的空白块 B、将 CPU 分配给进程

C、一个进程访问的页面集合 D、为进程分配 I/O 资源

24. 在一个可变分区存储管理中, 最佳适应算法是将空闲区表中的空闲区按__C__的次序排列。

A、地址递增 B、地址递减 C、大小递增 D、大小递减

25. 在 UNIX 系统中, 对换空间的管理采用得是__A__适应算法。

A、首次 B、最佳 C、最坏 D、下次

26. 内存分配的主要任务是为每道程序分配 A, 具体实现的方法有 B 与 C 两种方式, 对于 C 方———

法, 作业装入内存后不再申请新的空间; B 方法容许作业在内存中移动位置, 并采用 D 重定位技———
术, 在可变分区管理中, 借助于 E 进行重定位, 而在段式管理中则借助于 F 进行地址变换。 ——

选择答案:

(1) 动态 (2) 静态 (3) 段表 (4) 页表 (5) 部分装入 (6) 基地址寄存器

(7) 地址空间 (8) 外存空间 (9) 全部装入 (10) 动态连接 (11) 虚地址寄存器

(12) 物理地址寄存器

27. 在具有对换功能的操作系统中, 通常把外存分为

A B C D E F

答

7 5 6 1 6 3

文件区和对换区, 对换功能由 A 来实现。对文件区

案 ——

的存贮空间分配常采用 B 方式; 而对对换区的分配采——

用 C, 分配的基本单位是 D。 ——

选择答案:

(1) 高级调度 (2) 中级调度 (3) 低级调度 (4) 记录 (5) 页面 (6) 盘块 (7) 离散分配 (8) 连

续分配

28. 请求分段存贮管理系统中, 共享段 SEG 不在内存, 进程 A、

A B C D

答

2 7 8 6

B 执行中同时共享 SEG 段。设 A 先访问 SEG 段, B 在 A 后访问 SEG

案

段, 对下面给出的语句重新排序为: A、B、C、D、E、F、G、H、

I, 描述系统对 SEG 段进行内存分配的过程。

① B 执行访问 SEG 段, 产生缺段中断。

② 填写共享段表已分配的表项: SEG 段的说明、共享计数为 1、进程 A 说明。

③ 中断处理程序查共享段表, 发现 SEG 段已在内存。

④ 共享段表中的 SEG 段内存地址复制到 B 进程段表, 状态位改为 1。

⑤ 中断处理程序查共享段表, 分配一个空闲表项。

⑥ 共享段表中的 SEG 段内存地址复制到 A 进程段表，状态位改为 1。

⑦ 填写共享段表中的共享计数为 2、进程 B 的说明。

⑧ A 执行，访问 SEG 段进行地址变换时硬件产生缺段中断。

⑨ 给 SEG 段分配内存，SEG 段调入内存。

ABCDEFGHI

29. MS-DOS 操作系统采用了 A 内

答 ——

5 9 6 2 1 3 4 7

⑧

存管理方案和 B 重定位技术，每个

案 —

段在内存中 C。 ——

选择答案：

(1) 可以移动 (2) 不可以移动 (3) 静态 (4) 动态 (5) 页式

(6) 段式 (7) 四重分区 (8) 固定分区

ABC

答

8 3 2

案

4.2 填空题

1. 分页系统的页长为 1KB，虚拟地址 0x3C8F 对应的页号为 15，页内地址为 0x8C。 ——

2. 可变分区 管理是在作业装入和处理过程中，根据作业的实际需要动态地划分存

页号 页帧号 ——

储空间的。

0 3

3. 在一个分页存储管理的系统中，页长为 4KB，某一作业的页表如右所示，虚拟地

1 4

址 3000 对应物理地址 0x3BB8，12000 对应 0x6EE0。

2 6 ——

4. 地址空间是 逻辑 地址的集合，存储空间是 物理 地址的集合。 ——

5. 系统中有 4MB 内存，最大有效物理地址为 4MB-1，如果采用分页管理，页长 1KB，全部内存可

——

分为 4096 页帧。 ——

6. 所谓时间的局部性是指 程序即将用到的信息可能就是目前正在使用的信息。

——

7. 程序的空间局部性是指 程序即将用到的信息可能与目前正在使用的信息在空间上相邻或者临近。

——

32

8. 虚空间的大小取决于 计算机的寻址范围，如 32 位机的虚空间大小为 2=4GB。

——

9. 解决外零头的办法有 紧凑技术、 。 _____
10. 解决小内存大作业的方法有 覆盖技术、 交换技术、 请求式分页管理、 请求式分段管理以及请求式段页式管理。 _____
11. 所谓静态重定位是指 由动态重定位装配程序在程序装入时一次完成地址重定位 以后不再进行重定位。 —
12. 存储分配的三种方式 分区管理、 分页管理、 和分段管理。 _____
13. 覆盖是用于解决 大作业不能一次全部装入内存而引起的大作业与小内存之间 的问题。 _____
14. 在存储分配时，产生外零头的主要原因为 作业在装入时占有一个不可分割的主存空间，而且作 _____
业要求一次性地全部装入内存。 _____
15. 在请求式分页系统中，块的极小数取决于 工作集的大小。 _____
16. 页面置换算法分为 公平算法， 不公平算法 两大类。 _____
17. 段页式存储管理中访问主存中的一条指令或存取数据，则至少需访问内存 3 次 _____
18. 根据地址空间结构的不同，虚拟存储器有 页 式虚存和 段 式虚存两种形式。 _____
19. 可重定位分区分配是通过 紧凑过程 解决零头的。 _____
20. 已知一个三页长的进程具有页号 0、1、2，分别对应内存页面号为 2、3、6，页面大小为 1K，其中某一指令的虚地址为 1000，则对应的物理内存地址为 9192。 _____
21. 存储分配策略有 分区分配、 分页分配 和 分段分配 三种。 _____
22. 如果一个进程不能获得足够的块容纳它的工作集，那么将会发生 Belady 现象。 _____
23. 一个逻辑地址 32 个比特位采用请求页式管理、页长为 16KB 的计算机系统，其用户地址空间可达 4096 MB；页表长度为 256（十进制）KB；若处理器给出的逻辑地址为 9BCD（十六进制），送内 _____
存地址变换机构，地址变换机构将分离出页号为 2（十进制），如果该页所对应的物理块号为 111（十 _____
进制），则 9BCD 所对应的物理地址为： 1BDBCD（十六进制）。 _____
24. 在采用段式内存管理系统中，处理器给出的有效地址为 16 个比特位，系统允许的最大段长为 8KB，系统的地址空间可达 64（十进制）KB，地址空间中允许段的最大数量 8（十进制）。CPU 给出 _____
的有效地址为 9BCD（十六进制），则该有效地址对应的段号 4（十进制）。 _____

4.3 判断题

1. 动态分页管理中，对任一种页面置换算法，分配给一个进程的页帧数越多，发生缺页的次数越少。错
2. 请求调页的动态分页系统要求 CPU 的缺页中断支持。对
3. 使用全局置换算法，程序不可控制自身的缺页中断率。对
4. 用户在编程时直接使用物理地址的存储分配方式为静态方式。错
5. 在采用动态重定位的系统中已装入内存的作业，在其执行的过程中无需再进行地址转换工作。错
6. 在请求式分页系统中，缺页的中断率与程序结构无关。错
7. 一个作业的缺页中断率与置换算法无关。错
8. 当发生缺页中断时必须从内存中淘汰一页。错

9. 使用交换技术可使总存储空间需求大于实际存储空间的多个程序投入运行，所以说交换实现了虚拟存储器。错（因为要求作业全部装入）
10. 在请求分页系统中，如发现某页修改后，则该页不换出。错

4.4 简答题

1. 存储管理的实质是什么？存储管理的主要功能是什么？

答：内存管理就是对内存架构进行管理，使程序在内存架构的任何一个层次上的存放对于用户来说都是一样的。用户无需担心自己的程序是存在缓存、主存、磁盘还是磁带，反正运行、计算、输出的结果都是一样的。内存管理要达到的目标有：

○ 地址保护，一个程序不能访问另一个程序地址空间；

1

○ 地址独立：程序发出的地址应与物理主存地址无关。

2

存储管理的主要功能包括以下几点：

○ 在硬件的支持下完成统一管理内存和外存之间数据和程序段自动交换的虚拟存储；

1

○ 将多个虚存的一维线性空间或多维线性空间变换到内存的唯一的一维物理地址空间；

2

○ 控制内外存之间的数据传输；

3

○ 实现内存的分配和回收；

4

○ 实现内存信息的共享与保护。

5

2. 什么是虚拟存储器？其特点是什么？

答：由进程中的目标代码、数据等的虚拟地址组成的虚拟空间称为虚拟存储器。虚拟存储器不考虑物理存储器的大小和信息存放的实际位置，只规定每个进程中相互关联信息的相对位置。每个进程都拥有自己的虚拟存储器，且虚拟存储器的容量是由计算机的地址结构和寻址方式来确定。虚拟存储器就是要提供一个空间像磁盘那样大、速度像缓存那样快的主存储系统。

实现虚拟存储器要求有相应的地址转换机构，以便把指令的虚拟地址变换为实际物理地址；另外，由于内存空间较小，进程只有部分内容存放于内存中，待执行时根据需要再调指令入内存。

3. 实现地址重定位的方法有哪几类？

答：实现地址重定位的方法有两种：静态地址重定位和动态地址重定位。

(1)静态地址重定位是在虚空间程序执行之前由装配程序完成地址映射工作。静态重定位的优点是不需要硬件支持，但是用静态地址重定位方法进行地址变换无法实现虚拟存储器。静态重定位的另一个缺点是必须占用连续的内存空间和难以做到程序和数据的共享。

(2)动态地址重定位是在程序执行过程中，在 CPU 访问内存之前由硬件地址变换机构将要访问的程序或数据地址转换成内存地址。动态地址重定位的主要优点有：①可以对内存进行非连续分配。②动态重定位提供了实现虚拟存储器的基础。③动态重定位有利于程序段的共享。

4. 常用的内存信息保护方法有哪几种？它们各自的特点是什么？

答：常用的内存保护方法有硬件法、软件法和软硬件结合保护法三种。

上下界保护法是一种常用的硬件保护法。上下界存储保护技术要求为每个进程设置对上下界寄存器。上下界寄存器中装有被保护程序和数据段的起始地址和终止地址。在程序执行过程中，在对内存进行访问操作时首先进行访问地址合法性检查，即检查经过重定位之后的内存地址是否在上、下界寄存器所规定

的范围之内。若在规定的范围之内,则访问是合法的;否则是非法的,并产生访问越界中断。

保护键法也是一种常用的软件存储保护法。保护键法为每一个被保护存储块分配一个单独的保护键。在程序状态字中则设置相应的保护键开关字段,对不同的进程赋予不同的开关代码以和被保护的存储块中的保护键匹配。保护键可以没臂成对读写同时保护的或只对读写进行单项保护的。如果开关字段与保护键匹配或存储块未受到保护,则访问该存储块是允许的,否则将产生访问出错中断。

另外一种常用的硬软件内存保护方式是:界限存储器与 CPU 的用户态,核心态相结合的保护方式。在这种保护方式下,用户态进程只能访问那些在界限寄存器所规定范围内的内存部分,而核心态进程则可以访问整个内存地址空间。

5. 如果把 DOS 的执行模式改为保护模式,起码应做怎样的修改?

答:如果要把 DOS 的执行模式改成保护模式,起码要为每一个进程设置一对上下界寄存器。上下界寄存器中装有被保护程序和数据段的起始地址和终止地址。在程序执行过程中,在对内存进行访问操作时首先进行访问地址合法性检查,即检查经过重定位之后的内存地址是否在上、下界寄存器所规定的范围之内。若在规定的范围之内,则访问是合法的;否则是非法的,并产生访问越界中断。另外,还应该把指令的访问内存模式由访问实际物理地址改为由逻辑地址变换为物理地址的方式。

6. 动态分区式管理的常用内存分配算法有哪几种?比较它们各自的优缺点。

答:动态分区式管理的常用内存分配算法有最先适应法(FI)、最佳适应法(BF)和最坏适应法(WF)。

优缺点比较:

①从搜索速度上看最先适应法最佳,最佳适应法和最坏适应法都要求把不同大小的空闲区按大小进行排队。

②从回收过程来看,最先适应法也是最佳,因为最佳适应法和最坏适应法都必须重新调整空闲区的位置。

③最佳适应法找到的空闲区是最佳的,但是会造成内存碎片较多,影响了内存利用率,而最坏适应法的内存碎片最少,但是对内存的请求较多的进程有可能分配失败。

总之,三种算法各有所长,针对不同的请求队列,它们的效率和功能是不一样的。

8. 简述什么是覆盖?什么是交换?覆盖和交换的区别是什么?

答:将程序划分为若干个功能上相对独立的程序段,按照程序的逻辑结构让那些不会同时执行的程序段共享同一块内存区的内存扩充技术就是覆盖。交换是指先将内存某部分的程序或数据写入外存交换区,再从外存交换区中调入指定的程序或数据到内存中来,并让其执行的一种内存扩充技术。与覆盖技术相比,交换不要求程序员给出程序段之间的覆盖结构,而且,交换主要是在进程或作业之间进行,而覆盖则主要在同一个作业或同一个进程内进行。另外,覆盖只能覆盖那些与覆盖程序段无关的程序段。

9. 什么是页式管理?静态页式管理可以实现虚存吗?

答:页式管理就是把各进程的虚拟空间划分为若干长度相等的页面,把指令按页面大小划分后存放在内存中执行或只在内存中存放那些经常被执行或即将被执行的页面,而那些不被经常执行以及在近期内不可能被执行的页面则存放于外存中,按一定规则调入的一种内存管理方式。

静态页式管理不能实现虚存,这是因为静态页式管理要求进程或作业在执行前必须一次性地全部被装入内存,作业或进程的大小仍受内存可用页面数的限制。

10. 什么是请求页式管理?试设计和描述一个请求页式管理时的内存页面分配和回收算法(包括缺页处理部分)。

答:请求页式管理是动态页式内存管理的一种,它在作业或进程开始执行之前,不把作业或进程的程序段和数据段一次性的全部装入内存,而只装入被认为是经常反复执行和调用的工作区部分。其他部分则在执行过程中动态装入。

请求页式管理的调入方式可以是,当需要执行某条指令而又发现它不在内存时,或当执行某条指令需要访问其他数据或指令时,而这些指令和数据又不在内存中,从而发生缺页中断,系统将外存中相应的页面调入内存。

11. 请求页式管理中有哪几种常用的页面置换算法？试比较它们的优缺点。

答：比较常用的页面置换算法有：

(1)随机淘汰算法(random replacement algorithm)。即随机地选择某个用户页面并将其换出。

(2)轮转法 RR(round robin)。轮转法在内存可用区内循环地选择一个可以被换出的页，无论该页是刚被换进或已经换进内存很长时间。

(3)先进先出法 FIFO(first-in first-out)。FIFO 算法选择在内存驻留时间最长的一页将其淘汰。

(4)最近最久未使用页面置换算法 LRU(least recently unused)。该算法的基本思想是：当需要淘汰某一页时，选择离当前时间最近的一段时间内最久没有使用过的页面先淘汰。

(5)理想型淘汰算法 OPT(optimal replacement algorithm)。该算法淘汰在访问串中将来再也不出现的或是在离当前最远的位置上出现的页面。

12. 什么是段式管理？它与页式管理有何区别？

答：段式管理就是将程序按照内容或过程(函数)关系分成段，每段拥有自己的名字。一个用户作业或进程所包含的段对应于一个二维线性虚拟空间，也就是一个二维虚拟存储器。段式管理程序以段为单位分配内存，然后通过地址映射机构把段式虚拟地址转换成实际的内存物理地址。与页式管理一样，段式管理也采用只把那些经常访问的段驻留内存，而把那些在将来一段时间内不被访问的段放入外存，待需要时自动调入相关段的方法实现二维虚拟存储器。

段式管理和页式管理的主要区别有：

○ 页式管理中源程序进行编译链接时是将主程序、子程序、数据区等按照线性空间的一维地址顺序排列

1

起来。段式管理则是将程序按照内容或过程(函数)关系分成段，每段拥有自己的名字。一个用户作业或进程所包含的段对应于一个二维线性虚拟空间，也就是一个二维虚拟存储器。

○ 同动态页式管理一样，段式管理也提供了内外存统一管理的虚存实现。与页式管理不同的是：段式虚

2

存每次交换的是一段有意义的信息，而不是像页式虚存管理那样只交换固定大小的页，从而需要多次的缺页中断才能把所需信息完整地调入内存。

○ 在段式管理中，段长可根据需要动态增长。这对那些需要不断增加或改变新数据或子程序的段来说，

3

将是非常有好处的。

○ 段式管理便于对具有完整逻辑功能的信息段进行共享。

4

○ 段式管理便于进行动态链接，而页式管理进行动态链接的过程非常复杂。

5

13. 段式管理可以实现虚存吗？如果可以，简述实现方法。

答：段式管理可以实现虚存。

段式管理把程序按照内容或过程(函数)关系分成段，每段拥有自己的名字。一个用户作业或进程所包含的段对应于一个二维线性虚拟空间(段号 s ，段内相对地址 w)，也就是一个二维虚拟存储器。段式管理以段为单位分配内存，然后通过地址映射机构把段式虚拟地址转换成实际的内存物理地址。只把那些经常访问的段驻留内存，而把那些在将来一段时间内不被访问的段放入外存，待需要时产生缺段中断，自动调入。

14. 为什么要提出段页式管理？它与段式管理及页式管理有何区别？

答：因为段式管理和页式管理各有所长。段式管理为用户提供了一个二维的虚拟地址空间，反映程序的逻辑结构，有利于段的动态增长以及共享和内存保护等，这极大地方便了用户。而分页系统则有效地克服了碎片，提高了存储器的利用效率。从存储管理的目的来讲，主要是方便用户的程序设计和提高内存的利用率。所以人们提出了将段式管理和页式管理结合起来让其互相取长补短的段页式管理。

段页式管理与段式和页式管理相比，其访问时间较长。因此，执行效率低。

15. 为什么说段页式管理时的虚拟地址仍是二维的？

答：因为在段页式内存管理中，对每一段内的地址空间进行分页式管理只是为了克服在内存分配过程中产生的大量碎片，从而提高存储器的利用效率，它并没有改变段内地址空间的一维结构，所以段页式内存管理中的虚拟地址仍然和段式内存管理中的虚拟地址一样，是二维结构的。

16. 段页式管理的主要缺点是什么？有什么改进办法？

答：段页式管理的主要缺点是对内存中指令或数据进行存取时，至少需要对内存进行三次以上的访问。第一次是由段表地址寄存器取段表始址后访问段表，由此取出对应段的页表在内存中的地址。第二次则是访问页表得到所要访问的指令或数据的物理地址。只有在访问了段表和页表之后，第三次才能访问真正需要访问的物理单元。显然，这将大大降低 CPU 执行指令的速度。

改进办法是设置快速联想寄存器。在快速联想寄存器中存放当前最常用的段号 s ，页号 p 和对应的内存页面地址与其他控制项。当需要访问内存空间某一单元时，可在通过段表、页表进行内存地址查找的同时，根据快速联想寄存器查找其段号和页号。如果所要访问的段或页的地址在快速联想寄存器中，则系统不再访问内存中的段表、页表而直接把快速联想寄存器中的值与页内相对地址 d 拼接起来得到内存地址。

17. 什么是局部性原理？什么是抖动？你有什么办法减少系统的抖动现象？

答：局部性原理是指在几乎所有程序的执行过程中，在一段时间内，CPU 总是集中地访问程序中的某一个部分而不是对程序的所有部分具有平均的访问概率。抖动是指当给进程分配的内存小于所要求的工作区时，由于内存外存之间交换频繁，访问外存的时间和输入输出处理时间大大增加，反而造成 CPU 因等待数据而空转，使得整个系统性能大大下降。在物理系统中，为防止抖动的产生，在进行淘汰或替换时，一般总是把缺页进程锁住，不使其换出，从而防止抖动发生。

18. 什么叫物理地址？什么叫逻辑地址？什么叫地址映射？地址映射分哪几类？(静态、动态)

答：物理地址是内存中各存储单元的编号，即存储单元的真实地址，它是可识别、可寻址并实际存在的。

用户程序经过编译或汇编形成的目标代码，通常采用相对地址形式，其首地址为零，其余指令中的地址都是相对首地址而定。这个相对地址就称为逻辑地址或虚拟地址。逻辑地址不是内存中的物理地址，不能根据逻辑地址到内存中存取信息。

为了保证 CPU 执行程序指令时能正确访问存储单元，需要将用户程序中的逻辑地址转运行时可由机器直接寻址的物理地址，这一过程称为地址映射或地址重定位。

地址映射可分为两类：○ 静态地址映射；○ 动态地址映射。

12

19. 怎样对内存进行分区？(静态、动态；等长、不等长)

答：对内存空间的划分是可以静态的，也可以动态的；可以是等长的，也可以不等长。

静态划分是指系统运行之前就将内存空间划分成若干区域，通常，分配给进程的内存可能比进程实际所需的区域长。

动态划分是在系统运行过程中才划分内存空间。这样，系统可按进程所需要的存储空间大小为其分配恰好满足要求的一个或多个区域。

等长分区是将存储空间划分为若干个长度相同的区域。

不等长分区则是将存储空间划分若干个长度不同的区域。

20. 影响缺页中断率有哪几个主要因素？

答：影响缺页中断率的因素有四个：

①一般说来，分配给作业的主存块数多则缺页率低，反之缺页中断率就高。

②页面大，缺页中断率低；页面小缺页中断率高。

③程序编制方法。以数组运算为例，如果每一行元素存放在一页中，则按行处理各元素缺页中断率低；

反之，按列处理各元素，则缺页中断率高。

④页面调度算法对缺页中断率影响很大，但不可能找到一种最佳算法。

21. 什么叫碎片？怎样解决碎片问题？

答：所谓碎片是指内存中出现的一些零散的小空闲区域。

解决碎片的方法是移动所有占用区域，使所有的空闲区合并成一片连续区域。这一过程称为紧凑，这一技术就是紧凑技术。

22. 分区管理的基本思想是什么？主要缺点是什么？

答：基本思想：将内存划分成若干连续的区域，称为分区，每个分区装入一个运行作业。

主要缺点：不能充分利用内存，也不能实现对内存的扩充。

23. 什么是固定分区？什么是可变分区？各有什么优缺点？

答：固定分区：系统将内存划分为若干固定的分区，当作业申请内存时，系统为其选择一个适当的分区，并装入内存运行。由于分区大小是事先固定的，因而可容纳作业的大小受到限制，而且当用户作业的地址空间小于分区的存储空间时，浪费了一些存储空间。

可变分区：是指在作业装入内存时建立分区，使分区的大小正好与作业要求的存储空间相等。引入可变分区方法，使内存分配有较大的灵活性，也提高了内存利用率。但是可变分区会引起碎片的产生。

24. 为了提高存取速度，可以使用快表技术。试述这一技术是如何实现的？

答：快表技术是在地址映射机构中增加一个小容量的联想寄存器（相联存储器），它由高速寄存器组成，成为一张快表，快表用来存放当前访问最频繁的少数活动页的页号。

在快表中，除了逻辑页号、物理页号对应外，还增加了几位。特征位表示该行是否为空，用 0 表示空，用 1 表示有内容；访问位表示该页是否被访问过，用 0 表示未访问，1 表示已访问，这是为了淘汰那些用得很少甚至不用的页面而设置的。

快表只存放当前进程最活跃的少数几页，随着进程的推进，快表内容动态更新。当用户程序需要存取数据时，根据该数据所在逻辑页号在快表中找出对应的物理页号，然后拼接页内地址，以形成物理地址；如果在快表中没有相应的逻辑页号，则地址映射仍然通过内存中的页表进行，得到物理页号后须将该物理页号填到快表的空闲单元中。有无空闲单元，则根据淘汰算法淘汰某一行，再填入新得到的页号。实际上查找快表和查找内存页表是并行进行的，一旦发现快表中有与所查页号一致的逻辑页号就停止查找内存页表。

25. 试述段页式存储管理的基本思想

答：段页式存储管理的基本思想是：

○ 用页式方法来分配和管理内存空间，即把内存划分成若干大小相等的页面；

1

○ 用段式方法对用户程序按照其内在的逻辑关系划分成若干段；

2

○ 再按照划分内存页面的大小，把每一段划分成若干大小相等的页面；

3

○ 用户程序的逻辑地址由三部分组成，形式如下：段号、页号、页内地址

4

○ 内存是以页为基本单位分配给每个用户程序的，在逻辑上相邻的页面内存不一定相邻。

5

26. 简述虚拟存储技术的理论基础。

答：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。

○ 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。

1

○ 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

27. 在虚存中，页面在内存与外存中频繁地调试，系统效率急剧下降，称为颠簸。试说明产生颠簸的原因。通过什么方式可以防止颠簸的发生？

答：颠簸是由缺页率高而引起的。系统规定缺页率的上界和下界。当运行进程缺页率高于上界时，表明所分给它的物理页面数过少，应当增加；反之，当运行进行缺页率低于下界时，表明所分给它的物理页面数过多，可以减少。这样，根据缺页率反馈可动态调整物理页面的分配，以防止颠簸的发生。

28. 说明动态分页系统中的“抖动”现象及解决策略。

答：在更换页面时，如果更换的页面是一个很快就会被再次访问的页面，则在此缺页中断后很快又会发生新的缺页中断。整个系统的效率急剧下降，这种现象称为抖动（Trashing）。

内存抖动的解决策略主要有：

○ 如果是因为页面替换策略失误，可以修改替换算法来解决这个问题。

1

○ 如果是因为运行的程序太多，造成至少严格程序无法同时将所有频繁访问的页面调入内存，则要降低

2

多道程序的数量。

○ 否则，只能采取的办法只有 2 个：一是终止该进程；一是增加物理内存容量。

3

29. 操作系统在内存中的位置是怎样的？

答：根据操作系统是否占用 ROM 或是否采用内存映射的输入输出可以分为：

（1）操作系统占用 RAM 的底层，用户程序占用 RAM 的上层；

（2）操作系统占用 RAM 的底层和位于用户程序地址空间上面的 ROM，用户程序位于中间。其又可以细

分为：

○ 没有使用内存映射的输入输出，ROM 里

a

面全部是操作系统；

○ 使用了内存映射的输入输出，ROM 的一

b

部分是操作系统，另一部分属于 I/O 设备；

○ 使用了内存映射的输入输出，ROM 全部

c

属于 I/O 设备。

30. 画出分页系统地址变换结构图。若 CPU

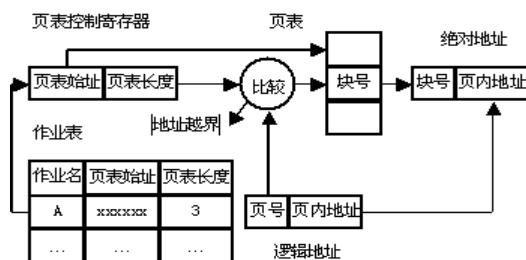
访问内存的时间为 100ns，检索快表的时间

为 20ns，若访问的页面不在快表中，则 CPU

存取一个数据为多少 ns？

答：若访问的页面不在快表中，则需要访问

2 次内存：一次是查页表，生成物理地址；一次是根据物理地址，访问内存。因此耗费的时间为：



$100 \times 2 = 200\text{ns}$ 。

31. 设有一个 32 位寻址的分页系统，页面大小为 16KB，假定页面号处于最左面，页内偏移量处于最右面，请问系统需要多少位来表示页面号和页内偏移？该系统能访问的最大虚拟页面号是多少？

14

$16\text{KB} = 2^4\text{B}$

答：由于，因此该分页系统的页内偏移占 14 位，所以系统需要 $32 - 14 = 18$ 位表示页面号，14 位表示页内偏移。

18

$2^{18} = 256\text{K}$

该系统能访问的最大虚拟页面号 = 。

32. 何谓比莱迪异常？如何解决该异常现象。

答：通常情况下，内存抖动是可以通过降低多道编程的度数来解决，这是因为减少同时运行的程序个数，就能使每个程序所占有的物理页面数增加。但是，如果增加物理页面数反而导致缺页次数增加的现象就称为比莱迪异常（Belady's anomaly）。

由于比莱迪异常是一种异常现象，而不是一种正常现象，因此如果在发生了比莱迪异常时，仍然是增加物理页面数，直到这种现象消失为止。

33. 什么是驱动分页系统出现的关键动机？

答：分区管理存在着如下 2 个重要问题：一是空间浪费问题。外部碎片浪费了大量的内存空间，虽然紧凑技术能完成碎片整理，但在碎片整理过程中，系统的响应时间将显著增加；二是程序受限问题。主要表现在：空间增长效率低下；空间增长存在“天花板”限制。

究其原因，○ 空间浪费，主要是每个程序的大小不一样，因此在空间分配时存在着不一致性，将空间按

1

照某种规定的大小进行分配即可解决该问题。

○ 程序增长受限，主要是要求程序一次性地全部装入到内存的前提导致的，因此放宽该条件即可，仅将

2

当前需要的页面放在内存，其他暂时不用的页面放在磁盘上，也就是部分装入。

34. 何谓缺页中断？缺页中断的处理步骤有哪些？

答：在分页系统中，如果 CPU 发出的虚拟地址所对应的页面不在物理内存，系统将产生中断，称这种中断为缺页中断。缺页中断服务程序负责将需要的虚拟页面找到并加载到内存。其主要步骤如下：

○ 硬件陷入到内核；

1

○ 保护通用寄存器；

2

○ 操作系统判断所需的虚拟页面号；

3

○ 操作系统检查地址的合法性；

4

○ 操作系统选择一个物理页面用来存放将要调入的页面；

5

○ 如果选择的物理页面包含有未写入磁盘的内容，则首先进行写盘操作；

6

○ 操作系统将新的虚拟页面调入内存；

7

○ 更新页表；

8

○ 发生缺页中断的程序进入就绪状态；

9

○ 恢复寄存器；

10

○ 程序继续运行。

11

4.5 综合题

1. 在请求调页的动态分页系统中，一个程序的页面走向为：2，4，8，3，2，4，5，2，4，8，3，5。
如果分配给此程序的页帧数为4，分别分析采用 FIFO、LRU 和最佳置换策略时的置换过程并计算缺页次数。

答：3 种不同的页面置换算法运行情况如下：

页面走向 2 4 8 3 2 4 5 2 4 8 3 5

2 2 2 2 2 2 4 8 3 5 2 4

4 4 4 4 4 8 3 5 2 4 8

FIFO

8 8 8 8 3 5 2 4 8 3

3 3 3 5 2 4 8 3 5

缺页标记 1 1 1 1 0 0 1 1 1 1 1 1 10

页面走向 2 4 8 3 2 4 5 2 4 8 3 5

2 2 2 2 4 8 3 3 3 5 2 4

4 4 4 8 3 2 4 5 2 4 8

LRU

8 8 3 2 4 5 2 4 8 3

3 2 4 5 2 4 8 3 5

缺页标记 1 1 1 1 0 0 1 0 0 1 1 1 8

页面走向 2 4 8 3 2 4 5 2 4 8 3 5

2 2 2 2 2 2 2 2 2 2 3 3

4 4 4 4 4 4 4 4 4 4 4

OPT

8 8 8 8 8 8 8 8 8 8

3 3 3 5 5 5 5 5 5

缺页标记 1 1 1 1 0 0 1 0 0 0 1 0 6

2. 分页式存储空间的分配由于块的大小是固定的，可以用一张位示图来构成主存分配表。现设主存有 8192 块，则可用字长为 32 位的 256 个字作为位示图。若块号、字号、位号（从高位到低位）都是从 0 开始，试问 4999 块对应的字号和位号；129 字的 29 位对应哪一块？

答：依题目所给条件，已知位示图如下所示：

0 1 2 31

...				

0 0 1 2 31

...

1 32 33 34

... ...?

2

... ..

... ..

255 8191

...

4999÷32=156，余 1。所以 4999 块对应的字号为 156，位号为 1。

129 字的 29 位对应的块号为：129*32+29=4157（块），即对应内存的第 4157 块。

3. 有一个程序要将 128×128 的数组置初值“0”。现假定分给这个程序的主存块数只有一块，页面的尺寸为每页 128 个字，数组中的元素每一行存放在一页中，开始时第一页在主存。若程序如下编制：

○ var A: array[1..128] of array [1..128] of integer;

1

for j := 1 to 128 do

```

for i := 1 to 128 do
  A[i][j] := 0
○ var A: array[1..128] of array[1..128] of integer;
2
for j := 1 to 128 do
  for i := 1 to 128 do
    A[i][j] := 0

```

这两种方法的缺页中断次数分别是多少？

答：○ 每执行一次 $A[i][j] := 0$ 就要产生一次缺页中断，于是总共要产生 $(128 \times 128 - 1)$ 次缺页中断。

1
○ 每访问一行数据才产生一次，因此总共只产生 $(128 - 1)$ 次缺页中断。

2
4. 在请求分页存储系统中，一个程序的页面走向为：4, 3, 1, 2, 5, 3, 4, 2, 3, 4, 1, 5, 3, 2, 4 并采用 LRU 页面置换算法，当分配给该程序的存储块数分别为 3 和 4 时，试求出在访问过程中发生缺页中断的次数，并比较两种结果，说明从中可以得到什么启示？

答：页框数为 3 时的页式管理示意图：

页面走向 4 3 1 2 5 3 4 2 3 4 1 5 3 2 4

4 4 4 3 1 2 5 1 4 2 3 4 1 5 3

累

计

LRU 3 3 1 2 5 1 4 2 3 4 1 5 3 2

1 2 5 1 4 2 3 4 1 5 3 2 4

缺页标记 1 1 1 1 1 1 1 1 0 1 1 1 1 1 14

页框数为 4 时的页式管理示意图：

页面走向 4 3 1 2 5 3 4 2 3 4 1 5 3 2 4

4 4 4 4 3 1 2 5 5 5 2 3 4 1 5

累

3 3 3 1 2 5 3 4 2 3 4 1 5 3

计

LRU

1 1 2 5 3 4 2 3 4 1 5 3 2

2 5 3 4 2 3 4 1 5 3 2 4

缺页标记 1 1 1 1 1 0 1 0 0 0 1 1 0 1 1 10

从上表中，可以得到：增加页框数量，能减少缺页中断次数。

5. 在某虚拟页面管理系统中。用户编程地址空间为 64K，物理空间为 32K，页面大小为 4K，某时刻页表内容如下：（注：状态位为 1 表示该页在内存，为 0 则不在。）

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

页号

2 1 6 0 4 3 - - - - 5 - - 7 - - - - -

块号

1 1 1 1 1 1 0 0 0 1 0 1 0 0 0 0

状态位

问：○ CPU 给出有效地址是多少位？地址变换机构（内存管理部件 MMU）给出的物理地址是多少位？

1

○ 虚地址：（1）5587h 对应的物理地址是多少（用十六进制表示）、（2）16400 对应的物理地址为多少

2

（用十进制表示）、（3）E253h 对应的物理地址为多少（用十六进制表示）？如访问的页面不再主存，注明页失效。

答：○ CPU 给出有效地址是 16 位。地址变换机构（内存管理部件 MMU）给出的物理地址是 15 位。

1

○ 虚地址：

2

（1）5587h 对应的物理地址是 0x3587。0x5587 的页号为 5，对应的块号为 3，因此物理地址为 0x3587。

（2）16400 对应的物理地址为 0x4010。16400=0x4010，其页号为 4，对应的块号为 4，因此物理地址也为 0x4010，即 16400。

（3）E253h 对应的物理地址是没有的，因为该页还没有装入到主存，发生页失效错误。

6. 某进程，若它对页面的访问串为：7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 试用 LRU、FIFO 两种算法实现页面更换，并给出各自的缺页次数。•••（设该进程在内存中占四个页架）

答：M=4 时，采用 LRU 算法，系统的淘汰过程：

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

页面走向

* * * * *

缺页标记

7 7 7 7 7 3 3 3 3 3 3 3 3 3 3 7 7

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

LRU

1 1 1 1 1 4 4 4 4 4 4 1 4 4 1 1 1

2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

淘汰页面

7 1 4 4 3

即 $F=9$ (次缺页)

采用 FIFO 算法，系统的淘汰过程：

[illegible]

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

页面走向

* * * * *

缺页标记

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

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

FIFO

70011222333400011

770 0 1 1 1 2 2 2 3 4 4 4 0 0

淘汰页面

7 0 1 23 4

即 $F=10$ (次缺页)

7. 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 与 LRU 页面调度算法, 列出各自的页面淘汰顺序和缺页中断次数, 以及最后留驻主存 4 页的顺序。(假设开始的 4 个页面已装入主存)

[illegible]

答:

页面走向 1 2 3 6 4 7 3 2 1 4 7 5 6 5 2 1

1 1 1 1 2 3 3 6 4 4 4 7 2 2 2 2

2 2 2 3 6 6 4 7 7 7 2 1 1 1 1

FIFO 3 3 6 4 4 7 2 2 2 1 5 5 5 5

6 4 7 7 2 1 1 1 5 6 6 6 6

6

??????

1 1 1 1 2 3 6 4 7 3 2 1 4 4 7 6

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

LRU 3 3 6 4 7 3 2 1 4 7 5 6 5 2

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

10

??????

最后驻留在系统中的四页均为 6, 5, 2, 1。

8. 分页系统中页面尺寸应该设计为多大?

答: 假定 p 表示页面大小, e 表示页表一个记录的大小, s 表示程序的平均尺寸, 则整个系统浪费的空间

$s*ep$

+

为: , 其中第一项为程序页表所占的内存, 第二项为程序平均浪费的页面空间。 ——

$p2\sqrt{}$

$p=2se$

对上式求极小值, 得到页面尺寸的最优大小为:。

第五章 设备管理

5.1 选择题

1. 哪种设备属于字符设备? D

A. 磁盘 B. 磁带 C. 光盘 D. 键盘

2. 在移臂调度时读写头从盘的一端开始朝另一端移动, 在移动的过程中搜索每个磁道上的请求, 若

有则服务之, 直至到达盘的另一端。在另一端, 磁头移动的方向是相反的, 并继续在移动中扫描服务, 则此种算法称为: C

A、先来先服务 B、最短查找时间优先

C、SCAN D、C-SCAN

3. 在设备分配中，独占分配方式的主要缺点是__A_____。

A、设备利用率低 B、设备利用率高 C、管理复杂 D、可使设备并行工作

4. UNIX 系统把设备分为__B__。

A、输入设备和输出设备 B、字符设备和块设备
C、系统设备和用户设备 D、共享设备和虚拟设备

5. 哪种设备属于块设备？B

A. 键盘 B. 磁盘 C. 显示器 D. 打印机

6. 在有通道支持的系统中，设备驱动程序根据 I/O 请求组织 A，然后驱动 B。由通道向 C 发出 I/O_____

命令，控制设备完成制定的操作。如果请求者进程已 D，CPU 响应通道发来的中断请求，由 IO 程序把_____

该进程 E。——

选择答案：

(1) 通道 (2) 阻塞 (3) 撤销 (4) 唤醒 (5) 输出文件 (6) 通道程序 (7) 设备 (8) 设备控制

器 (9) I/O 文件

A B C D E

7. 计算进程请求处理一个磁盘文件，系统输入进程通过单缓

答

7 8 2 4

6

冲 buffer 和中断处理程序把文件读入内存，流程图如下，请

案

填入 P、V 原语实现正确的同步操作，信号量 S 的初值为 A。

(输入程序)

开始

根据目录查到文件首物理块

LOOP: 启动磁盘机

B

内存 buffer 内容送计算机程序数据区

文件输入完否？否，转 LOOP

唤醒计算进程

输入进程自行阻塞

(中断处理程序)

入口

输入正确否？否，转 NEXT

C

NEXT: 恢复被中断操作进程现场

返回

选择答案：

(1) P(S) (2) V(S) (3) S 的初值为 1 (4) S 的初值为 0

A B C

8. 系统中有一台由某分配性通道支持的磁盘机，在通道与请求者进
答

1 2

4

程之间只设置了一个磁盘驱动程序来完成请求者进程与设备之间的通
案

信。假设请求者进程首次请求读某逻辑文件的第 n 号记录，请把下列
语句进行重新排列，简要描述从请求到完成所经历的如下 5 个步骤：

1) 请求者进程发出对文件第 n 号记录请求

2) 磁盘驱动进程运行

3) I/O 操作完成，CPU 响应通道发来的完成中断请求

4) 磁盘驱动进程运行

5) 请求者进程运行

注意：请从下列语句中挑选合适的语句描述以上的 5 个步骤：

1) 组织通道程序

2) 申请分配输入缓冲区

3) 唤醒磁盘驱动进程

4) 请求者把 I/O 参数通知磁盘驱动进程

5) 阻塞请求者进程

6) 求逻辑记录 n 所在的物理块号

7) 根据物理块号获得三维物理地址（柱面号、磁道号、扇区号）

8) 驱动磁道与设备

9) 磁盘驱动进程自行阻塞

10) 磁盘唤醒驱动进程

11) 分析中断原因，进行中断处理

12) 返回被中断的进程继续执行

13) 把正常完成的信号通知磁盘驱动进程

14) 把输入缓冲区中的第 n 号记录分离出来并传送到请求者进程的数据区

15) 磁盘驱动进程自行阻塞，等待新的请求唤醒

16) 唤醒请求者进程

17) 对输入数据加工

4 5 3 1 2 6 7 9 8 11

9. 操作系统在__A__中采用

答

14 13 15 16 12 17

10

了以空间换时间的技术。

案

A、SPooling 技术 B、覆盖技

术 C、通道技术 D、虚拟存储技术

5.2 填空题

- 系统中有一组如右表所示（按照到达顺序）的磁盘 I/O 请求等待服务，假设当前磁进程 磁道
道为 100，刚完成对 88 道的操作，分别计算不同调度方法下的磁头移动总道数。先来 2 12
3 180
先服务： 382 ；最短寻找时间优先： 296 ；电梯式查找： 228 。 _____
1 120
- SPOOLING 系统中输入井是 模拟脱机输入时的磁盘，用于收容 I/O 设备输入的数据。
6 72 _____
据。
5 54 ____
- 影响磁盘读写时间的因素有 寻道时间、旋转时间和数据传输时间 。

- 按输入输出特性可将设备分为 输入型外围设备、输出型外围设备和存储型外围设备 。

- 通道程序的首地址放于 通道地址字（CAW）的主存固定单元中（大型机）/主存中的 CPU 与 I/O 处

- 任何一个对磁盘的访问请求，•应给出访问磁盘的存储空间地址，•其地址由 柱面号、磁道号和扇

- 区号 组成。 _____
- 从设备分配的观点看，可将设备分为 独占 设备和 共享 设备和虚拟设备三类。 _____
- 可以通过 虚拟技术 把原独享设备改造成能为若干用户共享的设备。 _____
- 在使用通道设备的系统中，设备分配的步骤为： 分配设备 ， 分配控制器， 分配通道 。

- 操作系统的设备管理应具备的主要功能 提供和进程管理系统的接口、进行设备分配、实现设备和

- 设备，设备和 CPU 等之间的并行操作、进行缓冲区管理。

- 缓冲区的设置可分为 单缓冲、双缓冲、多缓冲 和 缓冲池 。 _____
- 利用缓冲区能有效地缓和 外围设备 和 处理机 之间速度不匹配地矛盾，虚拟设备功能是为一个

- 物理设备 变成能被多个进程同时使用的 逻辑设备 。 _____
- 从资源分配的角度看，可以把设备分为独占设备和共享设备。打印机属于 独占 设备，而磁盘属于

- 共享 设备。 _____
- 虚拟设备是通过 SPOOLing 技术把 独占 设备变成能为若干用户 共享 的设备。 _____

- 通道是一个独立于 CPU 的专管的处理机，它控制 外围设备 与内存之间的信息交换。 _____

- 对磁盘上一物理块信息的访问要经过 寻找时间、延迟时间、传送时间 三个过程。

5.3 判断题

1. 利用 Spooling 技术可将一台独占设备虚拟为几台“虚拟”设备。对
2. I/O 操作是 CPU 执行通道程序完成的。错
3. 启动外设的工作必须在管态下由操作系统完成。对
4. 中断系统是由硬件和软件配合完成的。对

5.4 简答题

1. 设备可以按照何种方式分类，每种分类方式又包括哪些？

答：○ 按设备的工作特性分类：（1）存储设备；（2）输入输出设备；

1

○ 按设备上数据组织方式分类：（1）块设备；（2）字符设备；

2

○ 按资源分配的角度分类：（1）独占设备；（2）共享设备；（3）虚拟设备。

3

2. 设备管理的目标和功能是什么？

答：设备管理的目标：

○ 向用户提供外部设备的方便、统一的接口，按照用户的要求和设备的类型，控制设备工作，完成用户

1

的输入输出请求。

○ 充分利用中断技术、通道技术和缓冲技术，提高 CPU 与设备、设备与设备之间的并行工作能力，以充

2

分利用设备资源，提高外部设备的使用效率。

○ 设备管理就是要保证在多道程序环境下，当多个进程竞争使用设备时，按照一定的策略分配和管理设

3

备，以使系统能有条不紊地工作。

设备管理的功能：

○ 设备分配和回收；

1

○ 管理输入输出缓冲区；

2

○ 设备驱动，实现物理 I/O 操作；

3

○ 外部设备中断处理；

4

○ 虚拟设备及其实现。

5

3. 外部设备的输入输出方式有哪些？

答：外部设备的输入输出方式有：

○ 询问方式：又称程序直接控制方式，在这种方式下，输入输出指令或询问指令测试一台设备的忙闲标

1

志位，决定主存储器和外围设备是否交换一个字符或一个字。早期计算机和微机往往采用这种方式，中央处理机的大量时间用在等待输入输出的循环检测上，使主机不能充分发挥效率，外围设备也不能得到合理使用，整个系统的效率很低。

○ 中断方式：中断机构引入后，外围设备有了反映其状态的能力，仅当操作正常或异常结束时才中断中

2

央处理机。实现了一定程度的并行操作，这叫程序中断方式。由于输入输出操作直接由中央处理器控制，每传送一个字符或一个字，都要发生一次中断，因而仍然消耗大量中央处理器时间。若为外围设备增加缓冲寄存器存放数据，则可大大减少中断次数。中央处理器在外围设备与缓冲寄存器交换信息期间可执行其它指令。例如行式打印机、卡片机、字符显示器等均配置数据缓冲寄存器，提高了中央处理器和外围设备并行工作的程度。

○ DMA 方式：在直接主存存取方式中，I/O 控制器有更强的功能，它不仅设有中断机构，而且，还增加

3

了 DMA 控制机构。在 DMA 控制器的控制下，它采用‘偷窃’总线控制权的方法，让设备和主存之间成批交换数据，而不必由 CPU 干预。这样可减轻 CPU 的负担，因每次传送数据时，不必进入中断系统；只要 CPU 暂停几个周期，从而，使 I/O 数据的速度也大大提高。目前，在小型、微型机中的快速设备均采用这种方式，DMA 的操作全部由硬件实现，不影响 CPU 寄存器的状态。DMA 方式线路简单，价格低廉，但功能较差，不能满足复杂的 I/O 要求。因而，在中大型机中使用通道技术。

○ 通道技术：通道又称输入输出处理器。它能完成主存储器和外围设备之间的信息传送，与中央处理器

4

并行地执行操作。采用通道技术主要解决了输入输出操作的独立性和各部件工作的并行性。由通道管理和控制输入输出操作，大大减少了外围设备和中央处理器的逻辑联系。从而，把中央处理器从琐碎的输入输出操作中解放出来。此外，外围设备和中央处理器能实现并行操作；通道和通道之间能实现并行操作；各通道上的外围设备也能实现并行操作，以达到提高整个系统效率这一根本目的。

4. 简述通道及通道控制结构。

答：通道又称输入输出处理器。它能完成主存储器和外围设备之间的信息传送，与中央处理器并行地执行操作。采用通道技术主要解决了输入输出操作的独立性和各部件工作的并行性。由通道管理和控制输入输出操作，大大减少了外围设备和中央处理器的逻辑联系。上例中，中央处理器只要用一条启动指令，就可通知输入机输入 1000 个字符，一秒钟之后发生中断再去处理。从而，把中央处理器从琐碎的输入输出操作中解放出来。此外，外围设备和中央处理器能实现并行操作；通道和通道之间能实现并行操作；各通道上的外围设备也能实现并行操作，以达到提高整个系统效率这一根本目的。

在一般大型计算机系统中，主机对外部设备的控制可以分成三个层次来实现，即通道、控制器和设备。一旦 CPU 发出启动通道的指令，通道就可以独立于 CPU 工作。通道控制控制器工作，控制器用来控制设备的电路部分。这样，一个通道可以连接多个控制器，而一个控制器又可以连接若干台同类型的外部设备。最终，设备在控制器控制下执行操作。

5. 简述通道控制的设备采用何种连接方式？其优点是什么？

答：一般设备的连续采用交叉连接，其好处是：

○ 提高系统的可靠性：当某条通路因控制器或通道故障而断开时，可使用其他通路。

1

○ 提高设备的并行性：对于同一个设备，当与它相连的某一条通路中的控制器或通道被占用时，可以选

2

择另一条空闲通路，减少了设备因等待通路所需要花费的时间。

6. 通道按传送数据的工作方式可以分哪几类？

答：按照信息交换方式和加接设备种类不同，通道可分为三种类型：

○ 字节多路通道：它是为连接大量慢速外围设备，如软盘输入输出机、纸带输入输出机、卡片输入输出

1

机、控制台打字机等设置的。以字节为单位交叉地工作，当为一台设备传送一个字节后，立即转去为另一台设备传送一个字节。在 IBM370 系统中，这样的通道可接 256 台设备。

○ 选择通道：它用于连接磁带、磁鼓和磁盘快速设备。以成组方式工作，每次传送一批数据；故传送速

2

度很高，但在这段时间只能为一台设备服务。每当一个输入输出操作请求完成后，再选择与通道相连接的另一设备。

○ 数组多路通道：对于磁盘这样的外围设备，虽然传输信息很快‘但是移臂定位时间很长。如果按在字

3

节多路通道上，那么通道很难承受这样高的传输率；如果接在选择通道上，那么；磁盘臂移动所花费的较长时间内，通道只能空等。数组多路通道可以解决这个矛盾，它先为一台设备执行一条通道命令，然后自动转换，为另一台设备执行一条通道命令。对于连接在数组多路通道上的若干台磁盘机，可以启动它们同时进行移臂，查找欲访问的柱面，然后，按次序交叉传输一批批信息，这样就避免了移臂操作过长地占用通道。由于它在任一时刻只能为一台设备作数据传送服务，这类似于选择通道；但它不等整个通道程序执行结束就能执行另一设备的通道程序命令，它类似于字节多路通道。数组多路通道的实质是：对通道程序采用多道程序设计技术的硬件实现。

7. 设备分配的任务是什么？

答：设备分配的任务是按照一定的策略为申请设备的进程分配合适的设备、控制器和通道。

8. 设备分配应坚持的原则是什么？

设备的独立性：不能因物理设备的更换而影响用户程序的正常运行；

系统的安全性：设备分配不能导致死锁现象发生。

9. 什么是设备的独立性？根据设备的类型，设备的分配策略有哪些？(独占设备、共享设备、虚拟设备与 SPOOLing 系统)。以磁盘为例，有哪些优化调度算法？应考虑哪些因素？

答：进程申请设备时，应当指定所需设备的类别，而不是指定某一具体的设备，系统根据当前请求以及设备分配情况在相应类别的设备中选择一个空闲设备并将其分配给申请进程，这称作设备的独立性。

磁盘调度一般可采用以下几种算法：

○ 先来先服务磁盘调度算法（FCFS）

1

○ 最短寻道时间优先磁盘调度算法（SSTF）

2

○ 扫描算法（SCAN）

3

设计磁盘调度算法应考虑两个基本因素：

○ 公平性；○ 高效性

12

10. 为实现设备的有效管理，应采用怎样的数据结构？

答：为实现设备、控制器、通道资源的分配与回收，系统需要记录有关的信息。通常设备管理要建立以下数据结构，以实施有效的管理。

○ 设备控制块

1

○ 控制器控制块

2

○ 通道控制块

3

○ 系统设备表

4

11. 简述中断、陷阱、软中断之间的异同。

答：中断即外中断，指来自处理机和内存外部的中断，包括 I/O 设备发出的 I/O 中断、外部信号中断、各种定时器引起的时钟中断以及调试程序中设置的断点等引起的调试中断等。陷阱即内中断，主要指在处理机和内存内部产生的中断。它包括程序运算引起的各种错误。软中断是通信进程之间用来模拟硬中断的一种信号通信方式。

中断和陷阱的主要区别：

○ 陷阱通常由处理机正在执行的现行指令引起，而中断则是由与现行指令无关的中断源引起的。○ 陷阱

12

处理程序提供的服务为当前进程所用，而中断处理程序提供的服务则不是为了当前进程的。○ CPU 在执

3

行完一条指令之后，下一条指令开始之前响应中断，而在一条指令执行中也可以响应陷阱。○ 在有的系

4

统中，陷入处理程序被规定在各自的进程上下文中执行，而中断处理程序则在系统上下文中执行。

软中断与硬中断的比较：相同点：其中断源发中断请求或软中断信号后，CPU 或接收进程在适当的时机自动进行中断处理或完成软中断信号所对应的功能。不同点：接收软中断信号的进程不一定正好在接收时占有处理机，而相应的处理必须等到该接收进程得到处理机之后才能进行。

12. 为什么要引入 I/O 进程？其功能是什么？

答：引入的主要目的是把 I/O 软件组织成一种层次结构，低层软件用来屏蔽硬件的具体细节，高层软件则主要向用户提供一个简洁、规范的界面。

I/O 进程解决以下 4 个问题：

○ 设备无关性。即程序员写出的软件在访问不同的外围设备时应该尽可能地与设备的具体类型无关，如

1

访问文件是不必考虑它是存储在硬盘、软盘还是 CD-ROM 上。

○ 出错处理。总的来说，错误应该在尽可能靠近硬件的地方处理，在低层软件能够解决的错误不让高层

2

软件感知，只有低层软件解决不了的错误才通知高层软件解决。

○ 同步（阻塞）——异步（中断驱动）传输。多数物理 I/O 是异步传输，即 CPU 在启动传输操作后便转

3

向其他工作，直到中断到达。I/O 操作可以采用阻塞语义，发出一条 READ 命令后，程序将自动被挂起，直到数据被送到内存缓冲区。

○ 独占性外围设备和共享性外围设备。某些设备可以同时为几个用户服务，如磁盘；另一些设备在某一

4

段时间只能供一个用户使用，如键盘。独占性外围设备和共享性外围设备带来了许多问题，操作系统必须能够同时加以解决。

为了合理、高效地解决以上问题，操作系统通常把 I/O 软件组织成以下四个层次：I/O 中断处理程序（底层）；设备驱动程序；与设备无关的操作系统 I/O 软件；用户层 I/O 软件。

13. 什么是 SPOOLING 系统？试简述它的实现思想。

答：操作系统中实现联机同时外围设备操作功能的部分称为斯普林（SPOOLING）系统。它的实现思想是：

利用中央处理器和通道并行工作的能力，用一台机器完成脱机外围设备操作技术中三台机器的工作。

操作系统中包含两个程序：“预输入程序”代替输入外围设备，“缓输出程序”代替输出外围设备。系统在磁盘上划分出专门称为“井”的区域，它分为“输入井”和“输出井”。“预输入程序”把作业流中作业信息传送到“输入井”保存，作业在执行时只要通过“输入井读”程序从上“输入井”获取数据，而不去启动低速的外围设备。作业执行的时候产生的结果也不直接输出到低速设备上，而是先通过“输出井写”程序输出到“输出井”，由“缓输出程序”将“输出井”中的数据再输出到低速设备上。

“缓输出程序”输出的时候，中央处理器可以处理别的事了。实现“输入并读”和“输出并写”程序统称为“井管理”程序。显然，斯普林系统由三个部分组成：“预输入”程序、“井管理”程序和“缓输出”程序。

14. 假定某磁盘有 8 个盘面，每个盘面有 1024 个磁道，每一个磁道有 64 个扇面。每个扇面的尺寸为 1KB。若平均寻道时间为 8ms，磁道到磁道的访问时间为 1.5ms，磁盘驱动器旋转速度为 7200rpm。对处于同一磁柱上的磁道进行访问时无需移动读写磁头。试回答下列问题：

○ 磁盘驱动器的容量是多少？

1

○ 磁盘驱动器的平均访问时间是多少？

2

○ 如果一个扇面的尺寸为 512 字节，试估算该磁盘驱动器传送一个 5MB 尺寸的文件所需的时间。

3

○ 该磁盘驱动器的爆发传输速度是多少？

4

答：容量=8×1024×64×1KB=512MB。

□

$60 \times 1000 / 7200 = 4.17 \text{ms/}$

○ 磁盘驱动器的旋转延迟=旋转时间/2=。

2

磁盘驱动器的平均访问时间=寻道时间+旋转延迟=8+4.17=12.17ms。

○ 为计算方便，假定文件为连续文件，即用完一个磁柱后才用下一个，则一个磁道一个磁道地连续存

3

放，而在一个磁道上则一个扇面一个扇面地连续存放。假定其起始地址为扇面号 #0，盘面号 #0 和磁柱号 #i。一个 5MB 的文件需要 1000 个扇面并且其占住的空间将从（磁柱 #i，盘面 0，扇面 0）开始，到（磁柱 i+9，盘面 6，扇面 7）结束。此时再假定磁盘驱动器的缓冲区的尺寸是无限大。

在上述假定下，磁盘驱动器需要 8ms 的时间进行寻道操作，即找到磁柱 i 需要 8ms 的时间，4.17ms 的时间用于等待 0 号扇面旋转到磁头下方， $8 \times (60/7200) \text{s}$ 的时间来读取该磁柱上所有 8 个磁道的数据。然后，磁头将花费 1.5ms 的时间移动到下一个相邻的磁道。假定在访问每个新磁道时都有一个旋转延迟，则有：

$8 + 9 \times (4.17 + 8 \times 8.35 + 1.5) + 8.3 + 6 \times 8.35 + (8/64) \times 8.35 = 1406.9 \text{ms}$

访问时间=。

旋转次数扇面字节数 7200

$\times \times = \times 64 \times 1 \text{K} = 7.68 \text{MB/s}$

○ 爆发传输速度=。

4

秒一次旋转扇面 60

5.5 综合题

1. 假定当前存取臂的位置在 130 号柱面上，并刚刚完成 125 号柱面的服务请求。请求队列按请求的

先后顺序排列如下：147, 86, 23, 45, 120, 30, 60, 170, 80。试写出为完成上述请求，分别采用

下列算法时存取臂移动的顺序。

(1)最短查找时间优先 (*SSTF*) 算法

(2)铲雪机 (*SCAN*) 算法

(3)电梯调度 (*C-SCAN*) 算法

答: SSTF 算法: 130↗120↗147↗170↗86↗80↗60↗45↗30↗23, 共移臂 207 道。

SCAN 算法: 130↗147↗170↗120↗86↗80↗60↗45↗30↗23, 共移臂 187 道。

CSCAN 算法: 130↗147↗170↗23↗30↗45↗60↗80↗86↗120, 共移臂 284 道。

2. 设某台机挂有两条 I/O 通道, 分别挂一台卡入机和一台打印机, 卡入机上有一叠数据卡片, 现要把这些数据逐一输入到缓冲区 B1, 经处理后传送到缓冲区 B2 中, 然后在打印机中印出。

问:

- (1) 系统可设有哪些进程来完成这个任务?
- (2) 这些进程之间有什么互相制约的关系?
- (3) 用 PV 原语写出这些进程的同步算法。
- (4) 用 Send 和 Receive 原语写出这些进程的同步算法。
- (5) 画出各进程的状态转换图。

解答:

(1) 卡入机、打印机和 CPU 可以并行工作, 因此可设立三个进程: 输入 (R)、复制 (C)、打印 (P)。

(2) 进程间关系为: R→B1→C→B2→P

R 受 C 制约: 当 C 未把 B1 信息取走, R 不能输入下一信息。

P 受 C 制约: 当 C 未把 B1 信息送入 B2, P 不能打印 B2 信息。

C 同时受 R、P 约束: 把 R 未把信息写入 B1; P 未把 B2 信息印出, 则 C 不能把 B1 信息送至 B2。

(3) 设四个信号量。它们初值均为零

B 私用信号量 S1 空。(为“0”表示 B1 空)

C 私用信号量 S1 满。(为“1”表示 B1 满)

C 私用信号量 S2 空。(为“0”表示 B2 空)

P 私用信号量 S2 满。(为“1”表示 B2 满)

PV 原语同步算法如下:

R: 输入到 B1→V(S1 满)→P(S1 空)过程循环往复

C: P(S1 满)→B1 的信息送入 B2→V(S1 空)→V(S2 满)→P(S2 空)过程循环往复

P: P(S2 满)→B2 的信息被打印→V(S2 空)过程循环往复

(4) 通讯原语同步算法

R: 输入到 B1→Send(c)→Receive(c)过程循环往复

C: Receive(B)→B1 的信息送入 B2→Send(R)→Send(P)→Receive(P)过程循环往复

P: Receive(C)→B2 的信息被打印→Send(C)过程循环往复

(5) 状态转换图如下:

第六章 文件管理

6.1 选择题

1. 在 BFD 和 SFD 分开的系统中, SFD 中应记录下列哪类信息? A

A. 文件名 B. 文件长度 C. 存取权限 D. 物理存储位置

2. MS-DOS 系统中的磁盘文件物理结构属于: B

A. 连续文件 B. 链接文件 C. 索引文件 D. 散列文件

3. 基于用户(主体)记录存取权限的方法属于: C

A. 存取控制表 B. 用户目录表 C. 存取控制矩阵 D. 权能表

4. UNIX 系统中的磁盘文件物理结构属于：C
- A. 连续文件 B. 链接文件 C. 索引文件 D. 散列文件
5. 以下哪种类型的文件不支持直接存取（D）
- A、连续文件 B、Hash 文件
- C、索引文件 D、链接文件
6. 可解决文件重名问题的最简单的目录结构是（C）
- A、单级目录 B、树型结构目录 C、二级目录 D、便于共享的目录
7. Hash 文件采用的寻址方法是__A__。
- A、计算 B、比较 C、索引 D、顺序
8. 在 UNIX 系统中使用的目录结构是__C__。
- A、单级 B、二级 C、树型多级 D、三级
9. 文件系统中文件存储空间的分配是以__D__为单位进行的。
- A、字 B、字节 C、文件 D、块
10. 成组链法是用__C__。
- A、文件的逻辑组织 B、文件的物理组织
- C、文件存储器空闲空间的组织 D、文件的目录组织
11. 使用“连访”方式共享文件是指__B__。
- A、不同目录表目指向同一物理入口地址
- B、一个表目指向另一个目录表目
- C、不同的 SFD 表目指向同一 BFD 表目
- D、通过工作目录转换为用户文件固有名进行访问
12. 在传统的操作系统中，设流式文件 youfile 已经打开，并把读写指针 offset 移到 2700 字节处，
- 根据要求说明系统完成用户程序读盘请求：read (fd, 500, 100) 的全过程。（fd：打开的文件描述符；500：存放读入数据的用户区首地址；100：本次读入的字节数。）假设：
- (1) 文件目录采用了符号文件目录（目录项由文件名与索引节点号组成）。
- (2) 物理文件的组织采用了混合索引：直接索引可索引 10 个物理块；一级索引可索引 128 个物理块；youfile 文件体被索引节点中的混合索引映射到从 200 号开始的连续物理块中。
- (3) 文件的逻辑块长等于物理块长，为 512 字节。
- (4) 读操作通过单缓冲进行。
- (5) 本次是第一次读操作。

请从给出语句中挑选合适语句，并对含有 “ ” 的语句填空，描述从请求到完成所经历的——

如下 6 个步骤：

○

请求者进程从用户空间进入核态：（U），（L）。

1

○

设备无关性软件执行：（I），（S），（J），（F），（H），（K）。

2

○

磁盘驱动进程运行：（R），（G），（B），（D）。

3

○

CPU 响应通道发来的完成中断请求：(M), (Q), (O), (N), (P)。

4

○

磁盘驱动进程运行：(T), (C)。

5

○

请求者进程返回用户空间：(E), (A)。

6

可供选择的语句为：

- A. 请求者返回用户态
- B. 启动通道与设备
- C. 唤醒请求者进程，磁盘驱动进程自行阻塞
- D. 设备驱动进程把输入缓冲区中的数据分离出来并传送到请求者进程的数据区，即从缓冲区内偏移地址 2700 （十进制）读 100 字节送用户数据区，修改读写指针 offset 为 2800 （十进制）。 _____
- E. 从中断返回
- F. 把读操作的参数（文件名、所在的设备、物理块号、缓冲区地址）通知磁盘驱动进程
- G. 组织通道程序
- H. 确定 youfile 所在的设备、把逻辑块号转换为物理块号 200 （十进制）。 _____
- I. 请求者进程阻塞
- J. 磁盘驱动进程根据读操作的参数将一维物理块号转换为三维物理地址（柱面号、磁道号、扇区号）
- K. 唤醒磁盘驱动进程
- L. 通过系统调用进入核态
- M. 磁盘驱动进程自行阻塞（如果设备请求队列无其它请求）
- N. 正在执行其它进程的 CPU 响应设备完成中断
- O. 通过外中断进入核态
- P. 再次唤醒磁盘驱动进程
- Q. 分析中断原因，进行磁盘中断处理
- R. 申请分配输入缓冲区
- S. 调用逻辑文件系统求要读的数据所在的逻辑块号 $[2700/512] = 5$ （十进制）。 _____
- T. 释放输入缓冲区
- U. 用户空间执行 read 函数，准备系统调用参数

6.2 填空题

1. MS-DOS 中，假设读目录文件前要先获得其 FAT 链，而获得 FAT 链只需访问磁盘一次。若不考虑磁盘缓冲，为定位 \DIR11\DIR22\DIR33\F0 的首簇，至少需要访问磁盘 4 次；通过相对路径 _____ 名.. \DIR33\F0 定位同一文件的首簇，至少需要访问磁盘 3 次。 _____
2. 文件存储空间的管理常用的技术有 位示图法、空闲块表、空闲块链表 。 _____
3. 文件的逻辑结构的基本形式有 流式文件、记录式文件 。 _____
4. 文件的物理结构的基本形式有 顺序（连续）文件；链接文件；索引文件 。 _____

4. 从用户角度看，文件系统主要是实现 文件的按名存取 。 _____
5. **UNIX** 文件系统把目录项两部分：一部分是文件 文件名 ， 另一部分是文件的 文件 i 节点号 。

6. 文件的访问类型有 顺序访问 和 随机访问 。 _____

6.3 判断题

1. 文件的物理结构仅与文件的存取方法相关。错
2. 顺序存取方法就是严格按照物理记录排列的顺序依次存取。错
3. 串联文件仅支持顺序访问。对
4. 打开文件操作的目的是建立用户和文件的联系。对

6.4 简答题

1. 文件、文件系统的概念？

答：文件是具有符号名的、在逻辑上具有完整意义的一组相关信息项的有序序列。

文件系统就是操作系统中实现文件统一管理的一组软件、被管理的文件以及为实施文件管理所需的一些数据结构的总称。

2. 文件从不同角度（性质和用途、信息的保存期限、保护方式、逻辑结构、物理结构、存取方式、内容，特别是逻辑结构和物理结构），可以分哪几类？

答：根据不同角度，可以将文件划分为不同类别：

○ 按性质和用途可分为：系统文件；库文件；用户文件；

1

○ 按信息的保存期限可分为：临时文件；永久性文件；档案文件；

2

○ 按文件的保护方式可分为：只读文件；读写文件；可执行文件；无保护文件；

3

○ 按文件的逻辑结构可分为：流式文件；记录式文件；

4

○ 按文件的物理结构可分为：顺序文件；链接文件；索引文件；Hash 文件；索引顺序文件

5

○ 按文件的存取方式可分为：顺序存取文件；随机存取文件；

6

○ 按文件内容可分为：普通文件；目录文件；特殊文件

7

3. 文件系统的功能和优点？

答：文件系统的功能：

○ 统一管理文件存储空间（即外存），实施存储空间的分配与回收；

1

○ 确定文件信息的存放位置及存放形式；

2

○ 实现文件从名字空间到外存地址空间的映射，即实现文件的按名存取；

3

○ 有效实现对文件的各种控制操作（如建立、撤消、打开、关闭文件等）和存取操作（如读、写、修改、

4

复制、转储等）；

○ 实现文件信息的共享，并且提供可*的文件保密和保护措施。

5

文件系统的优点：

○ 按名存取文件，以对用户透明的方式实现对名字空间的管理和信息浮动，使用方便灵活；

1

○ 采取保护、保密措施，安全可*；

2

○ 实现文件共享，节省空间和时间开销。

3

4. 文件的存取方式有哪两种？

答：文件的存取方式：○ 顺序存取；○ 随机存取。

12

5. 文件的存储设备有哪些？

答：常见的文件存储设备有磁盘、磁带、光盘等。

6. 什么是文件的物理结构？并具体阐述常用的几种文件物理结构及其优缺点。

答：文件系统在存储介质上的文件构造方式称为文件的物理结构。不论用户看来是什么文件，在存储介质上存储时，按何种构造方式记录呢，因为介质上的存储单位是物理块，那么这些物理块是顺序存放，还是链式结构，或者索引结构，都要由文件系统结构来实现。

常见的文件物理结构有以下几种：

○ 顺序结构，又称连续结构。这是一种最简单的物理结构，它把逻辑上连续的文件信息依次存放在连续

1

编号的物理块中。只要知道文件在存储设备上的起始地址（首块号）和文件长度（总块数），就能很快地进行存取。

这种结构的优点是访问速度快，缺点是文件长度增加困难。

○ 链接结构。这种结构将逻辑上连续的文件分散存放在若干不连续的物理块中，每个物理块设有一个指

2

针，指向其后续的物理块。只要指明文件第一个块号，就可以按链指针检索整个文件。

这种结构的优点是文件长度容易动态变化，其缺点是不适合随机访问。

○ 索引结构。采用这种结构，逻辑上连续的文件存放在若干不连续的物理块中，系统为每个文件建立一

3

张索引表，索引表记录了文件信息所在的逻辑块号和与之对应的物理块号。索引表也以文件的形式存放在磁盘上。给出索引表的地址，就可以查找与文件逻辑块号对应的物理块号。如果索引表过大，可以采用多级索引结构。

这种结构的优点是访问速度快，文件长度可以动态变化。缺点是存储开销大，因为每个文件有一个索引表，而索引表亦由物理块存储，故需要额外的外存空间。另外，当文件被打开时，索引表需要读入内存，否则访问速度会降低一半，故又需要占用额外的内存空间。

○ Hash 结构，又称杂凑结构或散列结构。这种结构只适用于定长记录文件和按记录随机查找的访问方

4

式。

Hash 结构的思想是通过计算来确定一个记录在存储设备上的存储位置，依次先后存入的两个记录在物理设备上不一定相邻。按 Hash 结构组织文件的两个关键问题是：

定义一个杂凑函数；解决冲突；

○ 索引顺序结构。索引表每一项在磁盘上按顺序连续存放在物理块中。

5

7. 什么是文件目录、目录文件与当前目录？

答：文件控制块的有序集合构成文件目录，每个目录项即是一个文件控制块。

为了实现文件目录的管理，通常将文件目录以文件的形式保存在外存空间，这个文件就被称为目录文件。目录文件是长度固定的记录式文件。

系统为用户提供一个目前正在使用的工作目录，称为当前目录。

8. 文件目录结构有哪几种，各有什么优缺点？

答：文件目录结构一般有一级目录结构、二级目录结构和多级目录结构。

一级目录结构的优点是简单，缺点是文件不能重名，限制了用户对文件的命名。

二级目录结构实现了文件从名字空间到外存地址空间的映射：用户名→文件名→文件内容。其优点是有利于文件的管理、共享和保护；适用于多用户系统；不同的用户可以命名相同文件名的文件，不会产生混淆，解决了命名冲突问题。缺点是不能对文件分类；当用文件较多时查找速度慢。

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

9. 为了提高检索速度，对文件目录应做怎样的改进？

答：可以利用目录项分解法解决这一问题，即把目录项（文件控制块）分为两部分：

名号目录项，包含文件名以及相应的文件内部号；

基本目录项，包含了除文件名外文件控制块的其他全部信息。目录文件也分为名号目录文件和基本目录文件。查找一个目录项就分成两步：首先访问名号目录文件，根据文件名查找相应的文件内部号；然后访问基本目录文件，根据文件内部号，可直接计算出相应基本目录项所在基本目录文件中的相对位置和物理位置，并将它直接读入内存。

目录项分解法的优点是提高了文件目录检索的速度。

10. 解释记录的成组和分解

答：当文件的一个逻辑记录的长度小于一个物理块的长度的时候，可以把若干个逻辑记录合并成一组存到一个物理块中，这个过程称为成组。访问某个记录的时候，需要把这个记录从它所在的块中的一组记录中分离出来，这一过程叫做分解。记录的成组和分解可以提高存储空间的利用率，并且可以减少存储设备的启动次数。

11. 假定某个文件由长度为 80 个字符的 100 个逻辑记录组成，磁盘存储空间被划分成长度为 2048 个字符的块，为有效地使用磁盘空间，你可采用成组方式把文件存放到磁盘上，回答下列问题：

①该文件至少占用多少磁盘存储块？

②若该文件是以链接结构形式在磁盘上的，现用户要求使用第 28 个逻辑记录，写出系统为满足用户要求而应做的主要工作。

答：①每块能存放的记录个数为 $\lceil 2048 / 80 \rceil = 25$ 个。一共需要 $100 / 25 = 4$ 块。

②首先系统计算出第 28 个记录在第 2 个物理块上，然后系统通过文件目录读出第一块物理块，在该块最后单元找到第二物理块的地址，读出第二物理块，再经过寻址按第 28 个记录在第二块中的位置读出该记录。

12. 举一例说明数据的分解操作过程。

答：就以上题为例，假定某个文件由长度为 80 个字符的 100 个逻辑记录组成，磁盘存储空间被划分成长度为 2048 个字符的块，为有效地使用磁盘空间，采用成组方式把文件存放到磁盘上，现用户要求每次读一个逻辑记录到他的工作区中，当对该逻辑记录处理后，要求把下一个逻辑记录读入到工作区，直到连续读出 8 个记录。

由于主存储器与外存之间的信息交换是以块为单位的，所以应当在主存中开辟一个 2048 字节的缓冲区。由上题可知每块中含有 25 个记录，文件一共占用 4 块，用户在指明了要读入记录的记录号 N 后，根据公式 $[N / 25]$ 得到该记录应该在 4 块的哪一块中，将该块读入到缓冲区中，并根据公式 $(N / 25)$ 得到该记录是块内的第几个记录（式中 $[]$ 表示取整， $()$ 表示取余数），将该记录从缓冲区拷贝到用户工作区，然

后 $N+1$ ，如果 $([N+1] / 25) = [N / 25]$ ，则将缓冲区中的下一个记录拷贝到用户工作区，而无需启动 I/O 操作，否则，按照新的块号将一个数据块读进缓冲区，并将记录从缓冲区拷贝到用户工作区。

13. 页式存储管理中用位示图表示主存空间的分配情况，磁盘存储空间的分配也可用位示图来表示，两者能合用一张位示图吗？

答：不行，主存空间和磁盘存储空间是两种不同的存储空间，应该使用不同的位示图来表示分配情况。

14. 为了实现按名存取，文件目录应包含哪些内容？

答：为了实现按名存取，文件目录至少应包括文件的名字和文件存放的物理地址，除此之外，目录中还可以包含其他的控制和管理文件的信息，如：文件类型、记录长度、记录个数、口令、建立日期、保存期限、上次修改时间等。

15. 怎样才能防止不同的用户可能给各自的文件取了相同的名字而造成混乱？

答：可以采用二级目录或多级目录结构。在主目录中登记每个用户的名字和用户文件目录的存放地址；在第二级用户文件目录中登记用户的每个文件的文件名及文件存放位置。这样，不同的用户有同名文件时，由于文件的路径是不一样的，所以不会产生混乱。多级目录是在二级目录的基础上，在用户目录下，根据项目和应用领域再建立子目录和孙目录，这样可以避免同一个用户的同名文件造成的混乱。

16. 有一个文件可供两个用户共享，但这两个用户却对这个文件定义了不同的名字，为了保证两个用户都能存取该文件，应怎样设置文件目录？简单画出目录结构关系且解释之。

答：采用二级目录结构。如图 4-3 所示，用户 zhangshan 和用户 lisi 对一个共享文件分别定义了不同的名字 ww 和 pw，只要在它们各自的目录表中把相应的文件存放地址填上共享文件在存储介质上的起始位置，当用户 zhangshan 存取 ww 文件，用户 lisi 存取 pw 文件时，文件系统按照目录查找文件时得到相同的文件存放位置。于是各用户使用了不同的文件名，却仍能共享同一文件。

17. 文件系统提供的主要文件操作有哪些？

答：文件系统提供的主要文件操作有以下几种：

① “建立”操作。用户要求把一个新文件存放到存储介质上时，首先要向系统提出“建立”要求。系统在接到用户的“建立”要求后，在文件目录中寻找空目录项进行登记

② “打开”操作。用户要使用存放在存储介质上的文件前，必需提出“打开”要求。系统在接到用户的“打开”要求后，找到该用户的文件目录，如果文件目录在外存上，还要把它调入到主存，然后从文件目录中找到与用户的需求相符合的目录项，取出文件存放的物理地址。如果是索引文件，还要将这个文件的索引表也调入到主存中，这样，后继的读操作能够很快地进行。

③ “读 / 写”操作。用户调用这个操作来读 / 写文件，系统只允许用户对已经过“打开”或“建立”操作的文件进行读 / 写。对顺序存取方式的文件，用户只需给出读 / 写的文件名，而无需给出读 / 写记录的编号，系统执行本操作的时候，每次顺序读 / 写一个或几个逻辑记录。对于采用随机方式的文件，用户除了要给出需读 / 写的文件名外，还要给出需读 / 写记录的编号（或主键），系统执行读操作的时候，按指定的记录号（或键）查索引表，得到记录存放的物理地址后，按地址将记录读出；执行写操作的时候，在索引表找到一个空登记项且找一个空闲的存储块，把记录存入找到的存储块中，同时在索引表中登记

④ “关闭”操作。经过“打开”或“建立”的文件，在读 / 写完毕后，需要执行“关闭”操作。执行关闭操作时要检查读到主存储器中的文件目录或索引表是否被改变，如果改过，则应把修改过的文件目录或索引表重新保存好。一个关闭后的文件不能再使用，如果要使用，必须重新执行“打开”操作。用户提出“关闭”要求时，必须说明关闭哪个文件。

⑤ “删除”操作。用户用本操作向系统提出删除一个文件的要求，系统执行时把指定文件的名字从目录和索引表中除取，并收回它所占用的存储区域。

18. 文件系统中为什么要设置“建立”、“打开”和“关闭”操作？

答：要把一个文件存放到存储介质上或使用一个已经建立在某存储介质上的文件前，首先应该把文件的属性（文件名、文件类型、可访问性、记录大小等），文件的管理信息（口令、建立日期、保存期限等）

以及存取方式，通过特定的形式告诉文件系统。“建立”。“打开”和“关闭”操作就是为此目的而设置的。

用“建立”操作向系统提出生成一个新文件的要求。

用“打开”操作向系统申请读一指定文件的权力。

用“关闭”操作表示已经不再要读 / 写某个文件了，向系统归还使用文件的权力。

19. 当用户要读一个尚未打开的文件的时候，系统怎么处理？

答：当用户要读一个文件的时候，系统先要验证该用户是否有使用权力，所以任何一个用户如果要读文件前都要执行“打开”操作。系统不允许隐式使用，那么当读一个还没有打开的文件，系统不会执行读操作，而是返回一个“文件未打开”的错误信息。如果系统允许隐式使用，那么系统将会替用户做打开文件的工作。

20. 简述成组链法的基本原理，并描述成组链法的分配与释放过程。

答：首先把文件存储设备中的所有空闲块按 50 块划分为一组。组的划分为从后往前顺次划分。其中，每组的第一块用来存放前一组中各块的块号与总块数。由于第一组的前面已无其它组存在，因此，第一组的块数为 49 块。不过，由于存储设备的空间块不一定正好是 50 的整倍数，因而最后一组将不足 50 块，且由于该组后面已无另外的空闲块组，所以，该组的物理块号与总块数只能放在管理文件存储设备用的文件资源表中。

成组链法的分配和释放过程：

首先，系统在初启时把文件资源表复制到内存，从而使文件资源表中放有最后一组空闲块块号与总块数的堆栈进入内存，并使得空闲块的分配与释放可在内存进行。

与空闲块块号及总块数相对应，用于空闲块分配与回收的堆栈有栈指针 Ptr ，且 Ptr 的初值等于该组空闲块的总块数。当申请者提出空闲块要求 n 时，按照后进先出的原则，分配程序在取走 Ptr 所指的块号之后，再做 $\text{Ptr}=\text{Ptr}-1$ 的操作。这个过程一直持续到所要求的 n 块都已分配完毕或堆栈中只剩下最后一个空闲块的块号。当堆栈中只剩下最后一个空闲块号时，系统启动设备管理程序，将该块中存放的下一组的块号与总块数读入内存之后将该块分配给申请者。然后，系统重新设置 Ptr 指针，并继续为申请者进程分配空闲块。文件存储设备的最后一个空闲块中设置有尾部标识，以指示空闲块分配完毕。

如果用户进程不再使用有关文件并删除这些文件时，回收程序回收装有这些文件的物理块。成组链法的回收过程仍利用文件管理堆栈进行回收。在回收时，回收程序先做 $\text{Ptr}=\text{Ptr}+1$ 操作，然后把回收的物理块号放入当前指针 Ptr 所指的位置。如果 Ptr 等于 50，则表示该组已经回收结束。此时，如果还有新的物理块需要回收的话，回收该块并启动 I/O 设备管理程序，把回收的 50 个块号与块数写入新回收的块中。然后，将 Ptr 重新置 1 另起一个新组。

显然，对空闲块的分配和释放必须互斥进行，否则将会发生数据混乱。

6.5 综合题

1. 基于索引节点共享文件方式有何优缺点？试说明利用符号链实现文件共享的原理。

答：. 基于索引节点共享文件方式的优点有：文件的索引节点包括文件的物理地址及其他文件属性等信息,这些内容不放在目录项中，文件目录只设置文件名和指向索引节点的指针，用户对文件的添加和修改只引起索引节点内容的改变，对其它用户是可见的，从而能提供给其它用户共享。

缺点：索引节点中设有一链接计数 count ，表示共享此文件的用户数，当 $\text{count}>1$ 时，文件主删除文件就会出现悬空指针，可能使其它用户的操作半途而废，若不删除文件，文件主必须为其它用户的操作付费。

2. 文件目录采用索引节点组织方式，文件名目录每个表项占 16 个字节，索引节点占 64 个字节，目录和索引节点分别从 111 号，2 号物理块开始存放（物理块长为 512 个字节）。假设索引节点编号是从 1 到某个最大值，文件 abc 为顺序文件， abc 在文件名目录的第 34 个目录项中，对应的索引节点号为 64。为打开文件 abc 需要启动几次磁盘，以及每次所读的物理块号？并说明原因。

答：由于物理块长为 512 字节，因此一个物理块可以存放的文件数=512/16=32；一个物理块可以存放的索引节点数量=512/64=8。

顺序文件 abc 在目录表中的第 34 项，因此存放在第 34/32=2 个目录块中，即 111+2-1=112 号。

顺序文件 abc 在索引表中的第 64 项，因此存放在第 64/8=8 个目录块中，即 2+8-1=9 号。

从上面的分析可以知道，要打开文件 abc，需要访问目录块 2 次（分别为 111 号和 112 号物理块），定位出文件 abc 的索引节点存放的物理位置，然后 1 次访问索引节点（第 9 号物理块），获取文件 abc 的索引节点信息，最后根据索引节点的地址指针访问文件 abc 的内容。

所以共启动磁盘操作 4 次。

3. 文件系统中的层次结构如下所示：

A 层：文件系统接口

B 层：逻辑文件系统 对对象操纵
和管理的软

C 层：基本 I/O 管理程序（文件组织模块）
件集合

D 层：基本文件系统（物理 I/O 层）

E 层：I/O 控制层（设备驱动程序）

F 层：对象及属性说明

指出以下各种功能在哪个层次上实现？把选择的层次分别填到对应的括号“（ ）”中。

- 1) 把请求读的记录从输入缓冲区中分离出来送到用户工作区。（ A ）
- 2) 磁盘空间的说明。（ F ）
- 3) 启动通道或设备。（ E ）
- 4) 将读/写参数（物理块号和缓冲区地址）向下一层次传送。（ D ）
- 5) 根据读/写记录号或读/写指针求数据所在的相对块号。（ B ）
- 6) 在目录中建立新的目录项。（ B ）
- 7) 根据相对块号确定物理块号。（ C ）
- 8) 组织 I/O 命令序列或通道程序。（ D ）
- 9) 处理设备发来的中断请求。（ E ）
- 10) 指定 I/O 缓冲区。（ E ）

说明：I/O 控制器中有通道设备控制器和指令执行部件。当 CPU 发出启动指令时，就可启动通道并使该通道从内存中调出相应的通道指令执行。

4. 如何提高文件系统的性能？

答：影响文件系统性能的自身因素主要是磁盘的旋转速度和寻道时间。磁盘的旋转速度在不断地提高，但对文件系统性能的影响却十分有限；寻道时间很难大幅度地减少，这是因为影响寻道延迟的因素是磁盘的机械磁臂的移动速度。

基于上述考虑，提高文件系统的性能，需要从操作系统设计上入手，主要包括：

○ 文件缓存。将文件内容的部分置于缓存，从缓存满足用户对文件的访问请求，而无需每次都到磁盘上

1

去读取，从而大大提高访问效率。

○ 预读取。在每次访问磁盘时，多读取一些数据出来。这样在用户随后需要访问这些数据时可以从内

2

存得到满足，从而提高文件访问效率。

○ 减少磁臂移动。由于磁盘访问的瓶颈是其机械运动部分，即磁臂移动和磁盘旋转，而磁臂移动又占主

3

导地位。缩短磁臂移动距离的方法有：将文件头（I-NODE）和数据放置在一起；或将文件读写操作进行适当排序。