

过关宝典

操作系统

课程代码：02326

过关宝典

《操作系统》

课程代码：02326

本课程的重点：

操作系统的特征、功能和作用；中断和异常处理及系统调用机制；进程和线程的概念；进程的数据结构和控制操作；进程状态及状态转换；进程调度算法及其实现；进程同步与互斥；死锁的概念及解决死锁的对策；存储管理的功能；可变分区存储管理方案；虚拟页式存储管理的原理、页面分配、置换和回收方法；文件系统的性能；I/O 软件的层次结构、设备独立性和缓冲技术。

本课程的难点：

操作系统与硬件的关系、操作系统与应用程序的关系；内核态与用户态的关系及相互转换；进程的并发执行；进程与线程的关系；线程机制的类型；各种调度算法以及使用场景；采用信号量及 P、V 操作解决进程同步互斥问题；经典的进程同步的解决方案；死锁避免、资源分配图；虚拟存储技术；虚拟页式存储管理中多级页表及其实现；TLB、页面、页框及磁盘块的对应关系；文件的链接结构、索引结构及多级索引结构；I/O 设备的分配、回收及控制方式；虚拟设备的实现。

考试方式和题目类型

考试方式：闭卷、笔试

考试时间：为 150 分钟

题目类型：

- (1) 单项选择题（本大题共 20 小题，每小题 1 分，共 20 分）
- (2) 填空题（本大题共 10 小题，每小题 2 分，共 20 分）
- (3) 简答题（本大题共 5 小题，每小题 4 分，共 20 分）
- (4) 综合题（本大题共 4 小题，每小题 10 分，共 40 分）

第一章 操作系统概论

第一节 操作系统的概念

一、计算机系统

定义：是一种可以按用户的要求接收和存储信息、自动进行数据处理并输出结果信息的系统。

二、操作系统的定义

定义：操作系统是计算机系统中的一个系统软件，它是这样一些程序模块的集合：它们能有效地组织和管理工作系统中的硬件及软件资源，合理地组织计算机工作流程，控制程序的执行，并向用户提供各种服务功能，使得用户能够灵活、方便、有效地使用计算机，并使整个计算机系统能高效地运行。

三、操作系统的特征

1. 并发性：是指在计算机系统中同时存在着若干个运行着的程序。
2. 共享性：操作系统需与多个用户程序共用系统中的各种资源。

【单选】对资源的共享一般有两种方式：互斥共享（如打印机、磁带机、扫描仪等）、同时共享。

3. 随机性：操作系统不能对所运行的程序的行为以及硬件设备的情况作出任何事先的假定。

四、研究操作系统的观点

1. 软件的观点
2. 资源管理的观点
3. 进程的观点

【单选】4. 虚拟机的观点：操作系统把原来的计算机（裸机）扩充成为功能强、使用方便的计算机系统，我们把这种计算机系统称为虚拟计算机。

5. 服务提供的观点

五、操作系统的功能

【单选】操作系统的功能：1. 进程管理、2. 存储管理、3. 文件管理、4. 设备管理、5. 用户接口。

第二节 操作系统的体系结构

一、Windows 操作系统的体系结构

【单选】Windows 体系结构是分层的模块系统，主要层次有硬件抽象层 HAL、内核、执行体和大量的子系统集合。

二、UNIX 操作系统的体系结构

【单选】UNIX 操作系统的系统架构图，其最里层是硬件；紧临最里层的是操作系统内核；往外一层是系统调用接口；最外层是应用程序。

三、Linux 操作系统的体系结构

【填空】Linux 操作系统四个部分：内核、shell、文件系统和应用程序。

四、Android 操作系统的体系结构

【填空】Android 操作系统四个部分：从低到高：应用程序层、应用框架层、系统运行库层和 Linux 内核层。

第三节 操作系统的发展

1. 手工阶段
2. 监控程序
3. 多道批处理
4. 分时与实时操作系统
5. UNIX 通用操作系统
6. 个人计算机操作系统
7. Android 操作系统

第四节 操作系统的分类

【单选】三种基本类型：1. 批处理系统 2. 分时系统 3. 实时系统

一、批处理操作系统

特点与分类

特点：成批处理，用户不能干预自己作业的运行。

目标：系统资源利用率高，作业吞吐率高。

分类：简单批处理和多道批处理。

二、分时系统

【单选】特点：多路性 交互性 独占性 及时性

三、实时操作系统

分类：硬实时系统和软实时系统

功能：（1）实时时钟管理 （2）过载防护 （3）高可靠性

四、嵌入式操作系统

五、其他操作系统

第五节 操作系统设计

一、操作系统的设计过程

三个部分：功能设计、算法设计、结构设计

二、操作系统的设计目标

1. 可靠性 2. 高效性 3. 易维护性 4. 可移植性 5. 安全性 6. 简明性

三、操作系统的结构设计

操作系统结构研究的目标：系统模块化 模块标准化 通信规范化

四、操作系统的结构

【填空】常见的操作系统结构：1. 整体式结构 2. 层次式结构 3. 微内核（客户/服务器）结构

【单选】微内核（客户/服务器）结构的特点：

1. 运行在核心态的内核
2. 运行在用户态的并以客户/服务器方式运行的进程层

第二章 操作系统运行环境

第一节 处理器

一、处理器的构成与基本工作方式

处理器中的寄存器：用户可见寄存器和控制和状态寄存器。

指令的分类：访问存储器指令、I/O 指令、算数逻辑指令、控制转移指令、控制器控制指令。

二、特权指令和非特权指令

特权指令：在操作系统中那些只能由操作系统使用的指令（如设置程序状态字、启动某设备、设置中断屏蔽、设置时钟指令、清内存指令和建立存储保护指令等）。

非特权指令：普通用户使用的指令。

三、处理器的工作状态

【单选】1.管态和目态

管态：操作系统管理程序运行时的状态，又称内核态、系统态等，具有较高特权。

目态：一般用户程序运行时的状态，又称用户态、普通态，具有较低特权。

2.处理器工作状态的转换

目态到管态的转换：唯一途径是通过中断。中断响应时交换中断向量，新的中断向量的 PSW 的处理器状态标志位管态。

管态到目态的转换：可通过设置 PSW 指令，实现从操作系统到用户程序的转换。

四、程序状态字

【单选】程序状态字一般包括：

1. **CPU 的工作状态代码**：指明当前的工作状态是管态还是目态。

2. **条件码**：反映指令执行后的结果特征，比如结果为 0 等。

3. **中断屏蔽码**：指出是否允许中断。

第二节 计算机系统硬件部件

一、存储系统

1.存储器的类型

【单选】（1）类型

读写型存储器（RAM），用来存储随机存取的程序和数据

只读存储器（ROM），存放一些固化的程序

【单选】（2）存储分块

存储的最小单位：位（bit）

最小编制单位：字节

分块：为了分配和管理方便，将内存划分为大小相等的块（物理页 Page），以**块**为单位分配内存空间，大小一般为 512B、1KB、4KB、8KB 等。

2.存储器的层次结构

3.存储器保护

【单选】界地址寄存器是被广泛使用的一种存储保护技术。

二、I/O 部件

1. I/O 结构
2. 通道
3. DMA 技术
4. 缓冲技术

【单选】采用缓冲技术最根本的原因是，处理器处理数据速度与设备传输数据速度不相匹配，需要用缓冲区缓解其间的速度矛盾。

三、时钟部件

分类：硬件时钟和软件时钟

用途分：绝对时钟和相对时钟

第三节 中断机制

【简答】一、中断与异常的概念

1. 中断与异常

中断：指处理器对系统中或系统外发生的异步事件的响应。

中断和异常的区别：中断是由外部事件引发的，而异常则是由正在执行的指令引发的。

2. 中断与异常的分类

典型的中断：时钟中断、输入输出中断、控制台中断、硬件故障中断。

典型的异常：程序性中断，访管指令异常。

【简答】二、中断系统

中断系统的组成：**中断系统的硬件中断装置和软件中断处理程序。**

1. 中断请求的接收
2. 中断响应
3. 中断处理
4. 几种典型中断的处理，包括：

(1) **I/O 中断**

(2) **时钟中断**

(3) **硬件故障中断**

(4) **程序性中断**：是程序指令出错、指令越权或者指令寻址越界而引发的系统保护。

分两类：一、程序性中断，只能由操作系统完成；二、程序性中断，可以由程序自己完成。

(5) **系统服务请求**

三、中断优先级、中断屏蔽与中断嵌套

1. 多级中断与中断优先级
2. 中断屏蔽
3. 中断嵌套

第四节 系统调用

【单选】为了从操作系统中获得服务，用户程序必须使用系统调用，系统调用陷入内核并调用操作系统。

一、系统调用简介

系统调用概念：就是用户在程序中调用操作系统提供的一些子功能，是操作系统提供给编程人员的唯一接口。

1. 系统调用与函数调用的区别

(1) 运行在不同的系统状态 (2) 状态的转换 (3) 返回问题 (4) 嵌套调用

【单选】2. 系统调用的分类

(1) 进程控制类 (2) 文件操作类 (3) 进程通信类 (4) 设备管理类 (5) 信息维护类

【单选】3. 系统调用与库函数、API、内核函数的关系

库函数是在 C 语言程序设计中，所有匹配标准的头文件的集合，以及常用的函数库实现程序。

系统调用是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”接口来获得操作系统内核提供的服务。

二、系统调用的处理过程

陷入 (trap)：在系统中为控制系统调用服务的机构称为陷入或异常处理机构。

陷入或异常指令 (访管指令)：把由于系统调用引起处理器中断的指令称为陷入或异常指令 (或称访管指令)。

第三章 进程与线程

第一节 多道程序设计

一、程序的顺序执行

【单选】特点：(1) 顺序性 (2) 封闭性 (3) 程序执行结果的确定性 (4) 程序执行结果的可再现性

二、程序的并发执行

特征：

- (1) 在执行期间并发程序相互制约
- (2) 程序与计算不再一一对应
- (3) 并发程序的执行结果不可再现

【单选】(4) 程序的并行执行和程序的并发执行

三、多道程序设计

1. 多道程序设计技术的引入

2. 多道程序设计环境特点

吞吐量：是指单位时间内系统所处理进程的道数，是衡量系统效率的尺度。

特点：(1) 独立性 (2) 随机性 (3) 资源共享性

3. 多道程序设计环境缺陷

第二节 进程

一、进程的定义

进程是具有一定独立功能的程序在某个数据集合上的一次运行活动，是系统进行资源分配和调度的一个独立单

位。

1. 进程与程序的联系和区别

(1) 进程和程序的联系

程序是构成进程的组成部分之一，一个进程的运行目标是执行它所对应的程序。

进程=程序+数据+进程控制块

【单选】(2) 进程和程序的区别

程序是静态的，进程是动态的。

二者是多对多的关系。

2. 可再入程序

一个能够被多个用户同时调用的程序称作是“可再入”程序。可再入程序必须是纯代码。

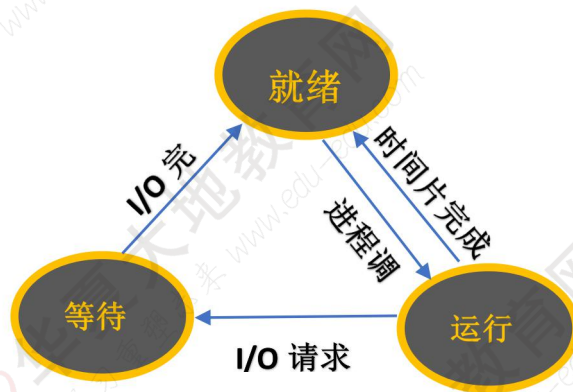
3. 进程的特征：(1) 并发性 (2) 动态性 (3) 独立性 (4) 交往性 (5) 异步性 (6) 结构性

【单选、填空】二、进程的状态与转换

1. 三状态进程模型：(1) 运行状态 (2) 就绪状态 (3) 等待状态

状态转换

- (1) 就绪→运行
- (2) 运行→就绪
- (3) 运行→等待
- (4) 等待→就绪



【单选、填空】2. 五状态进程模型

- (1) 运行状态 (2) 就绪状态 (3) 阻塞状态 (4) 创建状态 (5) 结束状态

【简答】3. 七状态进程模型

- 1) 挂起：把一个进程从内存转到外存。

阻塞到阻塞挂起：没有进程处于就绪状态或就绪进程要求更多内存资源时，会进行这种转换，以提交新进程或运行就绪进程。

就绪到就绪挂起：当有高优先级阻塞进程和低优先级就绪进程时，系统会选择挂起低优先级就绪进程。

运行到就绪挂起：对抢先式分时系统，当有高优先级阻塞挂起进程因事件出现而进入就绪挂起时，系统可能会把运行进程转到就绪挂起状态。

- 2) 激活：把一个进程从外存转到内存。

就绪挂起到就绪：就绪挂起进程优先级高于就绪进程或没有就绪进程时，会进行这种转换。

阻塞挂起到阻塞：当一个进程释放足够内存时，系统会把一个高优先级阻塞挂起进程激活，系统认为会很快出现该进程所等待的事件。

- 3) 事件出现：进程等待的事件出现；

- 4) 提交：完成一个新进程的创建过程，新进程进入就绪状态或就绪挂起状态。

三、进程控制块

1. PCB 的内容

2. 进程的组成

【填空】进程由程序、数据和进程控制块组成。

3.PCB 组织

【填空】组织方式：线性方式 索引方式 链接方式

4.进程的队列

分类：就绪队列、等待队列、运行队列

5.进程队列的组成

进程队列实际是 PCB 的链接，链接分为：单向链表和双向链表。

四、进程控制

进程控制：对进程整个生命周期中各种状态之间的转换进行的控制。由原语实现。

【填空】原语：就是由若干条指令组成的一个指令序列，用于完成一定功能的一个过程。这个指令序列的执行是连续的，具不可分割性，在执行时也不可间断，直到该指令序列执行结束。原语又称原子操作。原子操作在系统态下执行，常驻内存。

【简答】1.进程控制原语

(1) 创建原语

创建原语的过程：先申请一空闲 PCB 区域，然后将有关信息填入 PCB，置该进程为就绪状态，最后把它插入就绪队列中。

(2) 撤销原语

撤销原语的过程：找到要被撤销进程的 PCB，将它从所在队列中消去，撤销属于该进程的一切“子孙进程”，释放被撤销进程所占用的全部资源，并消去被撤销进程的 PCB。

(3) 阻塞原语

(4) 唤醒原语

2.UNIX 操作系统的进程创建操作 fork

第三节 线程

一、线程的基本概念

进程的属性：一个可拥有资源的独立单位；可独立调度和分派的基本单位。

1.什么是线程

【填空】在引入线程的操作系统中，线程是进程中的一个实体，是处理器调度和分配的基本单位。线程基本上不拥有系统资源。

【填空】2.线程的属性

- (1) 每个线程有一个唯一的标识和一张线程描述表。
- (2) 不同的线程可以执行相同的程序。
- (3) 同一个进程中的各个线程共享该进程的内存地址空间。
- (4) 线程是处理器的独立调度单位，多个线程可以并发执行。
- (5) 一个线程具有生命周期，经历等待、就绪、运行等状态变化。

二、进程和线程的比较

【单选】1.调度

在引入线程的操作系统中，把线程作为调度和分派的基本单位，把进程作为资源拥有的基本单位，从而使传统进程的两个属性分开。

2.并发性

3.拥有资源 一个进程中的资源可供它属下的所有线程共享。

4.系统开销

三、线程实现机制

1.用户级线程

【填空】第一种实现方式是用户级线程，不依赖于内核。

2.内核级线程

【填空】第二种实现方式是内核级线程，依赖于内核。

3.混合方式

4.实例：pthread 线程包

第四节 进程调度

一、概述

【简答】1.进程调度的主要功能

- (1) 记录系统中所有进程的执行状况；
- (2) 根据一定的调度算法，从就绪队列中选出一个进程，准备把处理器分配给它；
- (3) 分配处理器

2.进程调度的时机

处理器的调度方式：非抢占式和抢占式

二、调度算法设计原则

1.进程行为 I/O 密集型和计算密集型

2.系统分类 批处理、交互式、实时系统

【简答】3.调度算法的设计目标

批处理目标：平均周转时间短、系统吞吐量高、处理机利用率好。

吞吐量：是系统每小时完成的进程数量。

周转时间：是指从一个批处理进程提交时刻开始直到该进程完成时刻为止的统计平均时间。

【综合】三、进程调度算法

1.先来先服务算法（FCFS）：进程按照请求处理器的顺序使用处理器。

2.最短进程优先算法（SJF）

3.最短剩余时间优先算法（SRTN）：调度程序选择其剩余运行时间最短的进程运行。

4.最高响应比优先算法（HRRF）：

响应比 $R_p = (\text{等待时间} + \text{预计运行时间}) / \text{预计运行时间} = \text{周转时间} / \text{预计运行时间}$

【简答】5.轮转算法：

基本思想：将处理器的处理时间划分成一个个时间片，就绪队列中的诸进程轮流运行一个时间片。当时间片结束时，就强迫运行进程让出处理器，该进程进入就绪队列，等待下一次调度。

结论：时间片设得太短全导致过多的进程切换，降低了处理器效率；而设得太长又可能引起对短的交互请求的响应时间变长。

6.最高优先级算法

可抢占式最高优先级算法和不可抢占式最高优先级算法。

7.多级反馈队列算法

第五节 系统内核

内核的概念：为了提高系统的运行效率、保护系统的关键部分不被破坏，一般把操作系统中提供支持系统运行的各种基本操作和基础功能的一组程序模块集中安排，形成一个操作系统的核心，称为系统核心或系统内核，简称内核。

第四章 进程同步与互斥

第一节 进程间相互作用

一、相关进程和无关进程

- 1.相关进程：在逻辑上具有某种联系的进程
- 2.无关进程：在逻辑上没有联系的进程

二、与时间有关的错误

第二节 进程同步与互斥

一、进程的同步

进程的同步：是指进程之间一种直接的协同工作关系，一些进程相互合作共同完成一项任务。

二、进程的互斥

三、临界区

- 1.临界资源：若在系统中的某些资源一次只允许一个进程使用，则这类资源称为临界资源或共享变量。

【单选】2.临界区：访问临界资源的程序。

- 3.相关临界区：如果有若干进程共享某一临界资源（书上临界区，错误，改正），则该临界区称为相关临界区。

【单选】4.相关临界区的调度使用原则：

- （1）当临界资源（书上是临界区，请改正）空闲时，若有一个进程要求进入临界区，应允许它立即进入。---有空让进，有效利用资源。
- （2）若有一个进程已在临界区，其他要求进入临界区的进程必须等待。---无空等待，互斥进入
- （3）当没有进程在临界区，而同时有多个进程要求进入临界区，选择其一进入，其他等待。---多种择一
- （4）任一进程进入临界区的要求应在有限时间满足---有限等待，避免死等。
- （5）处于等待状态的进程应放弃占用处理器。---让权等待，避免忙等。

第三节 信号量及 P、V 操作

一、信号量

【单选】荷兰学者 Dijkstra 首先提出关于信号量的概念，并引入了在信号量上的 P、V 操作作为同步原语（P 和 V 分别是荷兰文的“等待”和“发信号”两词的首字母）。

二、P、V 操作

【单选】1. P 操作定义:

```
P (S)
{
S=S-1;
```

若 $S \geq 0$, 则该进程继续执行; 若 $S < 0$, 则该进程状态置为等待状态。

【单选】2. V 操作定义:

```
V (S)
{
S=S+1;
```

若 $S \leq 0$, 唤醒在 S 信号量队列中等待的一个进程, 将其状态改变为就绪态。

三、信号量与 P、V 操作的物理含义

【单选】信号量 S 表示某类可用的资源。对于不同的资源, 用不同的信号表示。

$S > 0$ 时, S 表示某类资源的可用数量

$S < 0$ 时, 其绝对值表示排在 S 等待队列中进程的数目。

执行一次 P 操作, 表示请求一个资源;

执行一次 V 操作, 表示进程释放一个资源。

【综合】四、用 P、V 操作实现进程之间的互斥

假设有进程 A、B 竞争进入临界区, 用 P、V 操作实现进程之间的互斥。

首先定义互斥型信号量 S, 并使之初值为 1。

进程 A:

```
P (S) ;
临界区操作;
V (S) ;
```

进程 B:

```
P (S) ;
临界区操作;
V (S) ;
```

【综合】五、用 P、V 操作实现进程间的同步

进程 get:

```
...
while(true){
...
P (S1) ;
从输入设备读数据, 并放入 buffer1;
V (S2) ;
...
}
```

进程 copy:

```
...
while(true){
...
P (S2) ;
P (S4) ;
将 buffer1 的内容复制到 buffer2;
V (S1) ;
V (S3) ;
...
}
```

进程 put:

```
...
while(true){
...
P (S3) ;
将 buffer2 的内容打印输出;
V (S4) ;
...
}
```

【综合】第四节 经典的进程同步问题

一、简单生产者--消费者问题

解决方案:

进程 P:

```
while(true){  
    P(empty);  
    生产一个产品;  
    送产品到缓冲区;  
    V(full);  
};
```

进程 Q:

```
while(true){  
    P(full);  
    从缓冲区取产品;  
    V(empty);  
    消费产品  
};
```

二、多个生产者--消费者问题

算法:

进程 P1、P2, ...:

```
i=0;  
while(true){  
    生产产品;  
    P(empty);  
    P(mutex);  
    往 buffer[i]中放产品;  
    i=(i+1)mod k;  
    V(mutex)  
    V(full);  
};
```

进程 Q1、Q2,...:

```
j=0;  
while(true){  
    P(full);  
    P(mutex);  
    从 buffer[j]取产品;  
    j=(j+1)mod k;  
    V(mutex);  
    V(empty);  
    消费产品;  
};
```

三、读者--写者问题

算法:

读者进程：

```
while(true){  
    P (mutex) ;  
    read_count=readcount+1;  
    if(read_count==1)  
        P(write);  
    V(mutex);  
    读文件;  
    P (mutex) ;  
    read_count=readcount-1;  
    if(read_count==0);  
        V(write);  
    V(mutex);  
};
```

写者进程:

```
while(true){  
    P (write) ;  
    写文件;  
    V(write);  
};
```

四、同步与互斥的综合应用

第五节 管程

一、管程的提出

信号量及 PV 操作的缺点:

- (1) 程序易读性差
- (2) 程序不利于修改和维护
- (3) 正确性难以保证

二、管程的概念及组成

定义：是一个由过程、变量及数据结构等组成的一个集合，它们组成一个特殊的模块或软件包。

【填空】组成：管程名称、共享数据说明、对数据进行操作的一组过程、对共享数据赋初值的语句。

【填空】管程中的共享变量在管程外部是不可见的，外部只能通过调用管程中所说明的外部过程（函数）来间接访问管程中的共享变量。

第六节 进程通信

【单选】P、V 操作是一类低级通信原语，不能承担进程间大量信息的交换任务，因此需要引入新的通信原语，解决大量信息交换问题。

【填空、单选】解决进程之间的大量信息通信的问题有三类方案：**共享内存、消息机制以及通过共享文件进行通信**，即管道通信。这三种方式可以称为**高级通信原语**。

一、共享内存

原理：在相互通信的进程之间设有一个公共内存区，一组进程向该公共内存中写，另一组进程从公共内存中读，通过这种方式实现两组进程之间的信息交换。

二、消息机制

1、消息缓冲

消息缓冲通信原理：进程间的数据交换，是以格式化的消息（也称为报文）为单位的。程序员直接利用操作系统提供的一组通信命令(原语)，实现大量数据的传递，通信过程对用户是透明的。

2、信箱通信

信箱通信原理：为了实现进程间的通信，可以设计一个通信机构--信箱，以发送信件和接收信件为进程间通信的基本方式。

3、管道通信

所谓“管道”，是指用于连接一个读进程和一个写进程以实现他们之间通信的一个共享文件，又名 pipe 文件。

第五章 死锁

第一节 死锁的产生

一、死锁的定义

【简答】定义：指在多道程序系统中，一组进程中的每一个进程均无限期地等待被该组进程中的另一个进程所占用且永远不会释放的资源。

处于死锁状态的进程称为死锁进程。

二、产生死锁的原因

【简答、填空】产生死锁的原因有：**一是竞争资源**，系统资源在分配时出现失误，进程间对资源的相互争夺而造成僵局；**二是多道程序运行时，进程推进顺序不合理**。

资源的概念

(1) 永久性资源（可重用资源）

(2) 临时性资源（消耗性资源）

三、产生死锁的四个必要条件

【填空】对于永久性资源，产生死锁的四个必要条件：**1.互斥条件** **2.不可剥夺条件** **3.请求和保持条件** **4.循环等待条件**。

第二节 死锁预防

【单选】死锁的预防方法：**有资源的静态分配策略**和**资源的有序分配法**。

一、死锁预防的概念

死锁预防：是指在任何系统操作前（如分配资源、调度进程等），事先评估系统的可能情况，严格采取措施，使得产生死锁的四个必要条件不成立。

基本思想：防患于未然。

具体做法：破坏产生死锁的四个必要条件之一。

二、静态的资源分配策略

分配原则：一个进程在申请新资源的要求得不到满足时，便处于等待状态，而处于等待状态的进程的全部资源可以被剥夺。

两个策略：1.破坏不可剥夺条件 2.破坏请求和保持条件

三、资源的有序分配法

【简答】基本思想：是将系统中所有资源顺序编号。

【单选】采用资源有序分配策略，破坏了死锁四个必要条件中的循环等待条件。

第三节 死锁避免

一、死锁避免的概念

基本思想：系统对进程发出的每一个系统能够满足的资源申请进行动态检查，并根据检查结果决定是否分配资源；如果分配后系统不会发生死锁，则予以分配，否则，不予分配。

和死锁预防的区别：死锁预防是破坏产生死锁的四个必要条件之一，严格地防止死锁的出现。而死锁避免是在系统运行过程中注意避免死锁的发生，即使死锁的必要条件存在，也不一定发生死锁。

二、安全状态与安全序列

【单选】不安全状态不一定导致死锁，但死锁状态一定是不安全状态。

【简答】【例 5.5】安全与不安全状态

三、银行家算法

【单选】最著名的死锁避免算法是由 Dijkstra 等人提出来的银行家算法。

第四节 死锁的检测与解除

一、死锁检测时机

【填空】检测死锁的实质是确定是否存在“循环等待”条件，检测算法确定死锁的存在并识别出与死锁有关的进程和资源，以供系统采取适当的解除死锁措施。

二、死锁检测的算法

算法规则：当任一进程 P_j 申请一个已被占用的资源 R_i 时，进行死锁检测，反复查找资源分配表和等待进程表，来确定 P_j 对资源 R_i 的请求是否导致形成环路，若是，出现死锁。

三、解除死锁的方法

【填空】解除死锁的方法：1.剥夺资源 2.撤销进程

第五节 资源分配图

一、资源分配图

作用：描述系统中资源分配和申请情况，对死锁进行分析并采取对策。

二、死锁定理

1.作用：判定死锁的法则

2.死锁定理

- (1) 如果资源分配图中没有环路, 则系统无死锁
- (2) 如果资源分配图中出现了环路, 则可能存在死锁。

三、资源分配图简化方法

第六节 哲学家就餐问题

【单选】哲学家 i ($1 \leq i \leq 4$) 不变, 第五个哲学家的程序改进如下:

```
do{
    P(chopstick[1]); //拿右筷子
    P(chopstick[5]); //拿左筷子
    ...
    eat;
    ...

    V(chopstick[1]); //放右筷子
    V(chopstick[5]); //放左筷子
    ...
    think;
}while(TRUE);
```

第六章 存储管理

第一节 存储管理概述

一、存储体系

定义: 各种速度和容量的存储器硬件, 在操作系统协调之下形成了一种存储器层次结构。

二、存储管理的任务

内存空间一般分为两个区域: 系统区和用户区。

存储管理的实质: 用户空间的管理。

操作系统中存储管理的主要任务: 1. 内存的分配和回收; 2. 存储共享; 3. 存储保护; 4. “扩充”内存容量。

三、地址转换

绝对地址: 存储器以字节为单位编址, 每个字节都有对应的地址。假定内存容量为 n , 则编号顺序为 $0, 1, 2, \dots, n-1$, 该地址称为物理地址或绝对地址。

【填空】由绝对地址对应的内存空间称为“物理地址空间”; 由逻辑地址对应的存储空间称为“逻辑地址空间”。

1. 地址重定位

【单选】把逻辑地址转换成绝对地址的工作称“地址重定位”或“地址转换”, 以称“地址映射”。

重定位的方式可以有“静态重定位”和“动态重定位”两种。

2. 静态重定位

【填空】定义: 由于地址转换工作是在程序开始执行前集中完成的, 所以在程序执行过程中就无须再进行地址转换

工作。

3.动态重定位

【填空】定义：内存在装入程序时，**不进行**地址转换，而是直接把程序装入到分配的内存区域中。在程序执行过程中，每当执行一条指令时都由硬件的地址转换机构将指令中的**逻辑地址**转换成绝对地址。

第二节 分区管理方案

一、固定分区

1.基本思想：多道程序环境下，整个用户空间划分为若干个固定大小的区域，每个分区中只装入一道作业。分区大小可以相同，也可以不同。

2. 内存分配表与分区的分配、回收

内存分配表是一张分区说明表，记录分区号、分区大小、分区起始地址及使用状态等。

二、可变分区

1. 基本思想

系统不预先划分固定分区，而是在装入程序时划分内存分区，使为程序分配的内存区的大小正好等于程序的需求量，且分区的个数是可变的。

【填空】优点：可变分区有较大的**灵活性**，较之固定分区能获得较好的**内存利用率**。

2. 紧缩技术

【填空】解决碎片问题的办法是在适当时刻进行碎片整理，通过移动内存中的程序，把**所有空闲碎片**合并成一个连续的大空闲区且放在内存的一端，而把**所有程序占用区**放在内存的另一端。这一技术称为“紧缩技术”或“压缩技术”。

3. 可变分区的实现

硬件支持：

两个专用控制寄存器：基址寄存器和限长寄存器。

4. 可变分区的分配策略

【填空】三种分配算法：（1）**首次适应算法** （2）**最优适应算法** （3）**最坏适应算法**

5. 分区的回收

6. 分区的保护

【填空】两种方法：（1）**系统设置界限寄存器** （2）**保护键方法**

三、分区管理方案的优缺点

优点：

简单、表格不多，实现起来容易，内存额外开销小，保护措施也简单。

在内存利用率方面可变分区比固定分区高。

缺点：

碎片多，不能为用户提供“虚存”，每个用户程序的存储受物理存储的限制。

第三节 覆盖与交换技术

一、覆盖技术

概念：是指一个程序的若干程序段，或几个程序的某些部分共享某一个存储空间。

二、交换技术

【单选、简答】**交换技术**是进程在内存与外在之间的动态调度，是由操作系统控制的。系统可以将那些不在运行中的进程或其一部分调出内存，暂时存储在外存上的一个后备存储区中，以腾出内存空间给现在需要内存空间的进程，后者可能需从外存的入内存，以后再将换出的进程调入内存继续执行。

主要内容：（1）**换出进程的选择** （2）**交换时机的确定** （3）**交换空间的分配** （4）**换入进程换回内存时位置的确定**。

第四节 虚拟页式存储管理方案

一、虚拟存储技术

【单选、简答】**虚拟存储技术的基本思想**是利用大容量的外存来扩充内存，产生一个比有限的实际内存空间大得多的、逻辑的虚拟内存空间，简称虚存，以便能够有效地技术多道程序系统的实现和大型程序运行的需要，从而增强系统的处理能力。

【简答】**虚拟存储器需要硬件支持：**

- （1）系统有容量足够大的外存
- （2）系统有一定容量的内存
- （3）实现虚-实转换的地址映射机制

【简答】**工作原理：**当进程开始运行时，先将一部分程序装入内存，另一部分暂时留在外存；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的工作；当没有足够的内存空间时，系统自动选择部分内存空间，将其中原有的内容交换到磁盘上，并释放这些内存空间供其他进程使用。

二、虚拟页式存储管理

【单选】页式存储器提供编程使用的虚拟地址由两部分组成：**虚拟页号**和**页内地址**。

三、物理内存的分配与回收

物理页面号=字号×字长+位号

字号= $\lfloor \frac{i}{\text{字长}} \rfloor$ ，位号= $i \bmod \text{字长}$

四、虚拟页式存储地址转换过程

1. 页式存储管理的地址转换

【单选】**页表：**记录装入内存的逻辑页面与物理页面的对应关系。

页式存储管理的地址转换

- （1）十进制计算：物理地址=物理页面号*块长+页内地址。
- （2）二进制计算：物理页面号作为绝对地址的高位地址，页内地址作为它的地址部分。

2. 页表项

物理页面号：页面在内存对应的物理页面号

有效位：页面是在内存还是外存

访问位：页面在内存中是否被访问过

修改位：页面在内存中是否被修改过

保护位：页面能否读/写

3. 页表

多级页表

【单选】大多数操作系统中采用**二级页表**，即由页表页和页目录一起构成进程页表。

散列页表

反置页表

4. 转换检测缓冲区 (TLB)

【单选】TLB 只存储当前进程最活跃的少数几页，随着进程的推进，TLB 的内容**动态进行更新**。

5. 缺页异常处理

缺页异常：若在页表中发现所要访问的页面不在内存，则产生缺页异常。

处理：查看有无空闲页面，若有，把要访问的页面调入内存；若无，选择一页换出内存，再把要访问的页面调入内存。

6. 页面调度策略

调入策略：决定什么时候将一个页由外存调入内存。两种方法：请求调页和预调页。

置页策略：当产生缺页时，将所调入的页面置于何处。

置换策略：如果内存已满，确定哪个页面从内存中移出，为新的页面腾出空位。三种方法：固定分配局部置换、可变分配全局置换、可变分配局部置换。

7. 页面置换算法

【综合】算法：理想页面转换算法 (OPT)、**先进先出页面转换算法 (FIFO)**、第二次机会页面置换算法、时钟页面转换算法 (CLOCK)、**最近最少使用页面转换算法 (LRU)**。

【综合】【例 6.1】、【例 6.2】

8. 缺页率

缺页率计算： $f = F / A$ (F 为缺页次数，A 为页面总访问次数)

五、虚拟页式存储管理的优点缺点

【单选、简答】**优点**：不要求进程的程序段和数据段在内存中连续存放，有效解决了碎片问题，提高了**内存利用率**。

缺点：存在页面空间的浪费，程序的最后一页往往有一部分得不到利用。

六、虚拟存储管理的性能问题

【单选】在虚存中，页面可能在内存与外存之间频繁地调度，有可能出现抖动或颠簸是**请示调页**。

第七章 文件管理

第一节 文件系统的基本概念

一、文件系统的任务

1. 文件的定义

文件的定义：一组带标识的、在逻辑上有完整意义的信息项的序列。

2. 文件系统的定义

【单选】**文件系统**是操作系统中统一管理信息资源的一种软件。

二、文件的存储介质及存取方式

1. 外存储设备的特点

特点：容量大、断电后仍可保存信息

组成：驱动部分和存储介质部分。

种类：磁盘、磁带、磁鼓、纸带、光盘、闪存

2. 外存储设备的存储介质

(1) 磁带

【单选】**磁带**：是一种顺序存取设备，所以文件只能组织成顺序文件。

【单选】**特点**：容量大，存取速度慢，适合顺序存储。

(2) 磁盘

分类：软盘和硬盘

特点：容量大，成本低，适合随机存储。

(3) 光盘

是利用在激光的作用下特性发生变化的一些材料制成的非磁性记录介质。

特点：容量大、速度快、价格便宜。

(4) 闪存

特点：电擦除，随机存取、可靠性高、寿命长。

3. 文件在存储设备中的存取方式

(1) 顺序存取 (2) 随机存取

三、文件的分类

【填空】1. 按文件的用途分类：(1) **系统文件** (2) **库函数文件** (3) **用户文件**

【填空】2. 按文件的组织方式分类：(1) **普通文件** (2) **目录文件** (3) **特殊文件**

3. 一些常见的文件分类方式

(1) 按文件的保护方式：只读文件、读写文件、可执行文件、无保护文件

(2) 按信息的流向分：输入文件、输出文件、输入输出文件

(3) 按存放时限分：临时文件、永久文件、档案文件

(4) 按存储介质分：磁盘文件、磁带文件、卡片文件等

(5) 按文件的组织结构分类：逻辑文件、物理文件

4. UNIX 类操作系统中文件的分类

(1) 普通文件

(2) 目录文件

【单选】(3) 特殊文件 把 **I/O 设备** 看成是一种文件--特殊文件。

第二节 文件的逻辑结构和物理结构

一、文件的逻辑结构

1. 设计文件逻辑结构的原则： (1) 易于操作 (2) 查找快捷 (3) 修改方便 (4) 空间紧凑

2. 文件的逻辑结构

文件的逻辑结构所描述的信息是文件中信息的组织形式，可分为三类：(1) 无结构的字符流式文件 (2) 定长记录文件 (3) 不定长记录文件构成的记录树。

二、文件的物理结构

1. 顺序结构

(2) 顺序结构的优缺点

优点：存取速度快，一旦知道了文件在存储设备上的起始块号和文件长度，便能快速进行存取；支持顺序存放

和随机存取。

【单选】缺点：文件不能动态增长；要求为一个文件分配连续的存储空间；不能灵活地删除和插入记录；出现碎片。

2. 链接结构

(1) 链接结构原理：将逻辑上连续的文件分散存储在若干个不连续的物理块中。每个物理块中都设有一个指针，指向其后续的物理块。

(2) 链接结构的优缺点

优点：解决了碎片问题，提高了磁盘空间利用率；文件可以动态扩充。

缺点：存取速度慢，不适于随机存取；可靠性差。

3. 索引结构

【综合】三、UNIX 的三级索引结构

第三节 文件目录

一、文件控制块

【填空】1. 文件控制块

为文件设置的用于描述和控制文件的数据结构。文件管理程序可借助于文件控制块中的信息，对文件施以各种操作。

【填空】2. 文件目录：

文件控制块的有序集合（文件与文件控制块一一对应）称为文件目录，一个文件控制块就是一个文件目录项。

二、文件目录和当前目录

1. 一级目录结构

优点：简单且能实现目录管理的基本功能——按名存取。

缺点：查找速度慢、不允许重名。

2. 二级目录结构

优点：解决了文件的重名问题和文件共享问题，提高搜索速度，查找时间降低。

缺点：不太适合大量用户和大量文件的大系统，增加了系统开销。

3. 多级目录

【单选】优点：层次结构清晰，便于管理和保护；有利于文件分类；解决重名问题；提高文件检索速度；能进行存取权限的控制。

缺点：查找一个文件按路径名逐层检查，由于每个文件都放在外存，多次访盘影响速度。

4. 当前目录与目录检索

三、目录项和目录文件

1. 目录项

一个文件控制块做成一个定长记录，这个记录称为目录项。

2. 目录文件

多个文件的文件控制块集中在一起组成了文件的目录。

文件目录以文件的形式保存，该文件称为目录文件。

四、目录项分解法

目的：加快目录检索速度。

五、UNIX 的文件目录结构

六、FAT 文件系统的实现

第四节 文件存储空间管理

一、磁盘空间管理

基本思想：对于磁盘空间的分配和回收的方法。

二、磁盘空间的分配与回收算法

【单选、填空】四种方案：

1. 位示图

位示图法的基本思想是，利用一串二进制位（bit）的值来反映磁盘空间的分配使用情况。

2. 空闲块表

3. 空闲块链

4. 空闲块成组链接法

三、空闲块成组链接法

第五节 实现文件系统的表目

一、系统打开文件表

二、用户打开文件表

第六节 文件及目录的操作

一、典型的文件操作

1. 建立文件
2. 打开文件 是使用文件的第一步，任何一个文件使用前都要先打开，即把文件控制块 FCB 送到内存。
3. 读文件
4. 写文件
5. 关闭文件
6. 删除文件

二、典型的目录操作

以 UNIX 系统为例：

1. 创建目录 creat
2. 打开目录 opendir
3. 读目录 readdir
4. 创建链接 link
5. 删除链接 unlink
6. 修改目录名 rename
7. 关闭目录 closedir
8. 删除目录 delete

第七节 文件系统的性能

一、磁盘高速缓存

基本思想：系统在内存中保存一些磁盘块，这些磁盘块在逻辑上属于磁盘，内存的这一区域被称为块高速缓存。

【简答】1.记录的成组

把若干个逻辑记录合成一组存储一物理块的工作称“记录的成组”，每块中的逻辑记录个数称“块因子”。

实现原理：信息交换以块为单位，故成组需要使用内存缓冲区来完成。缓冲区的长度=记录的长度×块因子。

2.记录的分解

【填空】为了提高存储空间的利用率和减少启动设备的次数，采用了记录的成组和分解技术。

二、RAID 技术

RAID 是英文 Redundant Array of Independent Disks 的缩写，中文简称为独立冗余磁盘阵列。

第八节 文件共享、保护和保密

一、文件共享

1.文件共享的概念：文件共享是指一个文件可以允许多个用户共同使用。

2.文件共享的分类

从共享时间段上看，共享文件有两种使用情况：（1）文件可以同时使用 （2）文件不允许同时使用

二、文件的保护

【填空】文件系统经常采用建立副本和定时转储的方法来保护文件。

三、UNIX 的文件使用权限管理方案

【填空】对文件存储权限的设置方法有：存取控制矩阵和二级存取控制等。

四、文件的保密措施

文件保密的目的：是防止不经文件拥有者授权而窃取文件。

常用的文件保密措施有以下几种：1.隐蔽文件目录 2.设置口令 3.使用密码 4.病毒防范

第八章 I/O 设备管理

第一节 I/O 设备管理的基本概念

一、I/O 设备管理的任务

- 1.解决 I/O 设备和 CPU 速度不匹配的矛盾
- 2.提供简单易用的接口，实现对设备的统一管理
- 3.保证设备的安全性

二、I/O 设备分类

- | | |
|----------------|----------------------|
| 1.按设备的使用特性分类 | 输入设备、输出设备、交互式设备、存储设备 |
| 2.按设备的信息组织方式分类 | 字符设备和块设备 |
| 3.按设备使用的可共享性分类 | 独占设备、共享设备、虚拟设备 |

虚拟设备是指在一类设备上模拟另一类设备，目的是为了提高设备利用率。

第二节 I/O 硬件和 I/O 软件的组成

一、I/O 硬件组成

【填空】设备控制器是一种电子部件，每个设备控制器都有若干个寄存器用来与处理器进行通信，包括控制寄存器、状态寄存器和数据寄存器。

二、I/O 软件组成

一般的 I/O 软件结构分为四层：中断处理程序、设备驱动程序、设备独立的操作系统软件 and 用户级软件。

【单选】设备驱动程序是操作系统底层中唯一知道各种输入输出设备的控制器细节以及其用途的部分。

三、设备独立性

【单选、简答】设计 I/O 软件的一个最关键的目标是设备独立性，也就是说，除了直接与设备打交道的底层软件之外，其他部分的软件并不依赖于硬件。I/O 软件独立于设备，就可以提高设备管理软件的设计效率，当 I/O 设备更新时，没有必要重新编写全部 I/O 软件。

第三节 I/O 设备控制方式

四种方式：程序控制方式 中断控制方式 DMA 控制方式 通道控制方式

一、程序控制方式

【单选】程序控制方式也称为 PIO（Programmer I/O）方式，是指由用户进程直接控制处理器或内存和外围设备之间进行信息传送的方式，也称为“忙-等”方式，轮询方式或循环测试方式，这种方式的控制者是用户进程。

优点：处理器和外设的操作能通过状态信息得到同步，而且硬件结构比较简单。

缺点：处理器效率较低，传输完全在处理器控制下完成，对外部出现的异常事件无实时响应能力。

二、中断控制方式

处理器一侧应具备的功能包括：具备识别中断请求的能力；具备响应中断的能力；具备按优先级响应中断请求的能力。

中断控制方式的处理过程：

- (1) 处理器通过数据总线发出命令，启动外设工作，当前进程阻塞，调度程序调度其他进程；
- (2) 外设数据准备好，置位中断请求触发器
- (3) 若此时接口中断屏蔽器状态为非屏蔽状态，则接口向处理器发送中断请求（IR）。
- (4) 处理器接收中断请求，且中断为允许中断状态，则中断判优电路工作。
- (5) 中断判优电路对优先级最高的中断请求给予响应（INTR），处理器中断正在执行的其他进程，转而执行中断服务程序。

优点：

- (1) 处理器与外设并行工作，提高计算机的利用率
- (2) 具有实时响应能力
- (3) 可及时处理异常情况，提高计算机的可靠性

三、DMA 方式

DMA（直接访问内存）方式，是一种完全由硬件执行 I/O 数据交换的工作方式，数据交换不经过处理器，而直接在内存和 I/O 设备之间进行。

DAM 方式传送过程分为三个阶段：（1）预处理阶段 （2）数据传送阶段 （3）后处理阶段

优点：操作均由硬件完成，传输速度快；处理器只在初始化和结束时参与，对数据传输基本不干预，可以减少大批量数据传送时处理器的开销；处理器与外设并行工作，效率高。

四、通道控制方式

【简答】按照信息交换方式的不同，一个系统中可以设立三种类型的通道，即选择通道、数组多路通道和字节多路通道。

1.选择通道是一种高速通道，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。

优点：是以数据块为单位进行传输，传输率高；

缺点：通道利用率低。

2.数组多路通道是对选择通道的一种改进，它的基本思想是当某个设备进行数据传送时，通道只为该设备服务。

优点：是同选择通道一样，以数据块为单位进行传输，传输率高；又具有多路并行操作的能力，通道利用率高；

缺点：控制复杂。

3.字节多路通道是一种简单的共享通道，在分时的基础上为多台低速和中速设备服务。

优点：多路并行操作能力与数组多路通道相同；

缺点：以字节为单位传输，传输效率低。

第四节 设备分配与回收

1.数据结构：系统设备表、设备控制表、控制器控制表、通道控制表。

2.分配原则

设备分配方式：静态分配和动态分配

静态分配效率低，动态分配可能死锁

3.分配策略

主要考虑的因素：I/O 设备的固有属性、分配算法、设备分配的安全性以及设备独立性。

设备固有属性：独占设备、共享设备、虚拟设备

分配算法：先来先服务、高优先级优先

一、独占设备的分配

1.设备的绝对号与相对号

绝对号：系统为每一台设备确定的一个编号，用来区分和识别各种不同类型的外部设备，以便进行管理。

相对号：由用户在程序中定义的设备编号称为设备的“相对号”。

2.设备的指定方式

两种方式：绝对号；设备类、相对号。

【单选】设备类、相对号 优点：使设备分配的适应性好、灵活性强。

3.独占设备的分配和释放

二、共享设备的分配

使用方法：（1）申请设备 （2）启动设备 （3）释放设备

第五节 磁盘调度策略

一、信息传输时间

块号计算: $b=k+s \times (j+i \times t)$

【综合】二、移臂调度及调度算法

移臂调度:根据访问者指定的柱面位置来决定执行次序的调度,称为“移臂调度”。

常用算法:先来先服务调度算法、最短寻找时间优先调度算法、电梯调度算法、单向扫描调度算法。

1. 先来先服务调度算法

根据进程访问磁盘的先后次序进行调度。

优点:公平和简单 缺点:效率低

2. 最短寻找时间优先调度算法

思想:选择这样的进程,其要求访问的柱面号,与当前磁头所在的柱面距离最近,以使每次的寻找时间最短。

优点:平均每次磁头移动距离明显低于 FCFS 的距离,固有较好的寻道性能,过去曾一度被广泛使用。

3. 电梯调度算法

思想:从移动臂当前位置开始沿着臂的移动方向去选择离当前移动臂最近的那个柱面访问,如果沿臂的移动方向无请求,就改变臂的移动方向再选择。

4. 单向扫描调度算法

思想:不考虑访问者等待的先后次序,总是从 0 号柱面开始向里扫描,按照各自所需要访问的柱面位置的次序去选择访问者。当移臂到达最后一个柱面后,立即返回 0 号柱面,再次进行扫描。

三、旋转调度优化

旋转调度:在移动臂定位后,若有若干个访问者等待访问该柱面的情况下,若从减少输入输出操作总时间为目标出发,应该优先选择延迟时间最短的访问者去执行。

根据延迟时间来决定执行次序的调度称为“旋转调度”。

【综合】四、信息的优化分布

记录在磁道上的排列方式也会影响磁盘的输入输出操作时间。

第六节 缓冲技术

一、缓冲的引入

1. 缓和 CPU 与 I/O 设备间速度不匹配的矛盾
2. 减少外部中断次数和处理器进行中断处理所花费的时间
3. 解决 DMA 或通道出现的瓶颈问题

二、缓冲的种类

【单选】根据系统设置的缓冲区的个数,分为:单缓冲 双缓冲 多缓冲 缓冲池。

【单选】1. 单缓冲:在 I/O 设备和处理器之间设置一个缓冲区。

2. 双缓冲:在 I/O 设备和处理器之间设置两个缓冲区。

3. 多缓冲:具有多个缓冲区,其中一部分专门用于输入,另一部分专门用于输出。

4. 缓冲池:把多个缓冲区连接起来统一管理,每个缓冲区既可用于输入,也可用于输出。

三、缓冲池管理

缓冲池的组成

三个队列:空缓冲队列 emq 输入队列 inq 输出队列 outq

第七节 虚拟设备技术

一、虚拟设备的实现原理--SPOOLing 系统工作原理

【单选】虚拟设备技术，又称为 SPOOLing 技术，把一台独占设备改造为可为多个进程共享的设备，实现了虚拟设备的功能。其含义是同时外部设备联机操作，也称假脱机技术。

三部分组成：输入程序模块、输出程序模块、作业调度。

【填空、简答】工作原理：

(1) 输入程序模块，在作业执行前就利用慢速设备将作业预先输入到后援存储器（如磁盘、磁鼓成为输入井），称为预输入。

(2) 作业进入内存后，数据直接从输入井取出。

(3) 作业输出数据时，把数据写入输出井，称为缓输出。待作业全部完成后，再由外部设备输出全部数据。

二、SPOOLing 的组成和实现

1.SPOOLing 系统的组成（如图 8-21 所示）

2.SPOOLing 系统的实现--打印机的值班进程

【单选】由于磁盘是共享的，SPOOLing 系统可以随时响应打印请求并把数据缓存起来，这样就把独占设备改造成了共享设备，从而提高了设备的利用率和系统效率。