

一、绪论

OS四个基本特征

- **并发性**：两个或以上进程同一时间段内执行
- **共享性**：资源供多个并发进程使用
- **虚拟性**：一个物理实体变为若干逻辑对应物
- **异步性**：多道程序环境下，程序执行过程不确定性

OS五个基本功能

- 处理器管理
- 存储器管理
- 设备管理
- 文件管理
- 作为操作系统与用户之间的接口

二、进程管理

2.2进程描述

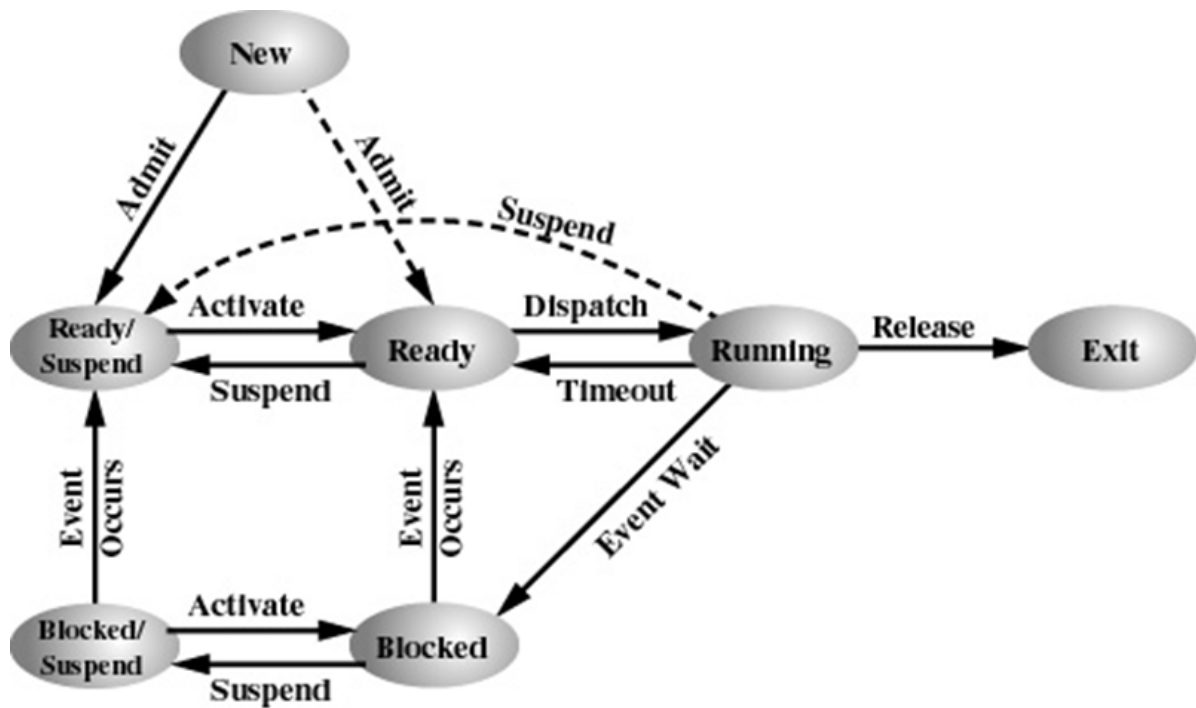
一个进程可以包含多个程序，一个程序可以对应多个进程。

概念：系统进行资源分配和调度的独立单位

结构：程序段、相关数据段、PCB（进程控制块）

特征：动态性*、并发性、独立性、异步性

各状态转换及事件原因（重点）



(b) With Two Suspend States

调度时机

- 创建了新进程，进入就绪队列
- 当前进程被阻塞
- 当前进程挂起自己，转向调度程序重新调度
- 当前进程终止
- 阻塞进程释放
- 挂起进程激活

引起阻塞事件

- 申请共享资源失败
- 等待某种操作完成
- 新数据尚未到达
- 等待新任务到达

PCB

作用

- 独立运行基本单位的标志。
- 能实现间断运行方式。cpu现场信息保留在PCB中。
- 提供进程管理、调度所需信息
- 实现与其他进程同步与通信

包含信息：进程标识符、处理机状态、进程调度信息、进程控制信息

2.3进程控制

操作系统内核

- **系统态**：管态、内核态。能执行一切指令，访问所有寄存器、存储区
- **用户态**：仅执行指定命令，访问指定寄存器

原语

不可分割的基本单位。要么全做，要么全不做。

执行过程中关闭中断。

2.4进程同步

制约关系

- 直接相互制约关系。进程为完成同一项任务而合作。
- 间接相互制约关系。进程对临界资源的访问。

临界区

每个进程访问临界资源的那段代码。

同步准则（互斥条件）

- 空闲让进
- 忙则等待
- 有限等待
- 让权等待

信号量编程（必考）

信号量机制

整型信号量：表示资源数目的整型量S

记录型信号量

- 整型变量value
- 资源信号量：s->value初值表示某类资源数目
- 互斥信号量：s->value初值为1，只允许一个进程访问临界资源

信号量应用

- 实现前趋关系
- 实现互斥

2.5经典进程同步问题

分析互斥条件，设定信号量初值

1. 生产者消费者

- 互斥信号量（对缓冲区访问）、资源信号量empty、资源信号量full
- 一般先执行资源信号量wait，再互斥信号量wait。
- 互斥信号量同一程序中成对出现，资源信号量分别在不同程序中。

2. 哲学家就餐

- chopsticks[5]={1,1,1,1,1}
- 用AND型信号量最简洁

3. 读者写者

- 变量readcount、互斥信号量rmutex（对readcount访问）、互斥信号量wmutex（读写互斥）
- 可同时读、读者优先、读写互斥、互斥写

1. 单缓冲区—生产者—消费者
2. 单缓冲区—生产者多消费者
3. 多缓冲区多生产者多消费者

进程间通信

1. 共享存储器系统
2. 管道通信系统
3. 消息传递系统
4. 客户机-服务器系统

线程

分派和调度部分称为线程，资源所有权部分称为进程。

进程与线程比较

三、处理机调度与死锁

3.1调度层次（重点）

高级调度

作业调度。从外存后备队列调入内存。

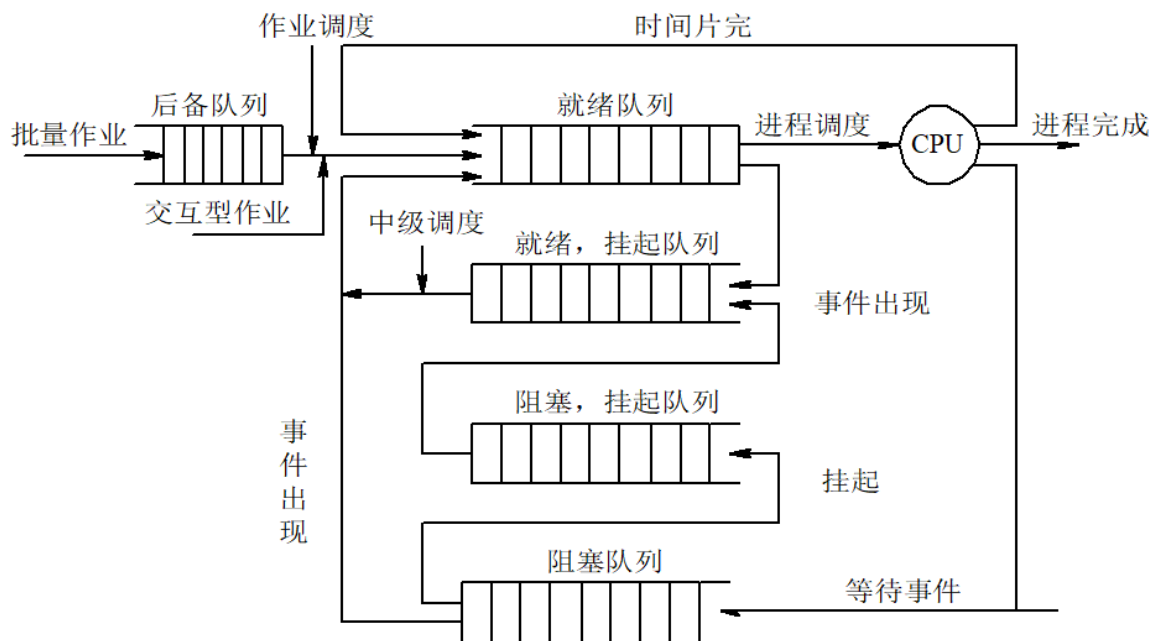
中级调度

内存调度。内存中暂不执行的进程调至外存等待（挂起），外存中具备运行条件的换入。

低级调度

进程调度。为进程分配处理机资源。

任务：保留处理机现场信息、按算法选取进程、分配处理器给内存。



3.2调度准则

吞吐量

CPU利用率

响应时间、周转时间、平均周转时间、平均带权周转时间

$$T = \frac{1}{n} \left[\sum_{i=1}^n \frac{T_i}{T_{Si}} \right]$$

3.3调度算法（重点）

先来先服务（FCFS）

短作业/进程优先（SJF）

时间片轮转（RR）

高优先权优先

- 非抢占式
- 抢占式

高响应比优先（HRRN）

$$\text{优先权} = \frac{\text{等待时间} + \text{要求服务时间}}{\text{要求服务时间}} = \frac{\text{响应时间}}{\text{要求服务时间}}$$

多级反馈队列（FB）

最早截止时间优先（EDF）实时调度算法

3.4死锁（重点）

概念

多个进程在运行过程中因争夺资源而形成的一种僵局。

原因

1. 竞争不可抢占性资源
2. 竞争可消耗性资源
3. 推进顺序不当

必要条件

- 互斥条件
- 请求等待条件
- 不可抢占条件
- 环路等待条件

预防死锁

除了第一个条件，都可以破坏

避免死锁

安全状态

系统能按某种进程顺序，依次为n个进程分配其所需资源，直至其最大需求。

公式

有n个并发进程，每个进程需要m个同类资源，最少需要x个资源能避免死锁。 $(m-1)*n+1 = x$

银行家算法（必考）

教材p120

死锁检测与解除

- 剥夺资源
- 撤销进程

四、存储管理

4.1程序装入和链接

重定位原因：

多道程序环境下，不能预知编译后目标模块放哪。它可以根据内存具体情况，将模块装入合适位置。

- 绝对装入方式。
- 可重定位装入方式。装入时地址转换。
- 动态运行时装入方式。执行时地址转换。

链接方式

- 静态链接
- 动态链接
- 运行时运行链接

4.2连续分配方式

为用户程序分配一个**连续的内存**空间。

单一连续分配

固定分区分配

产生**内零头**：分区内没有被利用的部分。

动态分区分配

产生**碎片（外零头）**：无法利用的空闲分区

1.数据结构

空闲分区表、空闲分区链表（双向）

2.分区分配算法

基于顺序搜索的动态分区分配算法（重点）

- 首次适应
- 循环首次适应
- 最佳适应
- 最坏适应

基于索引搜索的动态分区分配算法

- 快速适应
- 伙伴系统
- hash算法

3.分区分配操作

分配内存、回收内存

动态可重定位分区分配

相比动态分区分配增加了紧凑功能。

4.3离散分配方式

4.3.1分页存储管理

离散分配方式：多个分离的空间为进程占用。充分利用内存空间，无需再“紧凑”。

页、页面：逻辑地址空间。

块、页框：内存空间。

页表：页面映像表。包含页号、块号。

快表：保存进程最近访问过的页表项。

Diagram illustrating a hybrid paging and segmentation system:

- Page Table Register (页表寄存器):** Contains **Page Table Start Address (页表始址)** and **Page Table Length (页表长度)**.
- Logical Address L (逻辑地址L):** Split into **Page Number (页号)** and **Internal Address (页内地址)**.
- Overflow Check:** The **Page Number** is compared with the **Page Table Length** using a **>** operator. If the result is true, a **Overflow Interrupt (越界中断)** occurs.
- Page Table (页表):** A table with columns **Page Number (页号)** and **Block Number (块号)**. The **Page Number** from the logical address is used to look up the **Block Number** (labeled **b**).
- Fast Table (快表):** A table with columns **Page Number (页号)** and **Block Number (块号)**. The **Page Number** from the logical address is used to look up the **Block Number** (labeled **b**).
- Input Register (输入寄存器):** Receives the **Page Number** from the logical address and outputs the **Block Number** to the **Fast Table**.
- Physical Address (物理地址):** Formed by combining the **Block Number (b)** from the **Fast Table** and the **Internal Address (d)**.

段表寄存器

段表始址	段表长度
------	------

越界

段号S 位移量W

2	100
---	-----

有效地址

+

段号	段长	基址
0	1 K	6 K
1	600	4 K
2	500	8 K
3	200	9200

+

8292

物理地址

8K

8292

8692

主存

4.3.3段页式存储管理

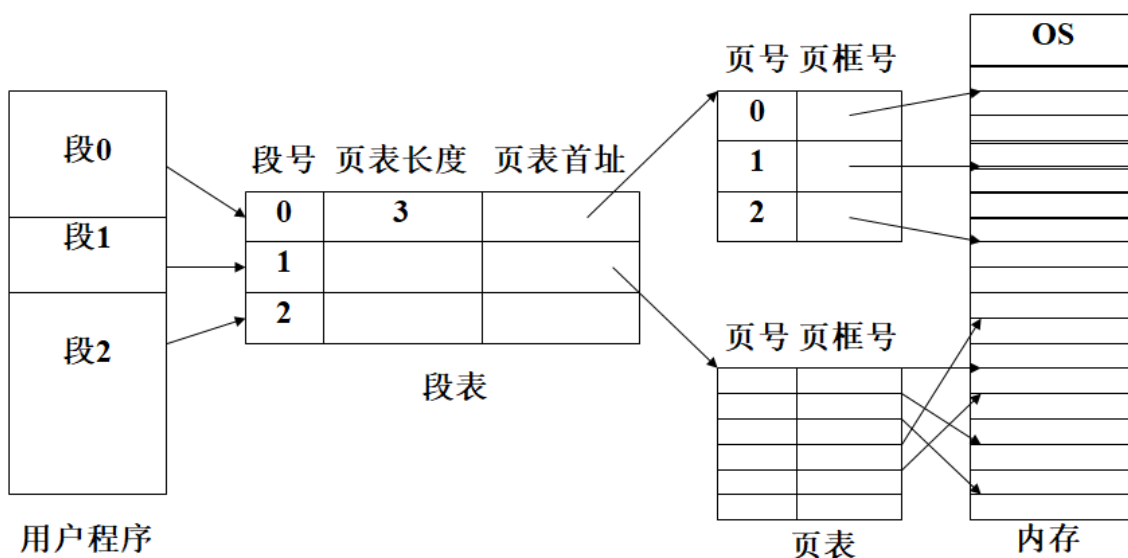


图3.26 段页式系统的数据结构

4.4虚拟存储器

为什么引入？**局部性原理**：时间、空间。仅将需要的一部分页面或段装入内存便可运行。

概念：

具有请求调入和置换功能，能从**逻辑上对内存容量加以扩充**的一种存储器系统。

逻辑容量取决于内存+外存。运行速度接近内存，成本接近外存。

特征：多次性、对换性、虚拟性

4.5请求分页系统

4.5.1硬件支持

1. 请求页表机制

页号	物理块号	状态位P	访问位A	修改位M	外存地址
----	------	------	------	------	------

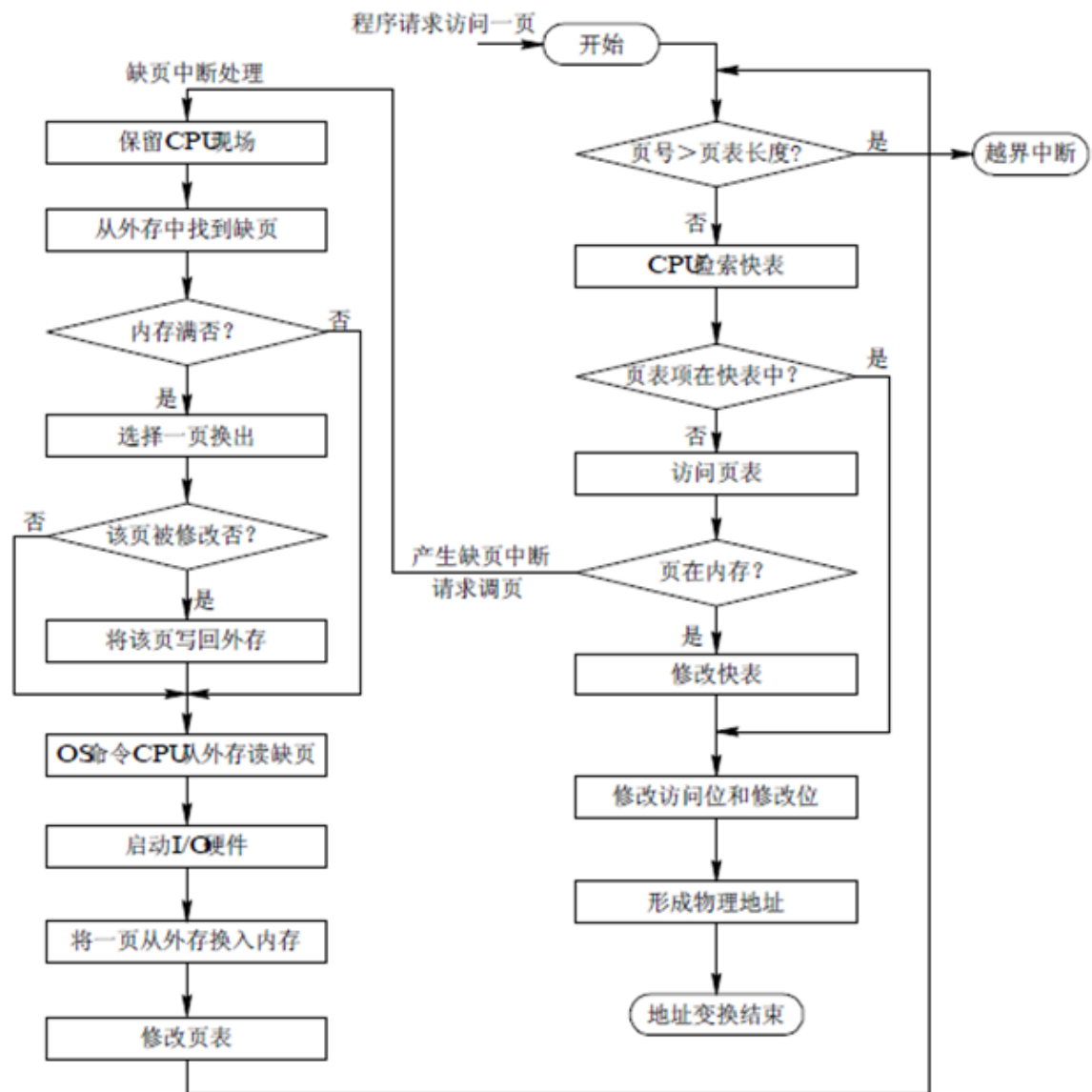
P：该页是否驻留内存

A：最近被访问情况

M：该页是否被修改

2. 缺页中断机构

3. 地址变换机构



4.5.2请求分页中内存分配

最小物理块数确定

物理块分配策略

- 固定分配局部置换
- 可变分配全局置换
- 可变分配局部置换

物理块分配算法

- 平均、比例、优先权

4.5.3页面调入策略

何时装入内存

- 预调页、请求调页

何处

外存分为：存放文件**文件区（离散）**、存放对换页面**对换区（连续）**

- 有足够对换区空间
- 缺少足够对换区空间

- UNIX方式

过程

上面的流程图

4.6 页面置换算法（重点）

缺页率的计算

最佳置换算法

先进先出（FIFO）

最近最久未使用（LRU）

Clock置换（NRU）

改进Clock置换

五、设备管理

I/O软件层次

1. 用户进程
2. 独立性软件
3. 设备驱动程序
4. 中断处理程序
5. 硬件

5.1 I/O控制方式

轮询方式

长期处于忙等状态

中断驱动方式

数据进入寄存器后，发送中断信号让cpu来处理。以**字节**为单位I/O

DMA方式

以**数据块**为单位，直接送入**内存**。仅传送数据块开始结束时需要CPU。CR,MAR,DR,DC

I/O通道方式

cpu向I/O通道发送一条指令，完成**一组数据块**相关操作和控制。

5.2 设备独立性

为什么引入设备独立性

实现应用程序使用设备和系统物理设备无关

如何实现设备独立性

应用程序使用的是逻辑设备

系统实现的映射为物理设备名

5.3SPOOLing技术

SPOOLing技术组成

- 输入输出井
- 输入输出进程
- 输入输出缓冲区

特点

- 提高I/O速度。速度不匹配问题
- 独占改为共享
- 实现虚拟设备功能

如何实现共享打印机

5.4缓冲管理

缓冲引入

- (1)缓和CPU与I/O设备间速度不匹配的矛盾。
- (2)减少对CPU的中断频率，放宽对CPU中断响应时间的限制。
- (3)提高CPU和I/O设备之间的并行性。

单缓冲

双缓冲

循环缓冲

缓冲池

5.5磁盘调度

磁盘存储器性能（重点）

寻道时间 T_s

平均旋转延迟 T_r

传输时间 T_t

总访问时间 T_a

$$T_a = T_s + \frac{1}{2r} + \frac{b}{rN}$$

磁盘调度算法

先来先服务FCFS

最短寻道时间优先SSTF

扫描算法Scan

循环扫描CScan

六、文件管理

6.1文件逻辑结构

顺序文件

索引文件

为变长文件建立索引表。

索引顺序文件

直接（哈希）文件

根据给定的记录键值，通过键值转换，直接获得指定记录的物理地址。

6.2外存分配方式

文件存储管理实际是对外存空间的管理。

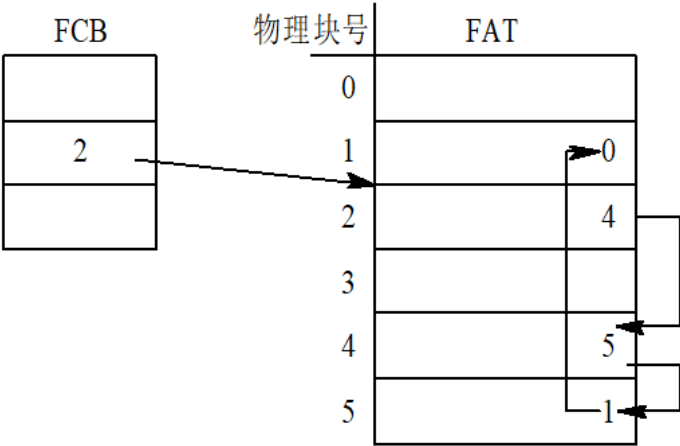
1.连续分配

文件分配表：第一个数据块地址、文件长度

2.链接分配（重点）

隐式链接：文件目录每个目录项中，指向链接文件的第一个盘块和最后一个盘块

显式链接：链接盘块号的指针显式存放在**文件分配表（FAT）**。文件的第一个盘块号作为文件地址填入FCB。



3.索引分配

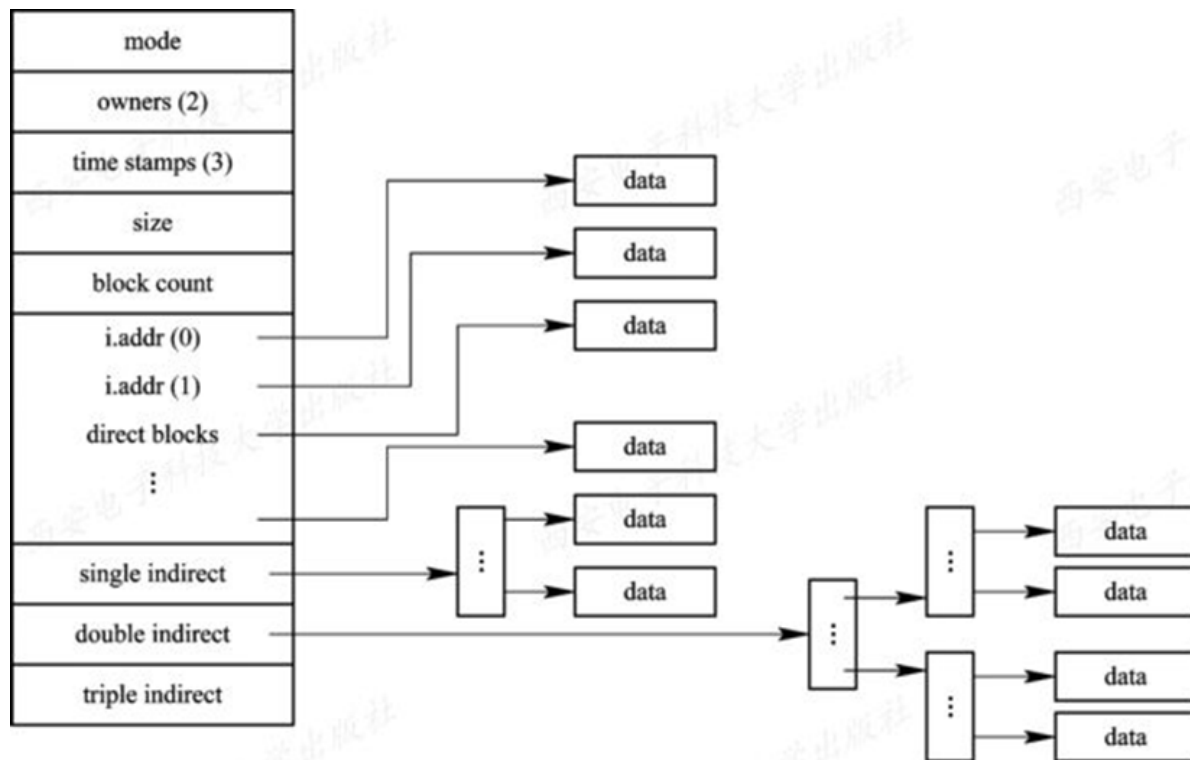
单级索引分配

多级索引分配

增量式索引分配（重点）

UNIX System V的组织方式

- 直接地址
- 一次间接地址
- 多次间接地址



6.3目录管理

文件控制块

索引节点

目录结构

6.4文件磁盘空间管理

空闲分区表

空闲链表

位示图

系统调用

是唯一从用户空间到内核空间的接口。

系统调用函数的实现

系统调用利用软中断实现，操作系统初始化过程中，会产生一张中断向量表，其中保存了中断服务程序的入口地址，发生软中断后通过中断号找到对应的中断服务程序，从而找到注册到操作系统中的服务系统调用的函数，完成功能。

系统调用的处理步骤

1. 处理机状态由用户态->系统态。
2. 保护被中断进程的cpu环境。各种寄存器内容压入堆栈
3. 将用户定义的参数传送到指定地方保存。
4. 转入相应的系统调用子程序。
5. 执行完成，恢复cpu现场，返回进程。

Linux 编程

重定向

管道

环境变量

系统变量

echo命令

单引号、双引号、反撇号、花括号

功能性语句

read, expr, test

结构性语句

说明性语句

简答题

操作系统核心使用的同步技术？

原语、自旋锁、信号量、关中断

利用位图管理磁盘分配、回收的原理？

使用若干连续磁盘块作为位图，位图中的每个位对应一个磁盘块，位号和块号是一一映射。某位为零时，表明对应的磁盘块空闲；为1时，则该盘块已被分配（或已坏）。系统启动时，从磁盘将此位图读入内存供使用；修改位图后或关机时，将内存位图写入磁盘，使位图数据一致，且永久保存。系统运行时，文件系统的管理函数根据位图便可以进行磁盘块的分配和回收。

现代OS基本特征？

简述

PCB作用？为什么说PCB是进程存在唯一标识？

作用：多道进程环境下支持多进程并发执行

原因：整个生命周期内，系统总是通过PCB控制和管理进程。

什么是死锁？预防死锁方式？

三个竞争导致没有外力将一直阻塞

什么是虚拟存储器？如何实现页式虚拟存储器？

概念：具有请求调入和置换功能，逻辑上对内存容量进行扩充的存储系统

请求页表机制，增加了。。。

请求调页技术：通过缺页中断机构，将所需页面调入内存

页面置换技术：内存空间紧张，选择被淘汰的页面

中断驱动和DMA方式区别？

中断频率：每个数据传输后中断；一批数据传输后中断

数据传输方式：cpu控制；DMA控制器

什么是重定位？为什么需要重定位？

执行文件中，逻辑地址转为物理地址的过程

程序中都是逻辑地址从0开始；进入内存后是相对地址，不可能从内存的0开始

临界资源？临界区？访问临界区准则？

必须互斥访问的资源

访问临界资源的代码

空闲让进、忙则等待、有限等待、让权等待

程序的局部性原理？

一段短时间内，局限于某个部分。

时间局部性：指令执行和下次执行，数据访问和下次访问，集中在较短时间内

空间局部性：指令和将执行指令，数据和将访问数据，集中在较小区域内

页式存储和段式存储区别？

页是物理地址，为了实现离散分配消除外零头；大小固定；地址空间一维

段是逻辑地址，为了更好满足用户需要；大小取决于程序；地址空间二维

进程同步与互斥

是指进程推进时不同的制约关系

同步：完成同一任务时，直接制约关系

互斥：使用临界资源，间接制约关系

解决死锁三个方法

预防：破坏必要条件

避免：安全状态

检测：申请和分配不加限制，出现死锁则解除

解除：剥夺资源+撤销进程

调度算法准则

吞吐量、cpu利用率、等待时间、响应时间=等待时间+需要的服务时间

影响缺页中断率的因素

分配给作业的主存块数多少

页面大小

程序编制方式。按行低，按列缺页率高

调度算法最佳

按资源分配管理技术，输入输出设备分为哪几类？

独享设备：一段时间只能被一个作业占用

共享设备：可由若干作业同时共享

虚拟设备：利用独享设备模拟共享设备，spooling技术。

何谓批处理操作系统？多道批处理操作系统优势？

准备好程序、数据、说明书；自动控制作业执行

1.多道程序并行，提高利用率

2.作业批处理，操作到作业的交接时间

3.作业调度可以选择作业

4.不再访问低速设备，直接访问告诉磁盘