

# 概论-重点

2017年12月5日 星期二 23:12

## 1. 早期操作系统设计的主要目标是什么? 填空

- 方便性**: 方便用户使用计算机。用户通过操作系统来使用计算机。
- 有效性**: 使计算机系统能高效可靠地运转, 提高系统资源的利用率。
- 便于操作系统的设计、实现和维护。

## 2. 操作系统是资源管理程序, 它管理系统中的什么资源?

处理机管理、存储器管理、设备管理、文件管理

## 3. 为什么要引入多道程序系统? 它有什么特点?

- 根本目的:
  - 提高CPU的利用率, 充分发挥系统设备的并行性
- 特点: 主存有多道, 宏观上并行, 微观上串行

## 4. 叙述操作系统的基本功能。

- 处理机管理: 解决处理机如何调度的问题: FCFS、优先级、时间片轮转
- 存储器管理: 存储分配、存储保护、主存扩充。
- 设备管理: 分配设备, 控制设备传输数据。
- 文件管理: 将程序、数据、操作系统软件等组织成文件, 存放在磁盘或磁带上, 方便用户访问。

## ★ 5. 批处理系统、分时系统和实时系统各有什么特点? 各适合应用于哪些方面? 填空, 三个系统

- 批处理系统:
  - 优点: 系统吞吐量大, 资源利用率高;
  - 缺点: 用户与作业无法交互, 作业平均周转时间较长。
  - 适合计算量大、自动化程度高的成熟作业
- 分时系统:
  - 特点:
    - 同时性: 若干用户同时使用一台计算机。
    - 独立性: 每个用户占有一台终端, 独立操作, 感觉不到别的用户存在。
    - 交互性: 用户可通过终端与系统进行人机对话。
    - 及时性: 用户的请求能在较短时间内得到响应。
  - 适用于短小作业
- 实时系统:
  - 特点:
    - 实时性。其响应时间由被控制对象所能承受的延迟来确定。
    - 可靠性。要具有容错能力, 可采用双工机制: 一台主机; 一台后备机。
    - 确定性。是指系统按照固定的、预先确定的时间执行指定的操作。其可确定性取决于系统响应中断的速度和处理能力。
  - 适用于需要计算机能对随机发生的外部事件做出及时的响应和处理的作业。是一个专用系统。

## 6. 操作系统的特性?

- **并发性:**
  - 并发是指系统中存在着若干个逻辑上相互独立的程序，它们都被启动执行，都还没有执行完，并竞争系统资源。
- **共享性:**
  - 是指系统中的资源可供系统中多个并发执行的进程共同使用。如打印机、磁带机、磁盘等。支持系统并发性的物质基础是资源共享
- **虚拟性:**
  - 把共享资源的一个物理实体变为若干个逻辑上的对应物。如，CPU的分时共享；虚拟存储器技术。
- **异步性(随机性):**
  - 有限的资源共享使并发进程之间产生相互制约关系。各个进程何时执行、何时暂停、以怎样的速度向前推进、什么时候完成等都是不可预知的

## 7. 衡量OS的性能指标有哪些？什么是吞吐量、响应时间和周转时间？

响应时间:

- 从用户发出请求到系统开始响应所需的时间

性能指标:

- **资源利用率:** 指在给定时间内，系统中某一资源（如CPU、存储器、外部设备等）实际使用时间所占比率。
- **吞吐量(Throughput):** 指单位时间内系统所处理的信息量。它通常是用每小时或每天所处理的作业个数来度量。
- **周转时间:** 指从作业进入系统到作业退出系统所用的时间。而平均周转时间是指系统运行的几个作业周转时间的平均值。

## 8. 什么是嵌入式系统？

以实际应用为中心、以计算机技术为基础、软硬件可裁剪的专用计算机系统。

## 9. 什么是对称多处理？它有什么好处？

对称多处理(SMP):

- 操作系统和用户程序可安排在任何一个处理机上运行，各处理机共享主存和各种I/O设备。

优点:

- 增加了系统的吞吐率。多个作业可以分配在任何一个处理机上执行，大大增加了系统的吞吐率。
- 增加了系统的可靠性。一个处理机的失效，这是性能的降低，不会影响整个系统。

## 10. 为了实现系统保护，CPU通常有哪两种工作状态？各种状态下分别执行什么程序？什么时候发生状态转换？状态转换由谁实现的？

- CPU通常有**核心态**、**用户态**两种工作状态。
- 核心态执行操作系统内核程序，用户态执行用户自编程序和系统外层的应用程序。（简答题）
- 通过**系统调用**可以实现状态的转换，具体是通过**中断和异常（同步、异步什么的判断考了）**来实现。
- 用户态到核心态的转换由硬件完成；核心态到用户态的转换由内核程序执行后完成。

11. 什么是系统调用？什么是特权指令？特权指令执行时，CPU处于哪种工作状态？

- 系统调用就是内核向用户提供的，用来运行系统内核子程序的接口。
- 特权指令是指关系系统全局的指令，如存取和操作CPU状态，启动各种外部设备，设备时钟时间，关中断，清主存，修改存储器管理寄存器，改变用户方式到核心方式和停机指令等。
- 特权指令执行时，CPU处于核心态。

12. 操作系统通常向用户提供哪几种类型的接口？其主要作用是什么？

- 操作接口：命令语言或窗口界面是用户使用计算机系统的主要接口。
- 编程接口：系统调用是用户与操作系统之间的编程接口

# 进程-重点

2017年12月7日 星期四 16:01

## 1. 程序顺序执行的特点

- a. 串行性
- b. 封闭性：程序在运行时独占全机资源，因此，这些资源的状态只能由该程序决定和改变，不受外界因素影响。
- c. 可再现性：只要初始条件相同，无论程序连续运行，还是断断续续地运行，程序的结果不变。

## 2. 何谓进程，进程由哪些部分组成？试述进程的四大特性及进程和程序的区别。

- a. 进程是可以和其他程序并行执行的程序关于某个数据集合的一次执行过程。
- b. 组成：
  - 程序：其包含的程序可以有一个或多个；
  - 数据：可能隐含包括一两个堆栈，主要用来保存函数调用和系统调用时要传递的参数、返回地址和一些临时变量等；
  - 进程控制块：是操作系统对进程进行管理控制使用的一个结构，用来记录进程的属性信息。
- c. 四大特性：
  - 动态性。进程是程序的一次执行过程，是临时的，有生命期的。
  - 独立性。进程是系统进行资源分配和调度的一个独立单位。
  - 并发性。多个进程可在处理机上交替执行。
  - 结构性。系统为每个进程建立一个进程控制块。
- d. 进程与程序的区别：
  - 进程是动态的，程序是静态的。程序是有序代码的集合，进程是程序的执行，没有程序就没有进程。通常，进程不可以在计算机之间迁移，而程序可以复制。
  - 进程是暂时的，程序是永久的。
  - 进程包括程序、数据和进程控制块。
  - 通过多次执行，一个程序可对应多个进程；通过调用关系，一个进程可包括多个程序。进程可创建其他进程，而程序不能形成新的程序。

## 3. 进程控制块的作用是什么？它主要包括哪几部分内容？

- 作用：进程存在的唯一标识
- 组成：
  - 进程标识数：用于唯一地标识一个进程，通常是一个整数。
  - 外部标识符，由用户使用。如：send进程、print进程等。
  - 进程的状态、调度、存储器管理信息：是调度进程所必需的信息，包括进程状态、优先级、程序在主存地址、在外存的地址等。
  - 进程使用的资源信息：分配给进程的I/O设备、正在打开的文件等。
  - CPU现场保护区：保存进程运行的现场信息。包括：程序计数器(PC)、程序状态字、通用寄存器、堆栈指针等。
  - 记帐信息：包括使用CPU时间量、帐号等。
  - 进程之间的家族关系：类UNIX系统，进程之间存在着家族关系，父/子进程。
  - 进程的链接指针：链接相同状态的进程。

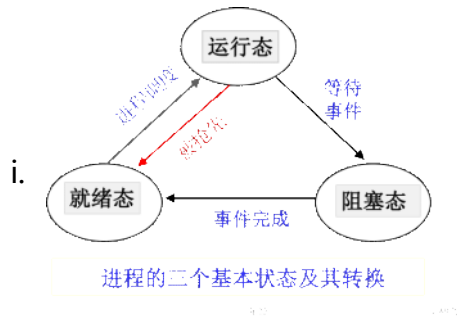
## 4. 进程的基本状态，试举出使进程状态发生变化的事件并描绘它的状态转换

图。

基本状态：

- 运行态(running)：进程正在CPU上运行。单CPU系统一次只有一个运行进程；多CPU系统可能有多个运行进程。
- 阻塞态(blocked)：又称等待态。当进程因等待某个条件发生而不能运行时所处的状态。等待I/O完成，等待一个消息
- 就绪态(ready)：已获得除CPU之外的全部资源，只要再获得CPU，就可执行。

状态转换图见书28页。



## 5. 什么是原语?什么是进程控制?

- 原语是不能被中断的。
- 进程控制是指系统使用一些具有特定功能的程序段来创建、撤消进程，以及完成进程各状态之间的转换。

## 6. 进程调度的功能、方式、时机、算法。作业调度，交换调度。作业的周转时间和作业的带权周转时间?

- a. 进程调度的功能：
- 记录系统中各进程的执行状况，管理进程控制块，将进程的状态变化及资源需求情况及时地记录到PCB中。
  - 选择就绪进程占有CPU
  - 进行进程上下文的切换，将正在执行进程的上下文保存在该进程的PCB中，将刚选中进程的运行现场恢复起来，以便执行。
- b. 进程调度的方式：
- 非抢先方式(非剥夺方式)
    - 某一进程占用CPU,直到运行完或不能运行为止，其间不被剥夺。
    - 用在批处理系统。
    - 主要优点：简单、系统开销小。
  - 抢先方式(剥夺方式)
    - 允许调度程序基于某种策略（优先级、时间片等）剥夺现行进程的CPU给其它进程。
    - 用在分时系统、实时系统。
- c. 进程调度的时机：
- 现行进程完成或错误终止；
  - 提出I/O请求，等待I/O完成时；
  - 在分时系统，按照时间片轮转，分给进程的时间片用完时；
  - 优先级调度，有更高优先级进程就绪；
  - 进程执行了某种原语操作，如阻塞原语和唤醒原语，都可能引起进程调度。
- d. 进程调度的算法：
- 先来先服务(FCFS)
  - 最短作业优先(SJF)
  - 响应比高者优先(HRN)

- 优先级调度法
- 轮转法(Round Robin)
- 多级反馈队列轮转法
- e. 作业调度：高级调度。
  - 多道批处理系统。
    - 多个用户作业以成批的形式提交到外存，形成后备作业队列。被作业调度选中进入内存，就处于运行态。
- f. 交换调度：中级调度。
  - 将主存就绪或主存阻塞等暂不具备运行条件的进程换出到外存交换区；或将外存交换区中的已具备运行条件的进程换入主存。
  - 交换调度可以控制进程对主存的使用。

## 7. 线程的定义，线程与进程的比较。系统对线程的支持。

- **线程：**
  - 是进程内的一个可执行实体，是处理机调度的基本单位。
- **线程与进程的比较：**
  - 拥有的资源
    - 进程拥有一个独立的地址空间，用来存放若干代码段和数据段。若干打开文件，以及至少一个线程。
    - 一个进程内的多线程共享该进程的所有资源，线程自己拥有很少资源。
  - 调度
    - 进程调度需进行进程上下文的切换，开销大。
    - 同一进程内的线程切换，仅把线程拥有的一小部分资源变换了即可，效率高。同一进程内的线程切换比进程切换快得多。
    - 不同进程的线程切换…
  - 并发性
    - 引入线程后，使得系统的并发执行程度更高。进程之间、进程内的多线程之间可并发执行。
  - 安全性
    - 同一进程的多线程共享进程的所有资源，一个线程可以改变另一个线程的数据，而多进程实现则不会产生此问题。共享方便。
- **系统对线程的支持：**
  - **用户级线程**
    - 有关线程的所有管理工作都由用户进程通过调用用户态运行的线程库完成。自己设计线程调度算法。
    - 内核以进程为单位进行调度。一个线程阻塞，其依附的进程也阻塞。
    - 多线程对应核心级一个进程。如，POSIX的Pthread线程库
  - **核心级线程**
    - 有关线程的管理工作都由内核完成。应用程序通过系统调用来创建或撤销线程。
    - 一个线程的阻塞，不影响其他线程的执行。
    - Windows、Linux、多处理机系统
  - **两级组合**
    - 既支持用户级线程，也支持核心级线程。
    - 用户级多个线程对应核心级多个线程。
    - 当内核了解到一个线程阻塞后，通知运行时系统，重新调度其他线程。
    - Solaris 用户线程 $\longleftrightarrow$ LWP $\longleftrightarrow$ 内核线程

## 8. 并发执行的进程在系统中通常表现为几种关系？各是在什么情况下发生的？

- 对资源的共享引起的互斥关系
  - 进程之间本来是相互独立的，但由于共享资源而产生了关系。间接制约关系，互斥关系。



- 协作完成同一个任务引起的同步关系
  - 一组协作进程要在某些同步点上相互等待发信息后才能继续运行。直接制约关系，同步关系。
- 进程之间的前序关系
  - 由于进程之间的互斥同步关系，使得进程之间具有了前序关系，这些关系决定了各个进程创建和终止的时间。

## 9. 什么叫临界资源?什么叫临界区?对临界区的使用应符合的四个准则（互斥使用、让权等待、有空让进、有限等待）。

- 临界资源：
  - 就是一次仅允许一个进程使用的系统中共享资源。
- 临界区(critical section):
  - 就是并发进程访问临界资源的那段必须互斥执行的程序。
- 四个准则：
  - 互斥使用：不能同时有两个进程在临界区内执行
  - 让权等待：等待进入临界区的进程，应释放处理机后阻塞等待
  - 有空让进：在临界区外运行的进程不可阻止其他进程进入临界区
  - 有限等待：不应使要进入临界区的进程无限期等待在临界区之外

## 10. 解决进程之间互斥的办法：开、关中断，加锁、开锁（又叫测试与设置，通常由一条机器指令完成），软件方法，信号量与P、V操作。

- 解决进程之间互斥的方法：
  - 软件实现方法
    - 信号量与P、V操作
    - 克服了加锁操作的忙等待现象，提高了系统的效率。
  - 硬件实现方法
    - 1) 开关中断 最简单的方法。在进程刚进入临界区后，立即禁止所有中断；在进程要离开之前再打开中断。因为CPU只有在发生时钟中断或其它中断时才会进行进程切换。简单，但限制了处理机交叉执行程序的能力
    - 2) 加锁、开锁（测试与设置，通常由一条机器指令完成）

## 11. 若信号量S表示某一类资源，则对S执行P、V操作的直观含意是什么？当进程对信号量S执行P、V操作时，S的值发生变化，当 $S > 0$ 、 $S = 0$ 、和 $S < 0$ 时，其物理意义是什么？

- 调用P(mutex)申请临界资源。
- 调用V(mutex)释放临界资源
- $S > 0$ 时表示有S个资源可用；
- $S = 0$ 时表示无立即可用资源；
- $S < 0$ 时表示前面有S个进程正在等待使用资源。

## 12. 在用P/V操作实现进程通信时，应根据什么原则对信号量赋初值？

## 13. 经典的IPC问题。（大题最后一题，不太难。）

## 14. 进程高级通信有哪些实现机制？（填空）

- 消息缓冲通信

- i. 消息缓冲池
- ii. 信箱
- b. 管道通信
- c. 共享存储区

## 15. 死锁产生的必要条件及解决死锁的方法

- a. 必要条件：（填空）
  - i. 互斥条件。独占性的资源。
  - ii. 保持和等待条件。进程因请求资源而阻塞时，对已经获得的资源保持不放。
  - iii. 不剥夺条件。已分配给进程的资源不能被剥夺，只能由进程自己释放。
  - iv. 循环等待条件。存在一个进程循环链，链中每个进程都在等待链中的下一个进程所占用的资源。
- b. 解决方法：
  - i. 鸵鸟算法。
    - 1) 忽略死锁。
  - ii. 死锁的预防。
    - 1) 通过破坏产生死锁的四个必要条件中的一个或几个，来防止发生死锁。
  - iii. 死锁的避免。
    - 1) 是在资源的动态分配过程中，用某种方法去防止系统进入不安全状态，从而避免发生死锁。
  - iv. 死锁的检测和恢复。
    - 1) 允许死锁发生，通过设置检测机构，及时检测出死锁的发生，然后采取适当措施清除死锁。

16. 理解银行家算法的实质。能够利用银行家算法避免死锁。（大题，作业会了即可）



# 存储器-重点

2017年12月9日 星期六 14:30

## 一. 存储器管理的功能。名字空间、地址空间、存储空间、逻辑地址、物理地址。

- 存储器功能：
  - 存储器分配：解决多道程序或多进程共享主存的问题
  - 地址转换或重定位：研究各种地址变换方法及相应的地址变换机构。
  - 存储器保护：防止故障程序破坏OS和其它信息
  - 存储器扩充：采用多级存储技术实现虚拟存储器及所用的各种管理算法。
  - 存储器共享：并发执行的进程如何共享主存中的程序和数据。
- 名字空间：源程序中的各种符号名的集合所限定的空间。
- 逻辑地址空间：程序限定的空间。
- 物理地址空间：物理存储区中全部物理单元的集合所限定的空间。
- 存储空间：由字或字节组成的一个大的阵列，每一个字或字节都有它自己的编号地址。
- 一个程序只有从地址空间装入到存储空间后才能运行。

## 二. 什么是地址重定位?分为哪两种?各是依据什么和什么时候实现的?试比较它们的优缺点。

- 地址重定位：把程序地址空间的逻辑地址转换为存储空间的物理地址。
- 静态重定位：
  - 在进程执行前，由装入程序把用户程序中的指令和数据的逻辑地址全部转换成存储空间的物理地址。只需经重定位直接装入程序即可使用
  - 特点：
    - 无硬件变换机构；
    - 为每个程序分配一个连续的存储区；
    - 在程序执行期间不能移动，主存利用率低；
    - 难以做到程序和数据的共享；
    - 用于单道批处理系统。
- 动态重定位：（相关概念考了选择）
  - 装入程序把程序和数据原样装入到已分配的存储区中。程序运行时，把该存储区的起始地址送入重定位寄存器。需硬件地址转换机构（即重定位寄存器）。
  - 特点：
    - 主存利用充分。可移动用户程序。移动后，只需修改重定位寄存器。
    - 程序不必占有连续的存储空间。
    - 便于多用户共享存储器中的同一程序和数据。
    - 用于多道批处理系统、分时系统
    - 需额外的硬件支持，增加了系统成本

## 三. 内存划分为两大部分：用户空间和操作系统空间。存储器管理是针对用户空间进行管理的。

## 四. 存储保护的目的是什么?对各种存储管理方案实现存储保护时，硬件和软件各需做什么工作?

- a. 防止地址越界：

- 进程运行时产生的所有存储器访问地址都要进行检查，确保只访问为该进程分配的存储区域。
- b. 正确地进行存取：
  - 对所访问的存储空间的操作方式（读、写、执行）进行检查，以防止由于误操作，使其数据的完整性受到破坏。

## 五. 试述可变式分区管理空闲区的方法及存储区的保护方式。覆盖与交换有什么特点？

- 管理空闲区的方法：分区说明表、空闲区链表。
- 分配算法：（算法会哦，考选择）
  - 首次适应法(first fit)
  - 最佳适应法(best fit)
  - 最坏适应法(worst fit)
- 固定式分区：设置上、下界寄存器来实现存储器保护
- 可变式分区：设置基址+限长寄存器来实现存储器保护（判断）
- 覆盖与交换：（判断）
  - 覆盖特点：打破了必须将一个进程的全部信息装入主存后才能运行的限制。在逻辑上扩充了主存。小主存可运行大进程。
  - 交换特点：打破了一个程序一旦进入主存，便一直运行到结束的限制。

## 六. 页表的作用是什么？简述页式管理的地址变换过程。能利用页表实现逻辑地址转换成物理地址。管理内存的数据结构有哪些？

- 页表的作用：记录逻辑页与主存块的映射关系。
- 页式管理的地址变换过程：（大题？？就是类似作业那种）
  - ①把该进程的页表始址和页表长度放入CPU的控制寄存器中
  - ②将程序计数器内容的页号部分与控制寄存器中的页表长度相比较，若页号p小于页表长度时转③，否则产生地址越界，终止程序运行。
  - ③将程序计数器中的页号与控制寄存器的页表始址相加，得到该访问操作所在页号在页表中的入口地址。这里的加是根据页表项占用的字节数决定的。
  - ④用该地址去访问页表，获得该页对应的主存块号
  - ⑤把主存块号与程序计数器中的页内地址相拼接，从而得到该操作对应主存的物理地址
- 记录存储空间使用情况的两种数据结构：
  - 存储分块表
  - 位示图（大题目balbala情况下用什么方式）

## 七. 什么是页式存储器的内零头？它与页的大小有什么关系？可变式分区管理产生什么样的零头（碎片）？

- 零头（碎片）：在分区存储管理中，每道程序要求占用主存的一个连续的存储区域，因而会产生许多碎片。
- 页尺寸大：页的内部碎片大（进程的代码段、数据段、堆栈段的最后一页都有内部碎片）

## 八. 段式存储器管理与页式管理的主要区别是什么？

段是由用户划分的；

页是为了方便管理由硬件划分的，对用户是透明的。

|                      |               |
|----------------------|---------------|
| 段的大小不固定              | 页的大小固定；       |
| 段式用二维地址空间；           | 页式用一维地址空间。    |
| 段允许动态扩充，便于存储保护和信息共享。 |               |
| 段可能产生主存碎片；           | 页消除了碎片。       |
| 段式管理便于实现动态链接，        | 页式管理只能进行静态链接。 |
| 段与页一样，实现地址变换开销大，表格多。 |               |

## 九. 什么是虚拟存储器。虚拟存储器的容量能大于主存容量加辅存容量之和吗？

- 虚拟存储器：是系统为了满足应用对存储器容量的巨大需求而构造的一个非常大的地址空间。
- 虚拟器的容量可以大于主存加辅存容量。

## 十. 实现请求页式管理，需要对页表进行修改，一般要增加状态位、修改位、访问位。试说明它们的作用。

- 有效位：用来指示某页是否在主存
- 修改位：指示该页调入主存后是否被修改过。“1”表示修改过，“0”表示未修改过。
- 访问位：指示该页最近是否被访问过，“1”表示最近访问过，“0”表示最近未访问

## 十一. 产生缺页中断时，系统应做哪些工作？

- 操作系统处理缺页中断，寻找一个空闲的页框。
- 若有空闲页，则把从磁盘读入信息装入该页框。
- 若无空闲页，则按某种算法选择一个已在内存的页面，暂时调出内存。
- 若修改过还要写磁盘。调入需要的页。之后要修改相应的页表和内存分配表。
- 恢复现场，重新执行被中断的指令。

## 十二. 会利用FIFO、LRU、OPT以及时钟页面置换算法描述页面置换过程，计算产生的缺页率。Belady异常。

页面置换算法：

- 最佳置换算法（OPT算法）
  - 选择以后不再访问的页或经很长时间之后才可能访问的页进行淘汰
- 先进先出置换算法(FIFO)
  - （可能出现抖动）
  - 当淘汰一页时，选择在主存驻留时间最长的那一页。
- 最近最少使用的页面置换算法(LRU)
  - （决不会出现Belady异常）
  - 根据局部性原理，淘汰那些在最近一段时间里最少使用的一页。
- 时钟页面置换算法

Belady异常：采用FIFO算法，当为进程分配的主存块多时，有时产生的缺页中断次数反而增多。（问产生原因。选择）

### 十三. 什么是程序的局部性原理? 什么叫系统抖动? 工作集模型如何防止系统抖动?

- 程序的局部性原理:
  - 时间局部性: 程序中往往含有许多循环, 在一段时间内会重复执行该部分。
  - 空间局部性: 程序中含有许多分支, 在一次执行中, 只有满足条件的代码运行, 不满足条件的代码不运行。即使顺序执行程序, 程序的地址域在短时间内变化不大。在进程运行过程中, 用到哪部分程序或数据再由系统自动装入。
- 抖动: (抖动产生的原因)
  - 刚被淘汰的页面马上又要用, 因而又要把它调入。调入不久再被淘汰, 淘汰不久再次装入。
  - 如此频繁地调入调出, 降低了系统的处理效率。
- 进程的工作集:
  - 在一段时间内, 进程集中在一组子程序或循环中执行,
  - 导致所有的存储器访问局限于进程地址空间的一个固定的子集。
  - 可以大大减少进程产生缺页中断的次数,
  - 甚至在一段时间内不产生缺页中断, 从而大大提高了系统的运行效率。

### 十四. 多级页表的概念, 多级页表中页表建立的时机。写时复制技术的概念。(多级页表那些计算会考的, 找到页, 地址变换)

- 多级页表: 页表在内存不必连续存放
  - 建立时机: 页表的建立不再是在进程装入主存时, 而是推迟到要访问页时, 才为包含该页的页表分配空间和建立页表页。
- 写时复制技术:
  - 若没有进程向共享主存页写时, 两个进程就共享之。
  - 若有进程要写某页, 系统就把此页复制到主存的另一个页框中, 并更新该进程的页表, 使之指向此复制的页框, 且设置该页为可读/写。

### ★/十五. 页的共享问题。需要一个专门数据结构来记录进程间共享页。

- ★◦ 页的共享可以避免在主存中同时有多个相同页的副本。节约主存。
- 通常, 只读页(如程序文本)可以共享, 可读/写的数据页不可共享。
- 需要一个专门的数据结构来记录共享页。把共享页锁在内存, 且在页表中增加引用计数项, 仅当其引用计数为0时, 才允许调出或释放盘空间。

# 文件-重点

2017年12月9日 星期六 21:20

## 一. 什么是文件和文件系统？文件系统的主要功能。UNIX系统如何对文件进行分类？它有什么好处？

- a. 文件：是存储在外部存储器上的具有符号名的相关信息的集合。
- b. 文件系统：
  - OS中管理文件的软件机构。
  - 包括管理文件所需的
    - 数据结构
    - 相应的管理软件
    - 被管理的文件。
- c. 文件系统的主要功能：
  - 管理文件存储器。
    - 记录空间使用情况，分配空间，调整或回收空间。
  - 实现按名存取。
    - 利用目录结构快速定位文件。
  - 应具有灵活多样的文件结构和存取方法，便于用户存储和加工处理信息。
  - 提供一套使用方便、简单的操作命令。
  - 保证文件信息的安全性。
  - 便于文件的共享。
- d. UNIX的文件分类：
  - ①. 普通文件：通常的文件。
  - ②. 目录文件：由文件目录构成的一类用来维护文件系统结构的文件。对其处理同普通文件。
  - ③. 特别文件：输入设备和输出设备（字符型特别文件），输入/输出型设备（字符块特别文件），管道文件。
    - 文件分类的目的：便于系统对不同类型的文件进行不同的管理，以实现文件的保护和共享等。

## 二. 文件目录的作用是什么？文件目录项通常包含哪些内容？文件控制块。

- a. 文件目录的作用：使用户实现按名存取文件，是文件系统的主要目的，实现的主要结构是文件目录。
- b. 文件目录：指记录文件的名称及其存放物理地址的一张映射表，表中包括了许多文件控制块。
- c. 文件控制块：包含了文件的说明信息和管理控制信息。

## 三. 文件的逻辑结构有几种形式？文件的存取方法？

- a. 逻辑结构的形式：
  - 无结构的字节流式文件。
    - 由无结构的先后到达的相关字节组成，其文件长度就是所包含的字节个数。
  - 有结构的记录式文件。
    - 分为定长记录式文件和变长记录式文件。
- b. 文件的存取方法：
  - i. 顺序存取：按照文件信息的逻辑顺序依次存取。是在前一次存取的基础上进行的。
  - ii. 直接存取/随机存取：基于文件的磁盘模型，磁盘允许对任意文件块进行随机读和

写。

- 1) 对记录式文件而言。根据记录的编号来直接存取文件中的任意一个记录。
- 2) 对字节流文件而言。根据系统调用命令把读/写指针调整到欲读/写位置上，然后读/写指定字节数的信息。

#### 四. 文件的物理结构有哪几种?对于不同的结构,文件系统是如何进行管理的?(考存取方式与物理结构没有关系么?, 判断)

- 连续文件/顺序文件
  - (支持顺序存取和随机存取)
  - 文件内容连续存放。
- 链接文件
  - (只能按文件的指针链顺序存取)
  - 不要求文件内容连续存放。
  - 把文件所占用的物理块用链接指针链接起来。
- 索引文件
  - (随机、顺序存取)
  - 为每个文件建立一张索引表。
  - 用索引表记录文件内容的存放地址,即记录文件的逻辑块号和对应的物理块号之间的关系。
- 索引顺序文件

#### 五. DOS文件卷的结构, DOS系统的文件物理结构是什么

- 磁盘低级格式化
  - 划分磁道和扇区
  - 扇区以头标开始,头标中记录该扇区所在的柱面号、磁道号、扇区号
- 硬盘分区
  - 分区规范规定:
    - 一个硬盘可以有多个主分区,
    - 但最多只能有一个扩展分区,
    - 一个扩展分区可以划分为多个逻辑分区,
    - 所以一个硬盘最多只能划为4个主分区/3个主分区和一个扩展分区。扩展分区只作为数据盘使用。书110-111
- 分区格式化
- DOS系统的文件物理结构是 链接文件

#### 六. 了解记录的组块和分解。

- 组块: 把多个逻辑记录存放在一个物理块中的工作叫做记录的组块。
- 分解: 从一个物理块中将一个逻辑记录分离出来的工作叫做记录的分解。

#### 七. 文件存储空间的管理方法有几种?(填空)它们各是如何实现文件存储空间分配和回收的?

- 空白文件目录
  - 适合于文件的静态分配(连续文件的分配)
- 空闲块链表
  - 空闲块链
    - 把所有空闲块连接成一个链表。
    - 优点: 简单。适合文件动态分配。
    - 缺点: 工作效率低。分配和回收多个盘块时要多次访问磁盘才能完成。
  - 空闲块成组链表



- 利用盘空闲块来管理盘上的空闲块，每个磁盘块记录尽可能多的空闲块而成一组。各组之间也用链指针链接在一起。
- 便于空闲块的分配与回收。
- 适合连续文件、链接文件和索引文件的存储分配。
- 位映像表(bit map)或位示图
  - 每一个二进制位对应一个物理盘块。为1时表示块已分配，为0时空闲。

## 八. 建立多级目录有哪些好处?文件的重名和共享问题是如何得到解决的?

- 多级目录的好处:
  - 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;
  - 能够控制存取权限。
- 解决共享问题:
  - 用户B和用户C共享的目录和文件之间的连线叫做链接
    - **符号链接: 创建一个新目录项，其中存有指向另一个文件或目录的绝对路径名。**
    - **硬链接: 在共享目录项中简单地重复被共享文件的信息，因此两个目录项完全相同。**

## 九. 文件系统中，常用的文件操作命令有哪些？它们的具体功能是什么?打开和关闭文件命令的目的是什么？（现代操作系统不需要打开文件就可以读么？判断）

- 操作命令:
  - **创建(Create)文件:** 在指定设备上为指定路径名的文件建立一个目录项，并设置文件的有关属性。
  - **删除>Delete)文件:** 根据文件的路径名找到指定的目录项，回收其占用的各个物理块，再将该目录项置为空。
  - **打开(Open)文件:** 根据文件路径名找到目录项，将文件的目录项复制到主存一个专门区域，返回文件在该区域的索引。建立进程与文件的联系。
- 目的: 避免多次重复地检索文件目录。  
系统维护了一个系统当前打开文件表。当读/写文件时，通过这个表的索引找到文件的主存目录项。不需要重复地对磁盘进行检索。
- 文件的操作命令:
  - 关闭(Close)文件:
    - 释放文件在主存专门区域中的目录项，切断用户与文件的联系。
      - 若该目录项被修改过，则复制到磁盘。
      - 若文件作过某些修改，应将其写回辅存。
  - 读(Read)文件:
    - 命令中必须指出要读的数据个数，以及存放数据的主存地址。
      - 根据文件所在设备、文件类型的不同，系统设置不同的读命令。
  - 写(Write)文件:
    - 命令中必须指出要写的数据个数，以及存放数据的主存地址，将主存中的数据写到指定的文件中。
  - 追加(Append)文件:
    - 限制了写文件的形式，将数据追加到文件尾。
  - 随机存取(Seek)文件:
    - 重新定位文件的读/写位置指针。
  - 得到文件属性(Get Attributes):
    - 进程在执行时常常需要了解文件的属性。在UNIX系统中，一个软件开发项目通常由

多个源文件组成，make程序用来管理这些软件开发项目。当make被调用时，它检查所有源文件和目标文件的修改时间，并且编排出需要重新编译的文件数。

- 设置文件属性(Set Attributes):
  - 修改文件的一些属性，以适应用户的要求。
- 重命名(Rename)文件:
  - 重新命名一个已经存在的文件。

## 十. 存取控制表ACL的概念。

为存取控制矩阵中的每一列建立一张存取控制表(ACL)，用一有序对(域, 权集)表示。

## 十一. 理解内存映射文件（memory mapped file）的过程。

将文件映射到进程地址空间的一个区域，返回虚拟地址，仅当需要对文件存取时，才传输实际的数据。使用与请求页式虚存管理相同的存取机制。访问的页不在主存时，产生缺页中断读入主存。OS提供映射文件的系统调用。

# 设备-重点

2017年12月10日 星期日 13:00

## 一. I/O设备通常大致可分为哪两大类？各自传输的信息单位有什么特点？

- 字符设备：
  - 人机交互设备。是以字符为单位发送和接收数据的，通信速度比较慢。键盘和显示器、鼠标、扫描仪、打印机、绘图仪等。
- 块设备：外部存储器。以块为单位传输数据。
  - 常见块尺寸：512B~32KB。如磁盘、磁带、光盘等。

## 二. 常用的四种数据传输方式。（书126-130）（填空）

- 程序查询方式（polling）
- 中断方式
- 直接存储器访问(DMA)方式
- 通道控制方式

## 三. 根据设备的使用方式，设备被分为几种类型？何为虚拟设备？它是通过什么技术实现的？（简述）

- 根据使用方式，设备被分为三类：
  - 独占设备：临界资源，如打印机。
  - 共享设备：多个进程可交叉访问。如磁盘。
  - 虚拟设备：是指设备本身是独占设备，而经过虚拟技术处理，可以把它改造成共享设备，供多个进程同时使用。
- Spooling技术
  - 是实现虚拟设备的具体技术。
  - 它利用可共享磁盘的一部分空间，模拟独占的输入/输出设备。
  - 以空间换时间。

## 四. 按照设备管理的层次结构，I/O软件划分为几层？各层主要实现哪些功能？

- I/O软件划分层：
  - 1) I/O硬件
  - 2) 中断处理程序
    - 进程在启动一个I/O操作后阻塞起来，I/O操作完成，控制器产生一个中断。
    - CPU响应中断，执行中断处理程序。
    - 检查设备状态。
      - 若正常完成，就唤醒等待的进程。然后检查是否还有待处理的I/O请求，若有就启动。
      - 若传输出错，再发启动命令重新传输；或向上层报告“设备错误”的信息。
    - 中断返回被中断的进程，或转进程调度。
  - 3) 设备驱动程序
    - 功能：
      - 设备初始化。
      - 启动设备传输数据的例程。
      - 中断处理例程。
  - 4) 独立于设备的软件
    - 功能：
      - 基本任务：实现所有设备都需要的功能，且向用户提供一个统一的接口。
      - 设备命名。
      - 设备保护。
      - 提供与设备无关的块尺寸
      - 缓冲技术

- 负责设备分配和调度
- 出错处理

#### 5) 用户层的I/O接口

- 大部分I/O软件都包含在操作系统中，有一小部分是由与用户程序连接在一起的库函数构成的。这些函数通常只是将系统调用时所需要的参数放在合适的位置，由系统调用实现真正的操作。

### 五. 何为设备的独立性？

设备独立性是指用户及用户程序不受系统配置的设备类型和具体设备的台号的影响。用户只是使用逻辑设备，具体的映射由操作系统完成。

### 六. 设计目标

- 设备独立性
- 设备的统一命名
- 出错处理
- 缓冲技术
- 设备的分配

### 七. 什么是SPOOLING技术？以输出为例，说明它的实现原理。（SPOOLING技术是以空间换时间）

书134

### 八. 一个特定磁盘上的信息如何进行编址？盘面号、磁道号 和扇区号（或柱面号、磁头号和扇区号）。

书110页

### 九. 要将磁盘上一个块的信息传输到主存需要系统花费哪些时间？

寻道时间 + 旋转延迟时间 + 读/写传输时间

### 十. 常用的磁盘调度算法：先来先服务、最短寻道时间优先、扫描法（SCAN, C\_SCAN, LOOK, C\_LOOK）。

# Linux进程管理

2017年12月10日 星期日 17:52

## 一. 进程控制块，其中与进程管理、存储器管理和文件管理有关的一些字段，线程组标识符。

进程描述符（task\_struct）：描述进程的数据结构。

唯一的标识符PID

pid t tgid: 线程组标识符

## 二. 与进程创建有关的函数：fork()、vfork()、clone()

- 创建子进程函数fork():
  - 创建成功之后，子进程采用写时复制技术读共享父进程的全部地址空间，仅当父或子要写一个页时，才为其复制一个私有的页的副本。
- 创建轻量级进程函数clone():
  - 实现对多线程应用程序的支持。共享进程在内核的很多数据结构，如页表、打开文件表等等。
- vfork()系统调用：
  - 创建的子进程能共享父进程的地址空间，为了防止父进程重写子进程需要的数据，先阻塞父进程的执行，直到子进程退出或执行了一个新的程序为止。

## 三. 理解进程切换的过程。涉及到页目录表、核心栈、硬件上下文。

- 进程切换：
  - 第一步，切换页目录表以安装一个新的地址空间；
  - 第二步，切换核心栈和硬件上下文。由schedule()函数完成进程切换。
- 进程切换只发生在核心态。在发生进程切换之前，用户态进程使用的所有寄存器值都已被保存在进程的核心栈中。
- 进程硬件上下文存放在进程描述符的thread\_struct thread中。该结构包含的字段涉及到大部分CPU寄存器，但像eax、ebx等通用寄存器的值仍被保留在核心栈中。

## 四. 进程调度方式。进程调度时机。

- Linux系统采用可抢先式的动态优先级调度方式。无论进程处于用户态还是核心态运行，都可能被抢占CPU。
- 调度时机：
  - 出现了更高优先级的实时进程
  - 进程执行了阻塞操作而进入睡眠状态。
  - 进程停止运行或被杀死。
  - 进程调用自愿放弃处理机。
  - 在基于时间片轮转的实时进程调度过程中，进程用完了自己的时间片。

## 五. Linux有很多内核线程，了解0号进程和1号进程的作用。（选择）

- 0号进程/idle进程/swapper进程。
  - 所有进程的祖先进程
  - 其进程描述符存放在init\_task变量中。
  - 每个CPU都有一个0号进程。
- 1号进程
  - 是由0号进程创建的内核线程init，

- 负责完成内核的初始化工作。
- 在系统关闭之前，init进程一直存在，它负责创建和监控所有用户进程。

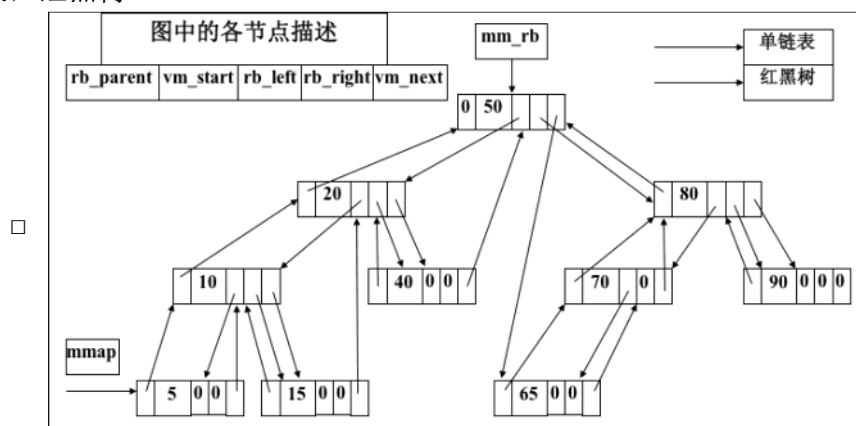


# Linux存储器管理 CH8

2017年12月11日 星期一 23:24

## 一. 进程地址空间的划分？管理进程私有地址空间的数据结构？链接虚拟内存区域的单链表和红黑树。指向映射文件对象的指针字段？指向进程页目录表的指针字段？

- 进程地址空间的划分：Linux把地址空间分成两部分。进程的私有空间是前3G，进程的公有空间是后1G的内存空间。
- 管理私有地址空间的数据结构：采用虚拟内存描述符mm\_struct。
- 进程地址空间的各个虚拟内存区域是采用单链表和红黑树相组合的方式进行管理的。（）  
填空，单链表，红黑树



## 二. Linux堆的管理：malloc()，free()。

- malloc(size): 请求size个字节的动态内存。
- free(addr): 释放内存。

## 三. 管理物理内存页框的数据结构？ 内存管理区zone结构，伙伴系统？ 分区页框分配器分配页框的过程。（简述）

- 页框描述符为struct page。
- Linux把内存节点划分为3个管理区zone。
  - ZONE\_DMA:
  - ZONE\_NORMAL:
  - ZONE\_HIGHMEM:
- 伙伴系统: 管理连续的空闲内存页框，以解决外碎片问题。
  - 伙伴系统以页框为单位，适合于对大块内存的分配请求。
- 分区页框分配器:
  - 负责处理对连续物理页框的分配请求。
  - 管理区分配器负责搜索一个能满足动态内存请求的内存管理区。
    - 当空闲页框不足时，管理区分配器应当触发页框回收算法。
  - 在每个管理区内的页框，除了一小部分页框被保留为每CPU页框高速缓存外（以满足本地CPU发出的对单个页框的请求），其它的由伙伴系统来管理。

## 四. 理解slab分配器的原理。slab分配器的作用？

- slab分配器用于为只有几十或几百个字节的小内存区分配内存。如，file对象。

slab分配器把小内存区看作对象，slab分配器对不再引用的对象只是释放但内容保留，以后再请求新对象时，就可直接使用而不需要重新初始化。

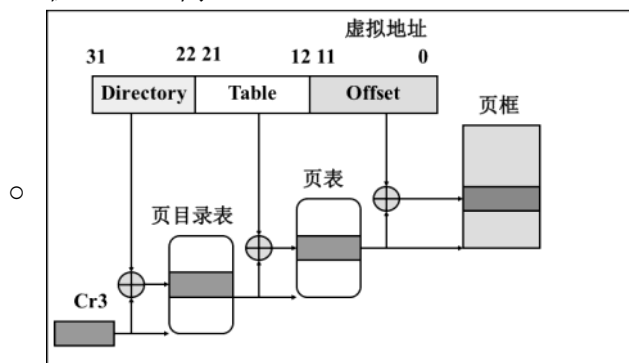
#### a. slab分配器

- 从分区页框分配器获得几组连续空闲页框
- slab分配器为不同类型的对象生成不同的高速缓存，每个高速缓存存储相同类型的对象。
- 高速缓存由一连串的slab构成，每个slab包含了若干个同类型的对象。



## 五. 进程页表建立的时机？了解页目录表项或页表项所包含的字段。

- 逻辑地址的划分，利用两级页表实现地址转换的过程。
- 处理机普遍采用二级页表模式，为每个进程分配一个页目录表，页表一直推迟到访问页时才建立，以节约内存。（判断）
- 虚地址分成3个域：页目录索引（前10位）、页表索引（中10位）和页内偏移（后12位）。win同



## 六. 请求调页。所缺的页可能存放的地方。

- 请求调页机制是把页框的分配一直推迟到进程要访问的页不在RAM中时引起一个缺页异常，才将所需的页调入内存。请求调页增加了系统中的空闲页框的平均数。
- 页面置换策略是LFU（Least Frequently Used）
- 所缺页的可能存放处：
  - i. 该页从未被进程访问过，且没有相应的内存映射。
  - ii. 该页属于非线性内存映射文件。非线性内存映射的是文件数据的随机页。给定文件的所有非线性映射虚拟内存区域描述符都存放在一个双向链表中。
  - iii. 该页已被进程访问过，但其内容被临时保存到磁盘交换区上。
  - iv. 该页在非活动页框链表中。
  - v. 该页正在由其它进程进行I/O传输过程中。

## 七. 了解盘交换区空间的管理方法。

- 盘交换区用来存放从内存暂时换出的数据页
- 每个盘交换区都由一组4KB的页槽组成。
- 盘交换区的第一个页槽用来存放该交换区的有关信息，有相应的描述符。

- 存放在磁盘分区中的交换区只有一个子区，存放在普通文件中的交换区可能有多个子区，原因是磁盘上的文件不要求连续存放。
- 内核尽力把换出的页存放在相邻的页槽中，减少访问交换区时磁盘的寻道时间。

# Linux文件系统

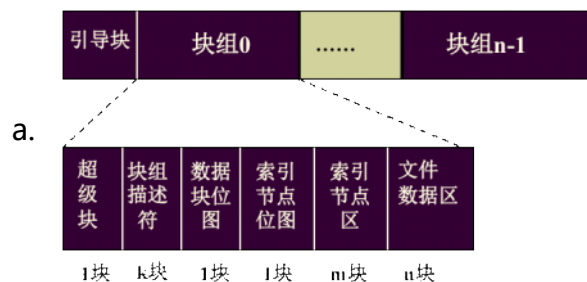
2017年12月12日 星期二 18:32

## 一. Ext2文件卷的布局？各部分的作用是什么？（文件保护方法，填空）

书186

- Ext2把磁盘块分为组，每组包含存放在相邻磁道的数据块和索引节点。块组的大小相等并顺序安排。
- Ext2用“块组描述符”来描述这些块组本身的结构信息，同时将超级块和所有的块组描述符重复存储于每个块组中。
- Ext2通过“位图”来管理每个块组中的磁盘块和索引节点。盘块位图，索引节点位图。

Ext2示意图



## ★ 二. Linux系统把一般的文件目录项分成哪两部分？这样做的好处是什么？（简答）

把通常的文件目录项分成简单目录项和索引节点两部分。

简单目录项包含了文件名和索引节点号等，可以提高文件目录的检索速度。

系统只保留一个索引节点，就可实现多条路径共享文件，减少信息冗余。

## □★三. Linux文件系统的索引节点中，索引表划分成几级？文件的索引表是如何增长的？要求能够利用索引表实现将文件中的字节地址转换成文件的物理块的操作。（作业）

索引表共15个元素，分成四级，12个直接索引项，1次间接索引项，2次间接索引项，3次间接索引项。

- 最初的12个元素是直接索引项，给出文件最初的12个逻辑块号对应的物理块号。
- 索引12是一次间接索引块，是一个存放盘块号的一维数组。对应的文件逻辑块号从12到  $(b/4) + 11$ ， $b$ 是盘块大小，每个逻辑块号占4B。
- 索引13是二次间接索引块，对应的文件逻辑块号从  $b/4 + 12$  到  $(b/4)^2 + (b/4) + 11$ 。
- 索引14是三次间接索引块，对应的文件逻辑块号从  $(b/4)^2 + (b/4) + 12$  到  $(b/4)^3 + (b/4)^2 + (b/4) + 11$ 。

## 四. 硬链接和符号链接的区别？

符号链接不与文件的索引节点建立链接，

同时这种链接允许在不同的文件系统之间建立。

## 五. Linux文件系统如何管理空闲存储空间？

- 磁盘块和索引节点的分配和回收
- 尽量在一个组块中的：
  - 文件的数据块和其索引节点
  - 文件和它的目录项
  - 父目录和子目录
- 每个文件的数据块尽量连续存放。

包括空闲块管理和空闲索引节点的管理。均采用位示图实现磁盘块和磁盘索引节点的分配和回收。P196

## ? 六. VFS通用文件模型中的四个主要对象? (填空)

- 超级块对象: Linux为每个安装好的文件系统都建立一个超级块对象。
- 索引节点对象: 代表一个文件, 对应于存放在磁盘上的文件控制块。
- 目录项对象: 代表一个目录项, 是一个文件路径的组成部分, 存放目录项与对应文件进行链接的信息。
- 文件对象: 记录了进程与打开的文件之间的交互信息。

## ? 七. Linux系统中, 进程打开一个磁盘文件要涉及哪些数据结构? 它们各有哪些关键字段? 他们的作用是什么? 参考图10.2

- `struct fs_struct *fs;` 指向文件系统信息  
`struct dentry *root, *pwd;` 每个进程都有自己的当前工作目录和根目录, 通过这两个目录与文件系统进行交互。
- `struct files_struct *files;` 指向进程打开文件信息  
`struct file **fd;` 指向文件对象指针数组  
`struct file *fd_array[ ];` 文件对象指针数组, 该数组包含32/64个文件对象指针, 若不够用, 内核就分配一个更大的数组, 地址放在fd字段。每个进程最多同时打开的文件数为1024个。

## 八. 一个文件在使用与不用时各占用系统哪些资源?

## 九. 了解安装表的作用

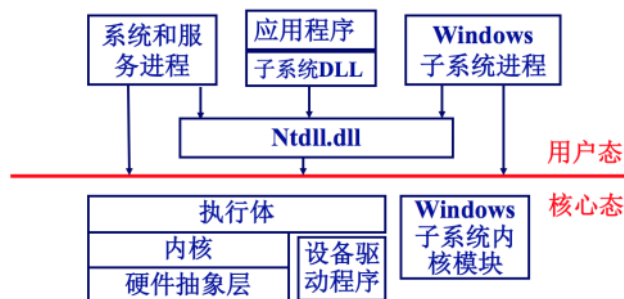
将安装点与被安装的文件系统信息保存。

# Windows 2000/XP模型

2017年12月11日 星期一 13:08

## 一. Windows 采用什么样的体系结构？

书264



## 二. 硬件抽象层HAL的作用是什么？

- HAL隐藏各种与硬件有关的细节。
  - 使内核、设备驱动程序和执行体免受特殊硬件平台差异的影响。
- 系统可移植性好。

## 三. Windows系统组件的基本机制。（填空）

- 陷阱调度
  - ①. 是操作系统处理意外事件的硬件机制。
  - ②. 当硬件或软件检测到异常或中断发生时，
    - 将暂停正在处理的事情，
    - 把控制转交给内核的陷阱处理程序。
    - 陷阱处理程序检测异常和中断的类型，
    - 并将控制转交给相应的处理程序。
  - ③. 中断是异步事件；异常是同步事件，系统服务调用被视为异常。
- 执行体对象管理器
  - 管理操作系统内的所有对象。
  - 有两种类型的对象：执行体对象、内核对象
  - 对象管理器的工作就是跟踪所有对象，便于对象访问的安全检测。
- 同步（自旋锁、内核调度程序对象）
- 本地过程调用LPC

## 四. 理解：延迟过程调用DPC，异步过程调用APC

- 延迟过程调用DPC：（选择）
  - DPC被用来执行一些相对于当前高优先级的任务来说不那么紧急的任务。
  - 有时内核在进行系统嵌套调用时，检测到应该进行重调度。为了保证调度的正确性，内核用DPC来延迟请求调度的产生。
  - 硬件中断服务例程可以把一些相对不紧急的事情放到一个DPC对象中处理，从而缩短处理机停留在高IRQL的时间。
  - DPC队列是一种机制，它能记住有哪些工作尚未处理。
  - 当IRQL降低到DPC/Dispatcher级别以下时，DPC中断就产生。调度程序依次执行DPC队列中的每个例程，直至DPC队列为空。
  - DPC队列是系统范围的。
- 异步过程调用APC：
  - 为用户程序和系统代码提供了一种在特定用户线程环境中执行代码的方法。
  - 每个线程都有自己的APC队列。APC队列也由内核管理。
  - 如果需要从内核空间复制一个缓冲区到用户进程地址空间，那么复制过程需要在用户进



程上下文进行，这样页表才能包含内核缓冲区和用户缓冲区。

## 五. Windows中有哪些对象，都有什么作用？

- 两种类型对象：执行体对象和内核对象。
- 执行体组件：进程和线程管理器、内存管理器、I/O管理器、对象管理等。
- 内核对象：由内核实现的一个初级对象集，对用户态代码不可见，仅供执行体使用。
- 一个执行体对象可以包含一个或多个内核对象。

## 六. 在多处理机系统中，提供了哪些同步和互斥机制？

- 互斥：内核引入自旋锁实现多处理机互斥机制。
- 同步：内核以内核对象的形式给执行体提供其他的同步机构——“调度程序对象”，
  - 包括：进程对象、线程对象、事件对象、信号量对象、互斥体对象、可等待的定时器对象及文件对象等。
  - 每个同步对象都有“有信号”或“无信号”两种状态。

## 七. 线程如何实现等待一个同步对象的操作？

线程通过调用由对象管理器提供的WaitForSingleObject(), WaitForMultipleObjects()系统服务来与调度程序对象同步。

每个处于等待状态的线程都有等待块列表，记录该线程正在等待的对象的情况。

每个调度程序头都有一个等待块列表头指针，将等待该对象的所有线程的等待块连接在一起。

# Windows 进程和线程管理

2017年12月11日 星期一 21:58

## 一. 管理进程和线程的数据结构：（填空）

- 执行体进程块EPROCESS、
- 执行体线程块ETHREAD、
  - 线程是内核支持的线程，是处理器调度的对象。每个线程都有一个执行体线程块。
- 内核进程块KPROCESS、
  - 它是调度程序对象。
  - 包括：基本优先级、默认时间片、进程的转锁、进程所在处理机簇、进程状态、进程中的线程的内核总时间和用户总时间、进程页目录指针、属于该进程的所有线程的内核线程块队列指针等。
  - 进程页目录的物理地址被保存在KPROCESS块中
- 内核线程块KTHREAD。

## 二. 创建进程：CreateProcess()； 创建线程：CreateThread()

- CreateProcess 创建新进程及其主线程，执行指定的程序。

主要流程：

- 打开将在进程中执行的映像文件(.exe)，创建一个区域对象，建立映像与主存之间的映射关系。
- 创建执行体进程对象，包括申请并初始化执行体进程控制块，创建并初始化进程地址空间、内核进程块和进程环境块等。
- 创建一个主线程。
- 通知Win32子系统，对新进程和线程进行一系列初始化。
- 完成地址空间的初始化，开始执行程序。

- CreateThread 创建新线程

主要流程：书286

## 三. 线程的7种状态，及其解释。

- 就绪状态(ready)
- 备用状态(standby)。已选好处理机，正等待描述表切换，以便进入运行状态。
- 运行状态(Running)
- 等待状态(waiting)
- 传输状态(transition)。核心栈被调到外存的就绪态。
- 终止状态(terminated)
- 初始化状态(Initialized)。正在创建过程中。



## 四. 线程调度:

a. 基于优先级的抢先式的多处理机调度系统。

b. 线程调度程序的数据结构:

负责记录各线程的状态。

- 32个就绪队列。每个优先级对应一个。
- 32位掩码的就绪位图。每一位指示一个调度优先级的就绪队列中是否有线程等待运行。
- 32位掩码的空闲位图。每一位指示一个处理机是否处于空闲状态。

## 五. 线程优先级的提升时机。

- I/O操作完成后的线程。(选择)
- 信号量或事件等待结束的线程。
- 前台进程中的线程完成一个等待操作。
- 由于窗口活动而唤醒图形用户接口线程。
- 线程处于就绪状态超过一定时间, 仍未能进入运行状态(处理器饥饿)。

# Windows 存储器管理 CH-16

2017年12月12日 星期二 15:31

## 一. 两种数据结构：虚拟地址描述符VAD、区域对象，这两种结构各有什么作用？

**虚拟地址描述VAD：**

- 当线程要求分配一块连续虚存时，系统并不立即为其构造页表，而是为它建立一个VAD结构。
- VAD结构：被分配的地址域、该域是共享的还是私有的、该域的存取保护以及是否可继承等信息。
- 进程页表的构建一直推迟到访问页时才建立。（“懒惰”方式）

**区域对象：**

- 利用区域对象将一个可执行文件装入主存。
- 使用区域对象可将一个大于进程地址空间的文件映射到进程地址空间。
- 缓存管理器利用区域对象访问一个被缓存文件中的数据。

## 二. 虚存内存区域：空闲的、保留的、提交的

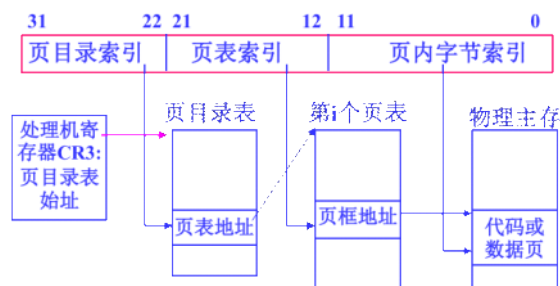
- 进程私有的2G地址空间的页可能是空闲的，或被保留，或被提交。
- 被保留：已预留虚存，还没分配物理主存
- 被提交：已分配物理主存或交换区。
- 分配主存时，可以先保留地址空间，后提交物理主存；也允许保留和提交同时实现。
- 第一阶段只保留地址空间，特别适合线程正在创建大的动态数据结构的情况。

## 三. 32位逻辑地址，二级页表。页目录表项和页表项具有相同的数据结构，该数据结构包含哪些数据项？进程页表建立的时机。进程的地址转换过程。

书307-308

### 2. 虚拟地址变换过程

32位的地址被分解为三部分：

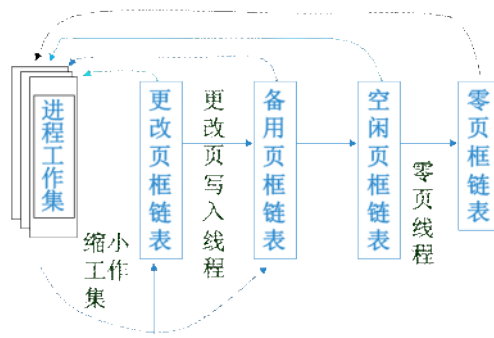


## 四. 管理物理内存的数据结构：页框数据库。页框的8种状态：活动、转换、备用、更改、更改不写入、空闲、零初始化、坏，页框的状态转换图

### 16.9. 原型页表项的概念。

书308-311

页框的状态转换图



## 五. Windows采用的页替换策略是什么？

在多处理器系统中，采用了局部先进先出置换策略。

而在单处理器系统中，更接近于最近最久未使用策略(LRU，也称为“时钟页面置换算法”)。

## 六. 原型页表项，区域对象的页表。虚拟页式中，采用原型页表实现多进程共享页。（简述）

# Windows文件系统

2017年12月12日 星期二 18:35

## 一. Windows所支持的文件系统类型有哪些？

FAT, NTFS

## 二. 虚拟簇号和逻辑簇号的概念。

- FAT和NTFS将卷划分成若干簇，并从卷头到卷尾进行编号，称为逻辑簇号(LCN)。
- 通过索引表建立文件的虚拟簇号(VCN)与磁盘的逻辑簇号之间的映射。

## 三. NTFS卷的结构，主控文件表MFT的作用。（简述）

分区引导扇区、主控文件表区，文件数据区

- **MFT**
  - 是NTFS卷的管理控制中心
  - 包含了卷上所有的文件、目录及空闲未用盘簇的管理信息

## 四. NTFS文件的物理结构：索引顺序结构。

## 五. 管理文件的目录结构采用B-树。

NTFS使用B-树结构来存放文件和子目录信息。



# 心得

2017年12月14日 星期四 21:01

操作系统这课，我自己学的非常的不好啊，没什么发言权，自己感觉看了很久的书很多东西看了半天也记不住。还是应该找重点理解。无论是理论部分还是windows和linux部分。

重点就是复习提纲。把上面提到的都理解了。一定要提早理解。到最后一起复习压力挺大的，而且还记不太住，那些简答和填空，都要背下来，每年都换着考。

试卷，这个考试并不太套路，试卷就是提供一个复习方向。告诉你这个考试还挺难的。所以我只放了我觉得算是值得看看的一套，其余的也没什么，好像也不是咱们学校的。

另外我把我这一年的考点在之前每章的复习文件里标注了我记得的部分。不