

408必胜宝典

数据结构

计组

操作系统

第一章

操作系统的提供的接口：

- 命令接口
 - 联机命令接口——说一句做一句
 - 脱机命令接口——说一堆做一堆
- 程序接口——系统调用
- GUI

用户态约核心态

核心态：也叫做管态

用户态：也叫做目态

用户态转向和心态的例子：

1. 发生一次中断
2. 用户程序中产生了一个错误状态
3. 用户企图执行一个特权指令

中断的分类：

- 内中断：来自CPU内部
 - 外中断：来自CPU外部
- 中断的本质：将CPU的控制权交给OS

中断处理和子程序调用的区别：

调用类型	中断	子程序调用
------	----	-------

调用类型	中断	子程序调用
入口地址	由中断隐指令根据中断向量得到	由调用程序根据寻址方式得到
保护环境	保存PC,PSW,通用寄存器	保存PC,通用寄存器
进程状态	从用户态转换为核心态	没有发生变化

有中断请求时，先由**中断隐指令**完成中断前程序状态保存，主要工作有：

1. **关中断**
2. 保存PC,PSW
3. 根据中断向量引出对应的中断服务程序(即中断处理程序，属于操作系统内核),**通用寄存器保护由中断服务程序完成(OS)**。当中断服务程序结束后再**开中断**
当中断服务程序执行的时候已经转换到了核心态

对于支持多重中断的OS来说步骤如下：

- 硬件完成：
 - 关中断
 - 保存断点，PC和PSW
 - 终端服务程序寻址
- 中断服务程序完成：
 - 保存现场和屏蔽字，主要是保存通用寄存器
 - 开中断(此时如果有更高级别的中断请求，可以转去处理那个中断)
 - 执行中断服务程序——这个是中断的目的
 - 关中断(准备恢复现场，回到中断之前)
 - 恢复现场和屏蔽字
 - 开中断
 - 中断返回

系统调用

什么时候需要系统调用？

如果一个操作可能会影响到其他进程，那么就需要借助OS来完成，此时需要系统调用

系统调用的过程：

1. 传递系统调用的参数,用户进程会把这次系统调用需要传递的参数放在某些寄存器中
2. 执行陷入(trap)指令，这个陷入指令的执行会引发一个内中断，之后CPU的控制权就交给OS
3. 执行相应的服务程序，OS在执行相关的系统调用程序时从对应的寄存器中取出用户传递的参数，然后完成用户请求的操作。

4. 返回用户态，系统调用结束后，OS把CPU的控制权还给用户

陷入指令，又称访管指令，trap指令。用于系统调用，用户程序通过陷入指令主动将CPU控制权还给操作系统。执行陷入之后，CPU状态从用户态转到核心态

第二章 进程管理

命题重点：

1. 进程和线程的比较，内核支持线程和用户级线程
2. 进程的状态变化，进程的创建与终止，进程的阻塞与唤醒
3. 作业运行顺序，各种调度算法，特别是高响应比算法和多级反馈队列算法的处理
4. 进程的同步和互斥，信号量机制
5. 死锁的判断，安全序列，银行家算法

进程与线程的比较

进程通信：

- 共享存储：在通信的进程之间存在一块可以直接访问的共享空间，通过对这个共享空间进行读/写操作来实现进程之间的信息交换。
 - 低级信息共享：基于数据结构的共享
 - 高级信息共享：基于存储区域的共享
- 消息传递：进程之间的数据交换是以格式化的消息(Message)为单位的。若通信的进程之间不存在可以直接访问的共享空间，则必须使用OS提供的消息传递方法实现进程通信。
 - 直接通信方式：发送进程直接把消息发送给接收进程，并将它挂在接受进程的消息缓冲队列上，接受进程从消息缓冲队列中取得消息。
 - 间接通信：发送进程把消息发到某个中间实体，接受进程从中间实体获取消息，中间实体也叫做信箱。所以也叫做信箱通信方式。
- 管道通信方式：

用户级线程 VS 内核级线程

将n个用户级线程映射到m个内核级线程上。

操作系统只能“看见”内核级线程，因此只有内核级线程才是处理机分配的单位。

例如：一个进程又两个内核级线程，三个用户级线程，在用户看来，这个进程有三个进程。但即使该进程在一个4核处理机的计算机上运行，也最多只能被分配到两个核，最多只能有两个用户级线程并行执行。

在多处理机系统中，内核能够同时调度同一进程中多个线程并行执行

进程调度与进程控制

引发进程创建的事件：

- 终端用户登录系统
- 作业调度
- 系统提供服务
- 用户程序的应用请求——创建子进程

创建进程的步骤：

1. 申请空白的PCB,并初始化PCB
2. 为新进程分配资源

引发进程终止的事件：

- 正常结束
 - 异常结束，由于发生异常而终止进程
 - 外界干扰，进程因外界请求而终止运行
- 销毁进程的步骤：
- 释放进程占用的资源
 - 将该进程的PCB从所在的队列中删除

进程状态与进场控制

引发进程阻塞的事件：

- 请求系统服务
- 启动某种操作
- 新数据尚未到达
- 无新工作可以做

阻塞是主动行为

引发进程唤醒的事件：

- 引发阻塞的事件完成

处理机调度

周转时间：做完完成时间-作业提交时间

带权周转时间=作业周转时间/作业实际运行时间

注意：区分作业的周转时间和进程的周转时间不一样。

进程调度的优先级：

- 系统进程的优先级高于用户进程

- 前台进程的优先级高于后台进程
- I/O繁忙型进程的优先级高于计算型进程

不能进程处理机调度的情况：

- 在处理中断的过程中
- 进程在**操作系统内核程序临界区**
- 其他需要完全屏蔽中断的原子操作过程中

如果在上述过程中引发了调度条件，并不能马上进行调度和切换，应置系统的请求调度标志，直到上述过程结束后才进行相应的调度和切换

调度算法

- FIFS——有利于长作业，有利于CPU繁忙型作业，不利于I/O繁忙型作业
- 短作业优先——对长作业不利，平均等待时间、平均周转时间最少
- 高响应比算法——有利于短作业，长作业不至于产生饥饿，等待时间越长，优先级越高

进程同步与互斥

互斥访问的原则：

- 空闲让进
- 忙则等待
- 有限等待
- 让全等待

单标志法:违背空闲让进

双标志前检查法：违背忙则等待

双标志后检查法：违背空闲让进，忙则等待

Peterson:不满足让权等待

记录型信号量：

经典的同步问题

- 生产者——消费者
- 读者——写者问题
- 哲学家进餐问题

死锁

死锁条件：

- 互斥条件

- 不剥夺条件
- 请求并保持
- 循环等待条件

预防死锁

避免死锁

死锁检测和接触

资源分配图

内存管理的命题重点

- 程序装入过程与原理，编译与链接，逻辑地址与物理地址
- 连续分配方式的原理和特点，动态分区分配算法的原理
- 分页存储管理的页表机制，分段存储管理的段表机制，分页与分段的比较
- 虚拟存储的原理，特征，缺页中断的处理过程与特点，虚拟地址到物理地址的转换，引入快表后虚拟地址的变换过程，各种页面置换算法的原理，页面分配策略，工作集的定义，抖动产生的原因和解决方法。

程序运行的基本原理

- 编译，把源代码编译成目标模块，每个目标模块各自具有逻辑地址空间
- 链接，由链接程序将上述目标模块，以及所需库函数链接，形成具有完整逻辑地址空间的装入模块，也就是一个可执行文件
- 装入，由装入程序把装入模块装入内存

分页存储管理

硬件支持：页表寄存器(PTR),存放页表在内存的始地址和页表长度

页表的开始地址和页表长度平时是存放在进程的PCB中的，只有进程上处理机运行时才会放在寄存器中。

缺页中断机构

缺页中断的处理过程：

缺页中断的目的是要将位于外存上的代码或数据装入内存，此时应将缺页的进程阻塞，如果内存中有空闲的也，则分配一块，将要调入的页装入该块，并修改页表中相应的页表项，此时若内存中没有空闲块，则需淘汰一个页。

缺页中断作为中断页同样要经历，诸如CPU环境，分析中断原因，转入缺页中断处理程序，恢复CPU环境等几个步骤。但与一般的中断相比，它有以下几个明显的区别：

- 在指令执行期间产生和处理中断信号，而非一条指令执行结束后

- 在一条指令执行期间可能会发生多次缺页中断

文件系统

文件管理的命题重点：

- 绝对目录与相对目录，文件控制块，文件共享的两种方法，文件的打开与关闭
- 问价按几种物理结构及特点，FCB的存储方式及原因，混合索引分配方式
- 各种磁盘调度算法，特别是SCAN和CSCAN算法

文件控制块FCB

一个FCB就是一个目录项，FCB和目录的目的是为了实现"按名存取"

文件共享

- 硬链接——指针指向
- 符号链接——Windows中的快捷方式

文件的打开

操作系统维护一个打开文件表(open-file table)，当需要一个文件操作时，可以通过该表的一个索引指定文件，就省略了搜索环节。

如果调用open的请求得到允许，进程就可以打开文件，而open通常返回一个指向打开文件表中的一个条目的指针。通过使用该指针(而非文件名)进行IO操作，以简化步骤并节省资源

文件的实现

文件分配：

- 连续分配——可以随机分配，但是需要连续的空间
- 隐式链接分配——方便删除和插入，只能顺序访问，稳定性存在问题
- 显示链接分配——FAT
- 索引分配——把每个文件的所有盘块号都几种放在一起构成索引块(表)。每个文件都有其索引块，这是一个磁盘块的地址数组。每个索引块的第i个条目指向文件的第i个块。目录条目包括索引块的地址。

文件的盘块索引存放的叫索引块，存放文件数据的块叫数据块

第五章 设备管理

命题重点：

- 各种I/O控制方式，特点及适用情况
- I/O软件的层次结构，设备无关性的原理
- I/O调度，单缓冲区和双缓冲区的原理及性能分析

I/O文件的层次结构

- 用户层I/O软件——系统提供的API
- 设备独立性软件——屏蔽设备的差异性，进行一些公有的操作
- 设备驱动程序——根据用户调用，驱动特定的设备开始工作，一个类型的设备需要一个驱动设备，向硬件设备发出一些具体的操作命令，将由设备控制器发来的信号传送给上层软件
设备驱动程序的功能
- *
 - 中断处理程序——用于保存被中断进程的CPU环境，转入相应的中断处理程序进行处理，处理完并恢复被中断进程的现场后，返回到被中断的进程
 - 硬件设备

中断处理层的主要工作有：进行进程的上下文切换，对处理中断信号源进行测试，读取设备状态和修改进程状态等

引入控制器后，系统可以通过几个简单的参数完成对控制器的操作，而具体的硬件操作则由控制器调用相应的设备接口完成，使CPU从繁重的设备控制操作中解放出来

单缓冲区和双缓冲区

单缓冲： $\text{Max}(C, T) + M$

双缓冲区： $\text{Max}(C + M, T)$

计网