

原 《计算机操作系统（第四版）》知识点归纳

大二期末要闭卷考计算机操作系统，于是自己总结归纳了一些内容，希望后来的同道之人可以省点时间~（做完归纳，考试就不怕了）

说明：一开始是少量知识点以“填空题”的形式记录，后面的内容是按章节归纳

对于批处理作业，必须提供相应的作业控制信息

操作系统的主要功能是管理计算机系统中的（资源）

分时系统的响应时间（及时性）主要是根据（用户所能接受的等待时间）决定的

推动分时系统形成和发展的动力是（方便用户）//人机交互

多道批OS动力是（提高资源利用率和OS吞吐量）

采用（微内核）结构时，将OS分成用于实现OS最基本功能的内核和提供各种服务的服务器两个部分

Windows 95是（单用户多任务）操作系统。

系统专门为运行中的应用进程创建新进程的事件是（提供服务）。

若进程正处于执行状态时，因终端的请求而暂停下来以便研究其运行情况，这时进程应转变为（静止就绪）状态

在引入线程的操作系统中，资源分配和调度的基本单位是（进程）。

死锁的预防是通过破坏产生死锁的四个必要条件来实现的。下列方法中，（一次性分配策略）破坏了“请求与保持”条件。

为了使作业的平均周转时间最短，应采用（短作业优先）算法

为了使短作业、长作业及交互作业用户都比较满意，应采用（多级反馈队列调度算法）。

下列算法中，（时间片轮转法）只能采用抢占调度方式

在动态优先权中，随着进程执行时间的增加，其优先权将降低

死锁的预防是通过破坏产生死锁的四个必要条件来实现的。下列方法中，（资源有序分配策略）破坏了“循环等待”条件。

在分时系统中除了进程调度以外，通常还设置了（中级调度）。

交换技术（还是）现代操作系统中常用的技术。

动态重定位是在作业（装入过程）中进行的

适用于动态链接的存储方式是（分段存储管理）。

（固定分区）支持多道程序设计，管理最简单，但存储碎片多

非虚拟存储器的论述：要求作业在运行前必须全部装入内存，且在运行过程中也必须一直驻留内存

虚拟存储器的论述：在请求段页式系统中，以段为单位管理用户的虚空间，以页为单位管理内存空间

由于有了虚拟存储器，于是允许用户使用比内存更大的地址空间

一个计算机系统的虚拟存储器的最大容量是由（计算机的地址结构）确定的，其实际容量是由（内存和硬盘容量之和）确定的

分段的尺寸受内存空间的限制，但作业总的尺寸不受内存空间的限制

在UNIX系统中，对物理块的分配是采用（可变分配全局置换）策略

在UNIX系统中，如果被换出的页的内容并未被修改，但对换区中尚无拷贝，此时应将它（换出到对换区）；若被换出的页的内容已被修改，应将它（换出到对换区）

第一章 引论

①为什么发明计算机系统：方便、有效、可扩充、开放

计算机系统作用：做接口、管理资源、资源的抽象

发展计算机系统的动力：提高利用率、更加方便、应用、体系、硬件更新都要跟上

②计算机系统发展史

一、无操作系统

（一）人工操作：单用户、CPU、内存长期空闲

（二）脱机输入/输出（OFF-LINE I/O）：装好卡片再上机。节约CPU空闲时间、提高I/O速度

二、单道批操作系统

描述：有个监督程序将磁带上的作业调入计算机

缺点：I/O太慢，CPU太快

三、多道批操作系统

描述：A在I/O，B趁机CPU

优点：肯定提高资源利用率、系统吞吐量变大

缺点：每个程序都要很久才处理完、无交互能力

未解难题：内存、外存、I/O、CPU、作业管理、接口

四、分时系统

描述：解决人机交互问题

优点：终于有人机交互、多用户共享主机

实际问题：由于多用户，所以要有“多路卡”、作业直接入内存、有个“时间片”调度作业

特征：多路、独立、及时（用户可接受）、交互

五、实时系统

描述：工业（武器）控制系统、信息查询系统、多媒体系统、嵌入式系统

类型1：周期性实时：真的很周期；非周期性实时：有开始截止时间和完成截止时间

类型2：硬实时：工业、武器系统；软实时：信息查询系统和多媒体系统

与分时系统比较：多路、独立、及时（毫秒级）、交互、可靠

六、微机时代

（一）单用户单任务：8位机的CP/M、16位机的MS-DOS

（二）单用户多任务：目前的32位系统，如Windows

（三）多用户多任务：UNIX、Solaris、Linux

⑥操作系统共同特性：

一、并发

（一）并发和并行宏观上一样，

并发：单处理机系统，微观上交替运行

并行：多处理机系统，微观上同时运行

（二）引入进程

进程：在系统中能独立运行并作为资源分配的基本单位，由机器指令、数据和堆栈等组成，能独立运行的活动实体

特点：用进程就可以并发执行了

二、共享

（一）互斥共享方式

例子：临界资源，打印机、磁带机

描述：你要先申请才能获得资源

（二）同时访问方式

描述：微观上还是并发

例子：多用户磁盘设备

条件：系统允许进程并发、系统能有效管理资源

三、虚拟

（一）时分复用技术

虚拟处理机技术：分身之术

虚拟设备：又是分身之术，骗用户以为有专人服务

时分复用：速度： $\leq 1/N$

（二）空分复用技术

描述：将程序、电话线分成若干部分，然后各部分分时进入内存运行

空分复用：空间： $\leq 1/N$

四、异步

描述：因为要并发，所以需要有一个机制调度进程

⑥操作系统主要功能

一、处理机管理功能

（一）控制进程

描述：要并发，就要进程、要进程，就要管理

（二）进程同步

进程互斥方式：临界资源要互斥

进程同步方式：合作完成共同任务，同步机构要协调先后次序

（三）进程通信

描述：对合作进程而言，需要交换信息。当他们处于同一计算机系统时，通常采用直接通信的方式。

例子：输入进程、计算进程、打印进程，需要信息交换

（四）调度

作业调度：选择作业、建立进程、分配资源、插入就绪队列

进程调度：从就绪队列中选出进程，分配CPU

二、存储器管理功能

（一）内存分配

任务：分配空间、减少碎片、追加内存空间

方式：静态分配，装入内存时确定，不允许追加、不允许移动；动态分配，允许追加、允许移动

（二）内存保护

任务1：每道程序只在自己的内存空间运行，互不干扰

任务2：不允许用户程序访问操作系统程序和数据、也不允许用户程序转移到非共享的其他用户程序中执行

（三）地址映射

任务：存储器要负责地址映射，在硬件支持下完成

（四）内存扩充

描述：用虚拟存储技术，从逻辑上扩充内存容量

任务1：请求-调入功能

任务2：置换功能

三、设备管理功能

任务1：完成用户进程的I/O请求：分配I/O设备，完成I/O操作

任务2：提高CPU和I/O利用率：提高I/O速度，方便用户使用I/O设备

（一）缓冲管理

描述：在内存中设置缓冲区

例子：单缓冲机制、双向同时传送数据的双缓冲机制、多个设备共同使用的公用“缓冲池”机制

（二）设备分配

描述：在系统中设置“设备控制表”、“控制器控制表”等数据结构，用于记录设备和控制器等标识符和状态。根据表就知道指定设备当前是否可用、忙碌。分配时，针对

（三）设备处理

描述：CPU向设备控制器发出I/O命令，要求完成I/O操作、反之，CPU接收控制器发出的中断请求，并响应、处理

四、文件管理功能

描述：管理用户、系统文件，方便使用；保证安全性

（一）文件储存空间管理

背景：多用户环境下，用户自己管理文件存储，会困难和低效

任务1：为每个文件分配外存空间、提高外存利用率、进而提高存取速度

任务2：系统中设置数据结构，记录文件存储空间使用情况，以供分配时参考

任务3：分配和回收

（二）目录管理

任务1：为每个文件建立目录项，包括文件名、属性、物理位置等，以实现按名存取

任务2：实现文件共享。

任务3：提供目录查询手段

（三）文件读/写管理和保护

文件读/写管理：根据用户请求，从外存中读取数据，或将数据写入外存

文件保护：防止未经核准的用户存取文件、防止冒名顶替存取文件、防止以不正确方式使用文件

五、操作系统与用户之间的接口

（一）用户接口

描述：方便用户直接、间接控制自己的作业

联机用户接口：等待用户键入命令

脱机用户接口：一开始就提供作业说明书，直到作业结束语句

图形用户接口：移动鼠标选择菜单项

（二）程序接口

描述：旧系统用汇编语言写，所以只有汇编语言的才能直接使用系统调用；如果是高级语言，就用一一对应的库函数

六、现代操作系统的新功能

（一）系统安全

描述：确保存储和传送数据的保密性、完整性和系统可用性，要用几种技术

技术：认证技术、密码技术、访问控制技术、反病毒技术

（二）网络的功能和服务

功能：网络通信、资源管理、应用互操作

（三）支持多媒体

功能：接纳控制功能、实时调度、多媒体文件的存储

§OS结构设计

一、传统操作系统结构

（一）无结构操作系统

又名：整体系统结构

（二）模块化结构OS

基本概念：

又名：模块-接口法

描述：有模块、子模块、接口

模块独立性：

标准：内聚性越高，模块独立性越高、耦合度越低，模块独立性越高

优点：提高设计正确性、可理解性和可维护性、增强可适应性、加快加速过程

缺点：接口难以满足需求、无序

（三）分层式结构OS

基本概念：有序分层，自底向上法铺设中间层

优点：易保证系统正确性、易扩充和易维护

缺点：系统效率降低

二、客户/服务器模式(Client/Server Model)简介

（一）客户/服务器模式的由来、组成和类型

组成：客户机、服务器、网络系统

（二）客户/服务器之间的交互

描述：客户发送请求消息、服务器接收消息、服务器回送消息、客户机接收消息

（三）客户/服务器模式的优点

描述：数据分布处理和存储、便于集中管理、灵活性和可扩充性、易于改编应用软件

三、面向对象的程序设计

（一）OOP的基本概念

描述：抽象，具体事物为对象

对象：封装好

对象类：创建多个相似对象

继承：继承父类，增加部分

（二）OOP的优点

描述：“重用”提高产品质量和生产率、使系统具有更好的易修改性和易扩展性、易于保证系统“正确性”和“可靠性”

四、微内核OS结构

描述：支持多处理机

例子：卡内基·梅隆的Mach OS、Windows 2000/XP

（一）基本概念

描述：足够小的内核、基于C/S模式、应用“机制与策略分离”原理、采用OOP技术

（二）基本功能

描述：进程管理、低级存储器管理、中断和陷入处理

（三）优点

描述：提高可扩展性、增强可靠性、可移植性强、提供对分布式系统的支持、融入OOP

（四）缺点

描述：效率降低

第二章 进程描述与控制

①前趋图与程序执行

一、前趋图与程序执行

（一）前趋图

描述：前一个做完，才到后一个做、禁止循环

二、顺序执行

描述：一个跟一个

特征：顺序、封闭（独占资源）、可再现

三、并发执行

描述：互不依赖才能并发执行

特征：间断、失去封闭、不可再现

②进程的描述

一、进程的定义和特征

进程实体：程序段、相关的数据段和PCB

定义：进程是进程实体的运行过程，是系统进行资源分配和调度的一个独立单位

特征：动态、并发、独立、异步

二、进程的基本状态及转换

进程的三基态：就绪（只欠CPU）、执行、阻塞（因故无法继续执行）

三态转换：如图

新增两态：创建状态、终止状态

五态转换：如图

三、挂起操作和进程状态的转换

挂起原因：终端用户需要、父进程请求、负荷调节、操作系统需要

引入挂起后的三态转换：如图

引入挂起后的五态转换：如图

四、进程管理中的数据结构

用于管理控制的数据结构：每个资源、进程都有一个数据结构用于表征实体——资源信息表、进程信息表，包括：标识、描述、状态等和一批指针，通过指针能够链接成队
分类：内存表、设备表、文件表、进程表

PCB的作用：作为独立运行基本单位的标识、能实现间断运行、提供进程管理所需的信息、实现与其他进程的同步与通信

PCB的信息：进程标识符（内外部）、处理机状态、进程调度信息、进程控制信息

PCB组织方式：线性方式、链接方式、索引方式

③进程控制

一、操作系统内核

描述：常驻内存的模块

目的：保护软件、提高OS运行效率

系统态、管态、内核态：高特权、访问所有寄存器、存储区、传统OS都在系统态运行

用户态、目态：低特权、执行指定指令、访问指定寄存器和存储区

支撑功能：中断处理、时钟管理、原语操作

资源管理功能：进程管理、存储器管理、设备管理

二、进程的创建

层次结构：UNIX有父子关系，Windows只有控制与被控制关系

进程图：描述家庭关系的图

引起创建进程的事件：用户登录、作业调度、提供服务（譬如打印）、应用请求

进程的创建：申请空白PCB、分配物理、逻辑资源、初始化PCB、如果能插入就绪，就插

三、进程的终止

引起进程终止的事件：正常结束、异常结束、外界干预

进程的终止过程：根据标识符、终止执行、立即调度、子孙终止、资源归还、移出队列

四、进程的阻塞与唤醒

引起进程阻塞和唤醒的事件：向系统请求共享资源失败、等待某操作完成、新数据尚未到达、等待新任务到达

进行阻塞过程：发生上述的某事件，就进入**block**过程，主动将状态改为阻塞，**PCB**插入阻塞队列（分类插入），处理机分配给另一就绪进程，切换，并保留被阻塞进程

进程唤醒过程：由释放资源的进程调用**wakeup**原语，即移出阻塞队列，合作/相关的进程中安排**wakeup**

五、进程的挂起与激活

进程的挂起：活动→静止，进程正在执行，就转向调度程序重新调度

进程的激活过程：从外存调入**active**原语到内存，检查进程现行状态，静止→活动

抢占调度策略：静止就绪进程→就绪队列，比较当前进程优先度，有机会立即剥夺当前进程运行

⑥进程同步

描述：能够并发、改善利用率、提高吞吐量、但使系统复杂

一、进程同步的基本概念

制约关系：间接相互制约关系、直接相互制约关系

间接相互制约关系：互斥共享

直接相互制约关系：合作共享，异步性要做好

临界资源：生产者-消费者问题、

临界区、：进入区、临界区、退出区、剩余区

同步机制应遵循的规则：空闲让进、忙则等待、有限等待、让权等待

二、硬件同步机制

关中断：缺点多：滥用关中断.造成严重后果、关中断时间过长、不适用于多CPU系统（因为一个处理器关中断并不能防止进程在其他处理器上执行相同的临界段代码）

Test-and-Set：不断测试**lock**，如果是**FALSE**，就进入临界区，并**lock == TRUE**；否则测试到**TS(s) == TRUE**

Swap指令：一直等，直到**key == TRUE**

但以上都不符合“让权等待”原则

三、信号量机制

整形信号量：**S≤0**，就一直等，直到释放互斥资源

记录型信号量：整形信号量不符合“让权等待”原则。如果有资源，就分配，如果无，就插入阻塞队列；释放资源，如果有等待，就激活

AND型信号量：一口气全分配

信号量集：有多个信号量（S信号量,至少要t个,每次分配d个）

四、信号量的应用

利用信号量实现进程互斥：**mutex = (-1, 0, 1) =**（无，一临一阻队，一临一信队）

利用信号量实现前趋关系：需要的信号量被占用了，就这样实现

五、管程机制

描述：为解决信号量机制分散、容易死锁的问题，发明新同步工具——管程

定义：定义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

组成：管程名称、数据结构的说明、对数据结构进行操作的过程、初始化的语句

特性：模块化、抽象数据类型、信息掩蔽

管程与进程不同：都有数据结构，一个公.一个私、管程操作同步.初始化.进程顺序执行、管程为解决互斥资源.进程实现并发性、进程调用管程.进程主动.管程被动、↑

条件变量：增加一个条件变量，万一发生意外，在管程中被挂起或被阻塞，下一个进程都可以继续执行

⑥经典进程的同步问题

一、生产者-消费者问题

记录型信号量解决：如果缓冲区空，而且能够获取信号量，就投放产品；如果缓冲区有产品，而且能够获取信号量，就消费

AND信号量解决：一口气全分配

管程解决：利用管程只有一个进程能够使用的属性

二、哲学家进餐问题

记录型信号量解决：先拿左.后那右、先放左.后放右

解决死锁：最多4人取筷子、先检查.有左右筷子才能取、奇左右.偶右左

AND信号量解决：一口气全分配

三、读者-写者问题

描述：可以多读一、一旦开始写.就不能读或写

记录型信号量解决：

读操作：等**rmutex**就是为了改**readcount**→无人读?看看是否在写.等**wmutex**→**readcount++**→自增完成.**rmutex**还你→读读读→等**rmutex**为了自减**readcount**→无人读

写操作：等**wmutex**.即无读无写→写完.还你**wmutex**

利用信号量集机制：

读：限制**reader**个数→如果**mx**是1.就读→最后释放一个**reader**个数

写：如果**mx**是1.并且读者数为0.就写→写完释放**mx**

⑥进程通信

一、进程通信类型

共享存储器系统：某些数据结构和共享存储区、管道通信系统、消息传递系统、C-S系统

二、消息传递通信的实现方式

（一）直接消息传递系统

1.直接通信原语：对称寻址方式、非对称寻址方式

2. 消息格式：较短的减少系统处理和存储的开销、较长可以方便
3. 进程同步方式：发塞收塞（进程间紧密同步、无缓冲）、发通收塞（平常状态）、发通收通（发生某事件无法继续运行）、（无发塞收通）
4. 通信链路：用“建立连接”原语建立通信链路，用完拆、用“发送命令”原语建立链路，还分单向和双向

（二）信箱通信（间接）

1. 定义：是数据结构，分信箱头和信箱体
2. 原语：创建和撤销，发送和接收
3. 类型：私用、公用（操作系统创建）、共享（进程创建）
4. 进程之间的关系：一对一、多对一、一对多、多对多

三、直接消息传递系统实例

消息缓冲队列通信机制中的数据结构：利用数据结构式消息缓冲区、在PCB增加有关通信的数据项

原语：设置发送区、申请PCB(B)的缓冲区i、复制到缓冲区、插入消息队列、移出消息队列、复制到接收区、释放缓冲区

⑥线程的基本概念

描述：就是为了提高程序并发执行的程度

一、线程的引入

进程的两个基本属性：进程是一个可拥有资源的独立单位、进程同时是一个可独立调度和分派的基本单位

进程并发执行所需的时空开销：创建进程、撤销进程、进程切换

线程——作为调度和分派的基本单位：线程轻装上阵

二、线程与进程比较

调度的基本单位：线程是调度和分派的基本单位、跨进程，会切换进程

并发性：线程的合作，能够并发

拥有资源：有TCB，但只是必不可少、保证独立运行的资源

独立性：同一进程的不同线程共享进程的内存地址空间和资源

系统开销：因为轻装，所以减少开销、提升速度

支持多处理机系统：对多线程进程，多个线程可以分配到多个处理机上

三、线程的状态和线程控制块

线程运行的三个状态：和进程一样

线程控制块TCB：标识符、一组寄存器、运行状态、优先级、线程私有存储区、信号屏蔽、堆栈指针

多线程OS中的进程属性：进程是可拥有资源的基本单位、多个线程可并发执行、进程已不是可执行的实体

⑥线程的实现

一、线程的实现方式

内核支持线程KLT：

优点：内核调度同一进程多个线程并行执行、一个线程阻塞，其他线程占有处理机、支持小数据结构和堆栈，切换较快开销小、内核本身采用多线程技术，提高系统执行速度
用户级线程ULT：

优点：无需内核，节省模式切换的开销、调度算法进程专用、与OS无关，甚至可以在操作系统平台实现

缺点：一个线程阻塞，同进程的其他线程都会塞、只有一个CPU，只有一个线程能执行、按进程分配，不公平

组合方式：

多对一模型：优点：开销小、缺点：一塞进程全塞、只有一线程访问内核、多线程不能同时在多个处理机上运行

一对一模型：一个用户级线程映射到一个内核支持线程

多对多模型：一对一和多对一的结合

二、线程的实现

内核支持线程的实现：创建线程、保存信息、调度和切换线程、撤销线程、回收资源

用户级线程的实现：

运行时系统：用于管理和控制线程的函数的集合，这些函数驻留用户空间，并作为用户级线程与内核之间的接口

内核控制线程：连接到LWP，连接到LWP的线程才能与内核通信

三、线程的创建和终止

线程的创建：初始化线程、创建后返回线程标识符

线程的终止：终止线程用函数或系统调用终止操作，但有些线程被建立就会一直执行。大多数OS，线程被中止后并不立即释放所占资源，只有“其他线程”执行分离函数才

第三章 处理机调度与死锁

⑥处理机调度的层次和调度算法的目标

描述：作业可能要经历多级处理机调度

一、处理机调度层次

（一）高级调度（长程调度/作业调度）

对象是作业、决定将外存中处于后备队列的作业调入内存，创建进程和分配资源，并放入就绪队列、主要存在于多道批处理系统，分时和实时系统不设置高级调度

（二）低级调度（进程调度/短程调度）

对象是进程（/内核级线程）、决定就绪队列哪个进程获得处理机、多道批、分时和实时都要配置

（三）中级调度（内存调度）

对象是暂时不能运行的进程、把这些进程调到外存，设为挂起状态、一有条件，稍微有空就变为就绪状态

★分级按运行频率划分

二、处理机调度算法的目标

（一）共同目标

提高资源利用率、公平、平衡、策略强制执行

（二）批处理系统目标

处理机利用率高、平均周转时间短、系统吞吐量高

（三）分时系统目标

响应时间快、均衡性

（四）实时系统目标

截止时间保证、可预测性

②作业与作业调度

一、批处理系统中的作业

（一）作业和作业步

作业：包括程序、数据和作业说明书，在批处理系统，作业是基本单位从外存调入内存

作业步：独立步骤

（二）作业控制块（JCB）

包括作业标识、用户名称、用户账号、作业类型、作业状态、调度信息、资源需求、资源使用情况等

流程：进入系统→创建JCB→根据类型放到后备队列等待调度→入内存→根据JCB和作业说明书控制→完成→回收资源，撤销JCB

（三）作业运行的三个阶段和三种状态

收容阶段-后备状态、运行阶段-运行状态、完成阶段-完成状态

二、作业调度的主要任务

也叫：接纳调度

考虑：接纳多少作业、接纳哪些作业

三、先来先服务FCFS和短作业优先SJF调度算法

（一）先来先服务

就这样，完成或阻塞才分配到其他进程，实际中和其他算法结合使用

（二）短作业优先

实际用得更多、要预知作业运行时间、长作业、紧迫作业不利、无人机交互

四、优先级调度算法和高响应比优先调度算法

（一）优先级调度算法

外部赋予作业优先级

（二）高响应比优先调度算法

集SJF、FCFS的优点，兼顾长作业，但要做相应比计算，增加系统开销（Exp，做过类似）

③进程调度

一、进程调度的任务、机制和方式

（一）进程调度的任务

保存处理机的现场信息、按某种算法选取进程、把处理器分配给进程

（二）进程调度机制

排队器：插入就绪队列

分配器：从就绪队列取出，分配处理机

上下文切换器：保存、装入新的

（三）进程调度方式

非抢占方式：只有完成或因某事无法继续运行、I/O、执行了原语操作如block，才会引起进程调度

优点：简单、开销小、适用大多数批处理系统

抢占方式：对分时系统而言，有人机交互、对实时系统而言，能满足任务需求

主要原则：优先权原则、短进程优先原则、时间片原则

二、轮转调度算法

（一）轮转法（RR）的基本原理

按FCFS策略排成就绪队列，每隔一定时间就产生一次中断

（二）进程切换时机

时间片未用完就完成：马上调度队首进程，启动新时间片

时间片完还没完成：中断，进程被调到就绪队列队尾

（三）时间片大小的确定

太短有利于短进程、太长退化为FCFS算法，要计算平均周转时间（带权周转时间）

三、优先级调度算法

（一）优先级调度算法的类型

非抢占式和抢占式

（二）优先级的类型

确定优先级的依据：

静态优先级：进程类型、进程对资源的需求、用户需求

动态优先级：先赋予优先级，随着进程推进或等待时间增加而改变

四、多队列调度算法

将一条就绪队列拆分成多条，各有各调度算法

五、多级反馈队列调度算法

（一）调度机制

多条就绪队列、队列内使用FCFS算法，一个时间片未完成就放到下一个队列的末尾，最后一个队列用RR方式，按队列优先级调度，前队列空才到本队列运行

（二）调度算法的性能

终端型用户、短批处理作业用户、长批处理作业用户

六、基于公平原则调度算法

（一）保证调度算法

保证处理机公平分配

功能：跟踪进程已经执行的处理时间、该时间除以 n ，计算实际处理时间和应获得时间之比、比较比率、比率最低的获得处理机

（二）公平分享调度算法

考虑多用户

④实时调度

描述：实时系统有两种实时任务——HRT和SRT

一、实现实时调度的基本条件

（一）提供必要信息

就绪时间、开始截止和完成截止时间、处理时间、资源要求、优先级

（二）系统处理能力强

（处理时间 i /周期时间 i ）总和 ≤ 1 ，未考虑任务切换花费的时间还应该留有余地

提高系统处理能力的途径：单处理机系统增强处理能力，显著减少每个任务的处理时间、多处理机系统，就变成（处理时间 i /周期时间 i ）总和 $\leq N$

（三）采用抢占式调度机制

执行完关键程序和临界区后，能及时将自己阻塞起来，以释放处理机

（四）具有快速切换机制

对中断的快速响应能力、快速的任务分派能力

二、实时调度算法的分类

根据任务性质：**H/S**

根据调度方式：非抢占式/抢占式

（一）非抢占式调度算法

轮转、优先调度

（二）抢占式调度算法

基于时钟中断的抢占式优先级调度算法：等待时钟中断发生才剥夺当前任务的执行

立即抢占优先调度算法：好快好快，只要任务未处于临界区，就立即剥夺当前任务执行

三、最早截止时间优先EDF算法

非抢占式调度方式用于非周期实时任务：开始截止时间早的排前

抢占式调度方式用于周期实时任务：最早截止时间优先算法

四、最低松弛度优先LLF算法

松弛度？？？

五、优先级倒置

（一）优先级倒置的形成

（二）优先级倒置的解决方法

继承动态优先级的方法：**P3**继承**P1**的优先级，一方面避免**P2**抢占处理机、另一方面释放**mutex**资源

⑤死锁概述

一、资源问题

指互斥资源、不可被抢占的资源

（一）可重用性资源和消耗性资源

可重用性资源：一个只能分配给一个进程使用、顺序为请求、使用、释放、数目相对固定、运行期间不能创建或删除

可消耗性资源：在进程运行中树木不断变化，可为0、可以不断创建、放入缓冲区、可以由进程创建、使用并不再返回资源类

（二）可抢占性资源和不可抢占性资源

可抢占性资源：可以被其他进程抢占，如CPU和主存

不可抢占性资源：一旦分配给进程就不能强制收回，要做到底

二、计算机系统死锁

（一）竞争不可抢占性资源引起死锁

我要你的，你要我的，形成环路，死锁

（二）竞争可消耗资源引起死锁

譬如消息通信机制，先读后写，就会死锁

（三）进程推进顺序不当引起死锁

竞争不可抢占性资源引起死锁的翻版，多画了一个图

三、死锁的定义、必要条件和处理方法

（一）死锁的定义

这组死锁进程都在等其他进程释放所占资源

（二）产生死锁的必要条件

互斥条件：一段时间只被一个进程占用

请求和保持条件：进程已经保持至少一个条件，但有提出新资源请求

不可抢占条件：只能在进程用完才能释放

循环等待条件：存在循环链

（三）处理死锁的方法

预防死锁：设置限制条件，破坏产生死锁四个必要条件来预防

避免死锁：在资源动态分配时，用防止系统进入不安全状态

检测死锁：通过检测机构及时检测死锁，采取适当措施以解脱

解除死锁：撤销进程，回收资源，分配给处于阻塞的进程

防范逐渐减弱，但资源利用率提高，进程因资源因素而阻塞的频度下降

⑥预防死锁

一、破坏“请求和保持”条件

保证：进程请求资源时，不持有不可抢占资源

第一种协议：一次性全部申请，从而破坏“请求条件”，缺点是资源被严重浪费，进程饥饿

第二种协议：先释放资源，再请求新资源

二、破坏“不可抢占”条件

当保持了某些不可抢占资源后，提出新资源请求不得满足，就必须释放已经保持的资源，等需要时再申请

缺点：增加周转时间、增加系统开销、降低系统吞吐量

三、破坏“循环等待”条件

总有一个进程占据了较高序号的资源，此后它继续申请的资源必然是空闲的

⑦避免死锁

一、系统安全状态

（一）安全状态

计算一个资源分配的安全性，如果分配不会导致不安全状态，才可将资源分配给进程

（二）安全状态之例

（三）由安全状态向不安全状态的转换

二、利用银行家算法避免死锁

（一）银行家算法中的数据结构

Availalbe、Max、Allocation、Need

（二）银行家算法

睇wiki“银行家算法”条目

声明：Request是请求向量

Request ≤ Need，否则出错

Request ≤ Available，否则P等待

系统试探分配资源给P，并执行Availalbe -= Request; Allocation += Request; Need -= Request

⑧死锁的检测与解除

一、死锁的检测

（一）资源分配图

圆圈代表进程、方格代表一类资源、边代表资源分配

（二）死锁的定理

简化资源分配图，如果不能完全简化，就会死锁

（三）死锁检测中的数据结构

类似于银行家算法？？？

二、死锁的解除

常用两种方法：抢占资源、终止/撤销进程

（一）终止进程的方法

终止所有死锁进程：会功亏一篑

逐个终止进程：找到代价最小，考虑优先级、已执行时间、还需的时间、已用资源、交互式还是批处理式

（二）付出代价最小的死锁解除算法

很不实际

第四章 存储器管理

①存储器的层次结构

一、多层结构的存储器系统

CPU寄存器；

高速缓存Cache、主存储器RAM、磁盘缓存；

固定磁盘、可移动存储介质

二、可执行存储器

就是CPU寄存器和主存，访问很快

二、主存储器与寄存器

（一）主存储器

又叫主存or内存，相比CPU执行速度，它还是很慢，所以引入寄存器和高速缓存

（二）寄存器

完全与CPU协同工作，但好贵

三、高速缓存和磁盘缓存

（一）高速缓存

备份主存中较常用的数据，以减少CPU对主存储器的访问次数

（二）磁盘缓存

因为磁盘I/O速度远低于主存访问速度，所以设置磁盘缓存来暂存频繁使用的一部分磁盘数据和信息

②程序的装入和链接

用户程序要在OS中运行,要先装入内存,再转换为一个可执行程序:编译、链接、装入

一、程序的装入

(一)绝对装入方式

当OS很小,且仅能运行单道程序时,完全有可能知道程序驻留在内存的位置,那么就可以产生绝对地址的目标代码

(二)可重定位装入方式(静态重定位)

多道程序下,逻辑地址和物理地址不同,要加上起始地址,但只在进程装入时一次完成,故称为静态重定位,可装到内存任何允许位置

(三)动态运行时的装入方式

需要重定位寄存器

二、程序的链接

(一)静态链接方式

要解决:对相对地址进行修改(子程序的相对地址)、变换外部调用符号(生成可执行文件后不再拆开,又叫静态链接方式)

(二)装入时动态链接

便于修改和更新(各模块分开存放)、便于实现对目标模块的共享(静态每个应用模块必有目标模块的拷贝,无法共享,但动态可以一个目标链接多个应用模块)

(三)运行时动态链接

运行时发现没有,就由OS去找这个模块内添加程序装入过程和节约大量内存空间

③连续分配存储管理方式

一、单一连续分配

分系统区和用户区,单用户,单任务操作系统

二、固定分区分配

划分分区的方法:大小相等(一台计算机控制多台相同冶炼炉)和大小不等

内存分配:通常按大小排队,建立分区使用表,如果找不到大小足够的分区,就拒绝分配内存

三、动态分区分配

动态分区分配中的数据结构

动态分区分配算法:之后介绍4种分配算法和3中索引搜索算法

分区分配操作:分配内存、回收内存

四、基于顺序搜索的动态分区分配算法

首次适应算法、循环首次适应算法、最佳适应算法、最坏适应算法

五、基于索引搜索的动态分区分配算法

快速适应算法、伙伴系统、哈希算法

六、动态可重定位分区分配

紧凑、动态重定位、动态重定位分区分配算法

④对换

一、多道程序环境下的对换技术

(一)对换的引入

避免内存全部被阻塞,外存却有很多作业无法进入内存

(二)对换的类型

整体对换、页面对换

二、对换空间的管理

(一)对换空间管理的主要目标

文件区管理主要目标:提高文件存储空间的利用率,提高对文件的访问速度

对换空间管理的主要目标:提高换入换出的速度

(二)对换区空闲盘块管理中的数据结构

与内存的相似

(三)对换空间的分配与回收

与内存分配和回收雷同

三、进程的换出与换入

(一)进程的换出

选择被换出的进程

进程换出过程

(二)进程的换入

如果发现许多进程运行时缺页且内存紧张,才启动对换程序,将部分进程调至外存

如果缺页率明显减少,系统吞吐量已下降,则可以暂停运行对换程序

⑤分页存储管理方式

其实分三种:分页、分段、段页式

一、分页存储管理的基本方法

(一)页面和物理块

页面:页号

页面大小:太小,进程占用较多页面、太多,碎片多、适中,1kB~8kB

(二)地址结构

页号：P

页内地址（偏移量）：d

两个都有公式计算

（三）页表

从页号到物理块号的映射

二、地址变换机构

（一）基本的地址变换机构

如果发现页号 \geq 页表长度,就引发越界中断;否则页表始址+页号*页表项长度=物理块号

（二）具有快表的地址变换机构

有个快表在高速缓存

三、访问内存的有效时间

$EAT = t_1 + t_2 = 2t_1$

$EAT = a * \lambda + (t + \lambda) * (1 - a) + t = 2t + \lambda - t * a$

四、两级和多级页表

（一）两级页表

（二）多级页表

五、反置页表

（一）反置页表的引入

（二）地址变换

⑥分段存储管理方式

一、分段存储管理的基本方法

（一）方便编程

逻辑地址是由段名（段号）和段内偏移量（段内地址）决定的

（二）信息共享

段是信息的逻辑单位,简化共享过程

（三）信息保护

就是可以加个标识,不允许读写

（四）动态增长

可以解决这个问题

（五）动态链接

4.2.2

二、分段系统的基本原理

（一）分段

每个段有一个段号

（二）段表

实现从逻辑段到物理内存区的映射

（三）地址变换机构

段表项数目比页表项数目少,其所需的联想存储器相对较少,减少存取数据的时间

（四）分页和分段的主要区别

页是信息的物理单位,分页是系统管理上的需要

while 段是存储管理方式中的逻辑单位,分段目的在于更好满足用户的需要

页大小固定,段大小不固定

分页的用户程序地址是一维的,分段是二维的

三、信息共享

（一）分页系统对程序和数据的共享

每个进程都有页表,也都指向相同的物理块号

（二）分段系统中的程序和数据的共享

可重入代码是一种不允许任何进程对它进行修改的代码

配以局部数据区,把执行中可能改变的部分拷贝到该数据区,这样执行时只需对该数据区中的内容修改即可

四、段页式存储管理方式

分段 \rightarrow 分页,每段一个段名

段号,比较,加法算段表段号 \rightarrow 得到页号,(比较),加法算页表页号 \rightarrow 得到物理块号,加页内地址 \rightarrow 得物理地址

第五章 虚拟存储器

出现问题:作业很大、大量作业要求运行

⑩虚拟存储器概述

一、常规存储管理方式的特征和局部性原理

（一）常规存储管理方式的特征

传统:一次性、驻留性

（二）局部性原理

绝大部分顺序执行、调用进度不超过5、循环结构由少数指令构成,但多次执行、多对数据结构的处理,这些处理局限于很小的部分

时间、空间局限性

（三）虚拟存储器的基本工作情况

将少数页面或段先装入内存即可运行

二、虚拟存储器的定义和特征

（一）虚拟存储器的定义

有请求调入功能和置换功能，能逻辑上对内存内容加以扩充的一种存储器系统

（二）虚拟存储器的特征

多次性、对换性、虚拟性

三、虚拟存储器的实现方式

（一）分页请求系统

硬件支持：请求分页的页表机制、缺页中断机制、地址变换机制

实现请求分页的软件

（二）请求分段系统

硬件支持：请求分段的段表机制、缺段中断机构、地址变换机构

软件支持

②请求分页存储管理方式

一、请求分页中的硬件支持

（一）请求页表机制

（二）缺页中断机构

（三）地址变换机构

二、请求分页中的内存分配

（一）最小物理块数的确定

（二）内存分配策略

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

（三）物理块分配算法

算法：平均分配算法、按比例分配算法、考虑优先权的分配算法

三、页面调入策略

（一）何时调入页面

预调页策略：手动指出哪些页要调入内存、成功率偏低

请求调页策略：一次调入一页，须较大系统开销

（二）从何处调入页面

系统拥有足够的对换区空间：进程进行前就把进程相关的文件拷贝到对换区

系统缺少足够的对换区空间：未修改过的不到对换区，以后要用再从文件区调入

UNIX方式：从文件区入，出到对换区、允许页面共享

（三）页面调入过程

？？？

（四）缺页率

？？？

③页面置换算法

抖动：一个进程在运行中把大部分时间都花费在页面置换工作上

一、最佳置换算法和先进先出置换算法

（一）最佳置换算法

要知道未来需要哪页，实际上不可能

（二）先进先出页面置换算法

删走最老的页

二、最近最久未使用和最少使用置换算法

（一）最近最久未使用

看最近的n个，最老的踢走

（二）LRU置换算法的硬件支持

寄存器：8位寄存器，R7~R0，R值最小的页被踢出

栈：最新访问的是栈顶

（三）最少使用置换算法

现实使用这个多，一旦访问就在最高位置一

三、Clock置换算法

（一）简单的CLOCK置换算法

也叫最近未使用算法，就是有个访问位，1→0→换出

（二）改进型CLOCK置换算法

四类：A M = 0 0 ~ 1 1

第一步：先找 0 0

第二步：再找 0 1，并置所有页0 X

第三步：再找0 0，最后找0 1，一定找到

优点：减少I/O

缺点：增加系统开销

四、页面缓冲算法

（一）影响页面换进换出效率的若干因素

页面置换算法、写回磁盘的频率、读入内存的频率

（二）页面缓冲算法PBA

显著降低页面换进、换出频率，减少页面换进换出的开销

换入换出的开销大幅减少，才能使用简单的置换策略，如FIFO

要在内存中设置：空闲页面链表、修改页面链表

五、访问内存的有效时间

如果考虑快表的命中率和缺页率： $EAT = \dots\dots\dots$

如果仅考虑缺页率： $EAT =$

④“抖动”与工作集

一、多道程序度与“抖动”

（一）现象

先增后减

（二）原因

进程太多，物理块不够分

二、工作集

（一）工作集的基本概念

如果可以预知，就可以先调入内存，大大降低缺页率，从而显著提高处理机利用率

（二）工作集的定义

引用的集合，类似FIFO

三、“抖动”的预防方法

（一）采取局部置换策略

“抖动”影响较小

（二）把工作集算法融入到处理机调度中

每个进程在内存的驻留页面是否足够多，如果是就调入新作业、否则增加新物理块

（三）利用 $L=S$ 准则调节缺页率

缺页之间的平均时间= 平均缺页服务时间

（四）选择暂停的进程

先暂停优先级最低的进程、在选择并不重要，但较大的进程

⑤请求分段存储管理方式

其实也类似于分页，要硬件和软件支持

一、请求分段中的硬件支持

（一）请求段表机制

段表项：段名、段长、段基址、存取方式、访问字段A、修改位M、存在位P、增补位、外存始址

A、M：改进型CLOCK置换算法

P：本段是否调入内存

增补位：看是否做过动态增长

（二）缺段中断机制 （图5-12）

万一虚段S不在内存中，就阻塞请求进程。如果没有空闲区，就要拼接空闲区或者淘汰实段以形成空闲区

之后读入段S，修改段表及内存空区链

（三）地址变换机构

就是一个地址变换机构

二、分段的共享和保护

（一）共享段表

共享进程计数count、存取控制字段、段号

（二）共享段的分配与回收

共享段的分配、共享段的回收

（三）分段保护

越界检查、存取控制检查、环境保护机构

第六章 输入输出系统

①I/O系统的功能、模型和接口

一、I/O系统的基本功能

（一）隐藏物理设备的细节

（二）与设备的无关性

自动安装并寻找驱动程序，即插即用

（三）提高处理机和I/O设备的利用率

让处理机和I/O设备并行操作

（四）对I/O设备进行控制

这是驱动程序的功能

（五）确保对设备的正确共享

独占设备：打印机、磁带机

共享设备：磁盘

（六）错误处理

低级能够解决就不向高级报告，请求高级软件解决

二、I/O系统的层次结构和模型

（一）I/O软件的层次结构

用户层I/O软件

设备独立性软件：映射、保护、分块、缓冲、分配

设备驱动程序：设置设备寄存器、检查状态

中断处理程序

（二）I/O系统中各种模块之间的层次视图

I/O系统上下接口（图6-2）

I/O系统的分层：中断处理程序→设备驱动程序→设备独立性软件

三、I/O系统接口

（一）块设备接口

块设备、隐藏磁盘二维结构、抽象命令映射为低层操作

（二）流设备接口（Unix的）

字符设备：效率低、不可寻址

get和put操作：有缓冲区

in-control指令：互斥方式实现共享

（三）网络通信接口

②I/O设备和设备控制器

一、I/O设备

（一）I/O设备的类型

按使用特性：存储设备、I/O设备（输入输出交互的）

按传输速率：低速、中速、高速

（二）设备与控制器之间的接口

接口：数据信号线、控制信号线、状态信号线

二、设备控制器

（一）设备控制器的基本功能

接收和识别命令、数据交换、标识和报告设备的状态、数据缓冲区、差错控制

（二）设备控制器的组成

设备控制器与处理机的接口、设备控制器与设备的接口、I/O逻辑

三、内存映像I/O

（一）利用特定的I/O指令

缺点：访问内存和访问设备要两种不同的指令

（二）内存映像I/O

就是k为界限， $0 \leq k \leq n-1$ ，就是内存地址； $k \geq n$ ，就是寄存器地址。统一了对内存和对控制器的访问方法

四、I/O通道

（一）I/O通道设备的引入

这是一种特殊的处理机，但只局限于I/O相关的指令、而且没有自己的内存

（二）通道类型

字节多路通道：一个大水喉，多条小水管；一个换头快，一个速率慢

数组选择通道：利用率低

数组多路通道：甚至可以并行操作

（三）“瓶颈”问题

增加通路即可解决

③中断机构和中断处理程序

一、中断简介

（一）中断和陷入

中断：由外部设备引起，暂停当前程序，执行中断处理程序

陷入：CPU内部事件引起的，多是出错故障

（二）中断向量表和中断优先级

中断向量表：asm有学

中断优先级：现实中有多个中断信号源，要规定不同优先级

（三）对多中断源的处理方式

屏蔽中断：顺序执行。优点简单；缺点无视实时中断请求

嵌套中断：有个优先级

二、中断处理程序

测定是否有未响应中断信号；

保护被中断进程的CPU环境；

转入相应设备处理程序；

中断处理；

恢复CPU现场并退出中断

④设备驱动程序

一、设备驱动程序概述

（一）设备驱动程序的功能

接收命令和参数，并转换为低层操作序列

检查I/O合法性，了解I/O工作状态，传递I/O设备操作有关参数，设置设备工作方式

及时响应设备控制器发来的中断请求，并根据中断类型，调用响应中断处理程序

（二）设备驱动程序的特点

抽象的I/O请求转换成具体的I/O操作，反映给I/O进程

和硬件特性紧密相关，终端驱动程序可以只有一个

常用控制方式：中断驱动、DMA

一部分必须用汇编语言写，很多驱动程序基本部分已经固化在ROM

允许可重入

（三）设备处理方式

一类设备一个进程

一个I/O进程负责各类设备的I/O操作

只为各类设备设置相应的设备驱动程序，供用户或系统进程调用（目前用得最多）

二、设备驱动程序的处理过程

（一）将抽象要求转换为具体要求

（二）对服务请求进行校验

譬如要求打印机输入数据

（三）检查设备的状态

检测寄存器中的不同位，了解设备的状态

（四）传送必要参数

波特率、奇偶校验等等参数

（五）启动I/O设备

了解数据是否到达

三、对I/O设备的控制方式

（一）使用轮询的可编程I/O方式

无限等待，好浪费CPU

（二）使用中断的可编程I/O方式

百倍提高CPU利用率

（三）直接存储器访问方式

至少传送一个数据块，DMA方式提高CPU和I/O之间的并行程度

（四）I/O通道控制方式

使用通道程序完成CPU指定的I/O任务

⑤与设备无关的I/O软件

一、与设备无关软件的基本概念

（一）以物理设备名使用设备

以前应用程序与物理设备直接相关

（二）引入了逻辑设备名

通过更换逻辑设备表即可改变显示终端

（三）逻辑设备名称到物理设备名称的转换

要搞一张逻辑设备表

二、与设备无关的软件

（一）设备驱动程序的统一接口

要有统一接口，同时抽象设备名要映射到适当的驱动程序上

（二）缓冲管理

设置缓冲区，缓和CPU和I/O设备之间的速度矛盾、提高CPU利用率

（三）差错控制

暂时性错误：只有连续多次出错才报告上层，否则由设备驱动程序自己处理

持久性错误：要查清发生错误的原因，避免以后再次发生错误

（四）对独立设备的分配与回收

独占设备要先申请

（五）独立于设备的逻辑数据块

注：与设备无关软件功能：设备驱动程序的统一接口、缓冲、错误报告、分配与释放专用设备、提高与设备无关的块大小

三、设备分配

（一）设备分配中的数据结构

系统设备表SDT→设备控制表DCT：类型、标识符、状态、设备队列队首指针、重复执行次数、指向控制器表的指针→控制其控制表COCT→通道控制表CHCT

（二）设备分配时应考虑的因素

设备固有属性：独占、共享、虚拟设备

设备分配算法：FCFS、优先级高优先

安全性：安全、不安全

（三）独占设备的分配程序

独占设备：分配设备、控制器、通道

如果要设备无关地找设备，就要从SDT找DCT，再逐个测试安全性

四、逻辑设备名到物理设备名映射的实现

（一）逻辑设备表LUT

逻辑设备名、物理设备名、设备驱动程序的入口地址

（二）逻辑设备表设置问题

整个系统一张LUT

或每个用户一张LUT

⑥用户层的I/O软件

一、系统调用与库函数

（一）系统调用

使用系统调用I/O设备，用户态→内核态→用户态

（二）库函数

库函数与调用程序接在一起

二、假脱机系统

（一）假脱机技术

利用专门的外围控制机，先将低速I/O设备上的设局传送到高速磁盘上，或相反

（二）SPooling的组成

输入井和输出井、输入缓冲区和输出缓冲区、输入进程和输出进程、井管理程序

（三）SPooling系统的特点

提高I/O速度、将独占设备改造为共享设备、实现虚拟设备功能

（四）假脱机打印机系统

磁盘缓冲区、打印缓冲区、假脱机管理进程和假脱机打印进程

（五）守护进程

有个假脱机目录，由守望进程按目录文件依次完成各个进程设备的请求，就可以将一个独占设备改为多个进程共享设备

⑦缓冲区管理

一、缓冲的引入

（一）缓和CPU与I/O设备间速度不匹配的矛盾

速度有差距，都可以设置缓冲区

（二）减少对CPU的中断频率

（三）解决数据粒度不匹配的问题

生产者生产的数量和消费者消费的数量差距

（四）提高CPU和I/O设备之间的并行性

CPU和打印机可以并行工作呢

二、单缓冲区和双缓冲区

（一）单缓冲区

缓冲区也会阻塞

（二）双缓冲区

CPU执行第一行中的命令时，用户可以继续向第二缓冲区输入下一行数据

如果两台电脑只设置但缓冲，就要再设置一个接收缓冲区，一个发送缓冲区

三、环形缓冲区

（一）环形缓冲区的组成

空区R，装满的区G，正在使用的现行工作缓冲区C；另外还有多个指针

（二）环形缓冲区的使用

Getbuf和Releasebuf

（三）进程之间的同步问题

Nexti赶上Nextg：输入速度>处理速度

Nextg赶上Nexti：处理速度>输入速度

四、缓冲池

（一）缓冲池的组成

专为生产者-消费者设置的，包含一个管理数据结构和一组操作函数，管理多个缓冲区

包括空白缓冲队列、输入队列、输出队列

（二）Getbuf过程和Putbuf过程

设置MS(type)互斥访问缓冲池队列和RS(type)进程同步使用缓冲区

（三）缓冲区的工作方式

收容操作、提取输入、收容输出、提取输出

⑧磁盘存储器的性能和调度

一、磁盘性能简述

（一）数据的组织和格式

数据组织和格式：磁盘-双面可存储盘片（存储面）-扇区-磁道（柱面）

（二）磁盘的类型

固定头磁盘、移动头磁盘

（三）磁盘访问时间

二、早期的磁盘调度算法

（一）先来先服务

就是先来的先找，很公平很简单，但平均寻道好长

（二）SSTF最短寻道时间优先

选择一个与磁头距离最近的磁道

三、基于扫描的磁盘调度算法

（一）扫描算法SCAN

来回

（二）循环扫描算法CSCAN

单程

（三）NStepSCAN和FSCAN调度算法

N步扫描算法：将磁盘请求队列分成若干长度为N的子队列，再用FCFS依次处理这些子队列

FSCAN算法：只分两个队列，一个现在要扫描的，一个是扫描时新冒出来的

第七章 文件管理

①文件和文件系统

一、数据项、记录 and 文件

（一）数据项

基本数据项：描述以对象某种属性的字符集，如学号、姓名、年龄这些不能再细分的

组合数据项：若干个基本数据项组成的，就是还可以细分的

（二）记录

描述一个对象在某方面的属性，注意要有关键字key，方便查找

（三）文件

多条记录组成文件

文件属性：类型、长度、物理位置、建立时间（即最后一次修改时间）

二、文件名和类型

（一）文件名和拓展名

没什么好说的，都懂

（二）文件类型

按用途分：系统文件、用户文件、库文件

按文件数据形式：源文件、目标文件、可执行文件

按存取控制属性：只执行文件、只读文件、读写文件

按组织形式和处理方式分类：普通文件、目录文件、特殊文件

三、文件系统的层次结构

（一）对象及其属性

管理对象：文件、目录、磁盘存储空间

（二）对对象操纵和管理的软件集合

I/O控制层、基本文件系统层、基本I/O管理程序、逻辑文件系统

（三）文件系统的接口

命令接口、程序接口

四、文件操作

（一）最基本的文件操作

创建、删除、读、写、设置读写位置

（二）文件的“打开”和“关闭”操作

（三）其他文件操作

设置和获得文件的属性、查询文件状态

有关目录的，就是创建、删除、改变当前目录等

②文件的逻辑结构

一、文件逻辑结构的类型

有结构文件：记录式文件

无结构文件：流式文件

（一）按文件是否有结构分类

有结构文件（如数据库）：定长记录、变长记录

无结构文件（txt）：源程序、可执行文件

（二）按文件的组织方式分类

顺序文件：可以定长可以变长，一直按顺序下去（如犯人的记录）

索引文件：加张索引表

索引顺序文件：分组，组内是顺序，组头有索引

二、顺序文件

（一）顺序文件的排列方式

串结构：要从头开始找

顺序结构：有个关键字

（二）顺序文件的优缺点

最高效、但交互应用中效率好差、而且增加删除一个记录困难

所以要配置一个运行记录文件，按时合并

三、记录寻址

（一）隐式寻址方式

一个一个读，读了（ $n-1$ ）个才找到 n

（二）显式寻址方式

定长就方便，直接乘索引号即可

变长就要加上 Li ，表示一段记录的长度

或者利用关键字查找

四、索引文件

（一）按关键字建立索引

索引文件三要素：索引号、长度、指针

多个索引表的索引文件：从不同属性查找同一对象

五、索引顺序文件

（一）索引顺序文件的特征

引入文件索引表，可以实现对索引顺序文件的随机访问；

增加溢出文件，可以记录新增加、删除和修改的记录

（二）一级索引顺序文件

分组，组首进入索引顺序文件

（三）两级索引顺序文件

索引顺序表做组

六、直接文件和哈希文件

（一）直接文件

关键字本身就决定记录的物理地址，所以可以直接查找，有键值转换

（二）哈希文件

$A = H(K)$ ，通常是指向某一目录表相应表目的指针

⑨文件目录

要求：

实现“按名存取”

提高对目录的检索速度

文件共享

允许文件重名

一、文件控制块和索引结点

（一）文件控制块FCB

包含三类信息：基本信息、存取控制信息、使用信息

基本信息：文件名、文件物理位置、文件逻辑结构、文件物理结构

存取控制信息：各类人的存取权限

使用信息类：建立时间、最近修改时间、当前使用信息

（二）索引结点

引入：怕文件目录太大，只用文件名，轻量级文件目录

磁盘索引结点：文件主标识符、文件类型、文件存取权限、文件物理地址、文件长度、文件连接计数、文件存取时间

内存索引结点：索引结点编号、状态、访问计数、文件所属文件系统的逻辑设备号、链接指针

二、简单的文件目录

（一）单级文件目录

整个文件系统只有，一张目录表，目录项有：文件名、文件扩展名、文件长度、文件类型、文件物理地址和其他属性

每次创建都要搜索有没有相同的文件名

优点是简单，但只实现了“按名存取”，其他三个要求没有实现

（二）两级文件目录

MFD→UFD

会有隔离，这个结构可以有效将多个用户隔开，在各个用户完全无关时，这是一个优点。

但如果要合作完成一个大任务时，这种隔离就会使诸多用户之间不便于共享

三、树形结构目录

（一）树形目录

一个目录文件中的目录项，可以既作为目录文件的FCB，又是数据文件的FCB

（二）路径名和当前目录

路径名：唯一通路，用/连接

当前目录，相对路径名，绝对路径名

（三）目录操作

创建、删除、不删除非空目录、可删除非空目录

改变、移动、链接目录、查找目录

（四）目录查询技术

线性检索法：在单级目录中，用用户提供的文件名，顺序查找；在树形目录中，就按路径名查找

Hash方法：建立一张Hash索引文件目录，利用Hash方法查询——利用用户提供的文件名，转换为文件目录索引值，再用索引值在目录中查找

注：如果使用了通配符，就无法用Hash方法检索了

“冲突”：1.看目录项是否空 2.看文件名是否匹配 3. 如果不匹配，就要在Hash值加上一个常数，形成新的索引值

⑩文件共享

一、基于有向无循环图实现文件共享

（一）有向无循环图DAG

由附加操作**Append**来完成，而新增加的盘块只会在执行了操作的目录中，新增加的部分不能被共享

（二）利用索引结点

用索引结点，任何用户对共享文件所进行的**Append**操作或修改，都将引起相应结点内容的改变

还增加一个**count**链接计数

其他用户在使用，拥有者删了，文件依然存在

二、利用符号链接实现文件共享

（一）利用符号链接的基本思想

即允许一个文件或子目录有多个父目录，但只有一个是“主”父目录

（二）如何利用符号链实现共享

由系统创建一个**LINK**类型的新文件，取名为**F**，并将**F**写入链接父目录**D5**中，就可以实现**D5**与**F8**的链接。新文件只有被链接文件**F8**的路径名——所以叫做“符号链接”——新

（三）利用符号链实现共享的优点

用户删了链接文件，也不会删掉本来的文件；文件主删了文件，其他用户访问不了，自然会删掉符号链

（四）利用符号链的共享方式存在的问题

读盘需时、符号链太多，琐碎

第八章 磁盘存储器的管理

①外存的组织方式

连续组织方式、链接组织方式、索引组织方式

一、连续组织方式

位于同一磁道，读写不用移动磁头

优点：顺序访问容易、顺序访问速度快

缺点：要求为一个文件分配连续的存储空间、要事先知道文件长度、不够灵活删除和插入、对于动态增长的文件，很难分配空间

二、链接组织方式

优点：消除外部碎片，提高外存利用率；对插入、删除修改记录都非常容易；能够适应动态增长

隐式链接：一个跟一个，如同链表；碎片多；万一一个错，整个文件用不了

显式链接：将各物理块的指针显式存在内存的一个表内，每个**FCB**对应一个字段，因为在内存查找，所以大大提高检索速度，还减少访问磁盘的次数，而这叫做**FAT**

五、索引组织方式

（一）单级索引组织方式

不支持高效直接存取，要对于一个较大的文件进行存取，就要顺序地查找许多盘块号

FAT需要占用较大内存空间

所以创造“表中表”——索引分配图

（二）多级索引组织方式

就是多级，如同“全语言字典”

优点：大大加快对大型文件的查找速度

（三）增量式索引组织方式

增量式索引组织方式的基本思想

小的用直接寻址、中的用单级索引组织方式、大的用两三级索引组织方式

Unix System V的组织方式：索引结点有**13**个地址项，前十个是直接地址，最大放**40KB**；第十一个是一次间接地址，**1K**个盘块号，允许文件长达**4MB**；如果还超过**4ME**

②文件存储空间的管理

一、空闲表法和空闲链表法

（一）空闲表法

即在外存所有空闲区建立一张空闲表，每个空闲区对于一个表项，记录序号、第一空闲盘块号、空闲盘块数

分配和回收与内存相似，但在外存为加快分配速度，连续分配依然有用。小的连续分配，大的离散分配，多媒体文件依然连续分配

（二）空闲链表法

空闲盘块链：一直分配，按链表分配。如果删除文件就挂在链尾

优点：简单，缺点：但为一个文件分配时可能重复操作多次，分配和回收效率较低，而且盘块链会很长

空闲盘区链：优点：分配回收效率较高，盘区链短；缺点复杂

二、位示图法

（一）位示图

利用二进制的一位来表示磁盘中一个盘块的使用情况，也可以是二维数组

（二）盘块的分配

扫描位示图，找到一个或一组0，之后计算盘块号： $b = n(i - 1) + j$ ，修改位示图=1

（三）盘块的回收

从盘块号转换为行列号：

$i = (b - 1) \text{ DIV } n + 1$

$j = (b - 1) \text{ MOD } n + 1$

三、成组链接法

（一）空闲盘块的组织

每组含有盘块总数**N**和该组所有盘块号记入前一组的第一个盘块的**S.free(0)~S.free(99)**，这样各组第一个盘块可链接成一条链

（二）空闲盘块的分配与回收

当栈中空闲盘块号已达**100**时，表示栈已满，便将先有栈中**100**个盘块号记入新回收的盘块中，将盘块号作为新栈底

《计算机操作系统》总结一

一、操作系统概述1.1 操作系统的基本概念 在信息化时代，软件被称为计算机系统的灵魂。而作为软件核心的操作系统，已经与现代计算机...

来自

想对作者说点什么

LAST_JEDI：写的不错，不是markdown难受

(2个月前 #4楼)

风吟。：挺好的啊

(2个月前 #3楼)

天烽：博主，这是西安电子科技大学出版社的第四版么？

(1年前 #2楼)

chevis236：总结的挺妥帖的（/滑稽）

(1年前 #1楼)

计算机操作系统基础

第三章 操作系统基础3.1 操作系统的定义 是计算机硬件的第一级扩充，是机器的管理者，控制和管理计算机硬件和软件资源、合理的组织...

来自：weixin_4

计算机操作系统（笔记）

计算机操作系统

来自：Xior

计算机操作系统第四版第一章知识点归纳总结

第一章、操作系统引论一、操作系统基础1、操作系统的定义：操作系统（Operating System, OS）是指控制和管理整个计算机系统的硬件和...

来自：

不停抽插，就是不想射有多爽！老中医说：多吃他就可以做到！

通聚商贸·熾燚

计算机操作系统知识梳理

1、进程和线程以及它们的区别（1）进程是对运行时程序的封装,是系统进行资源调度和分配的基本单位,实现操作系统的开发。（2）线程是...

来自：wzk64

计算机操作系统

1 有效性，方便性，可扩充性，开放性2 作用：用户和计算机硬件的接口（命令方式，系统调用方式，图形、窗口方式）；计算机系统资源...

来自：ny1763

计算机操作系统原理--Linux实例分析

本书是结合Linux介绍操作系统的原理的书，是一本理论的书。 本书从计算机硬件的分类来叙述如何高效利用计算机资源，及我们使用的操...

来自：l

计算机操作系统是什么？

1）背景：早期要想让计算机执行程序，那么在编写程序的时候就要参考一大堆的硬件功能函数，并且要学习机器语言才能编写的成，所以如...

计算机操作系统种类详解

计算机操作系统种类详解作者:GsuhyFihx(邮箱：2534294648@qq.com) 请勿转载，支持原创一、类别1.实时操作系统：实时操作系统（Rea...

来自：G

不停抽插，就是不想射有多爽！老中医说：多吃他就可以做到！

通聚商贸·熾燚

下载

计算机操作系统学习指导与题解 第三版 汤子瀛

12-25

计算机操作系统学习指导与题解 第三版 汤子瀛 对于要考研的同学还有基础比较薄弱的同学有帮助，上面很多例题都很经典，详解也比较全面，建议参考

文章热词

操作系统操作系统视频教程操作系统培训操作系统学习操作系统学习

相关热词

c# 知识点c++的知识点c++程序设计第四版pdfc++操作系统内存c++语法知识点简明python教程第四版python学习手册第四版

计算机操作系统概述

硬件上配置操作系统的目标： 有效性：体现在提高系统资源利用率和提高系统的吞吐量；方便性；可扩充性；开放性；操作系统作用（三...

来自：weixin_3

jamesehng

关注

636篇文章

波书文学

关注

120篇文章

DiuDiu_Yang

关注

57篇文章

firstbloodz

关注

12篇文章

https://blog.csdn.net/sinat_31710221/article/details/53056058

20/25

| | | |
|---|------------------------------|------------------|
| <div>操作系统的知识点整理</div> <div>操作系统 操作系统是管理计算机硬件资源，控制其他程序运行并为用户提供交互操作界面的系统软件的集合。操作系统是计算机系统的关键... 来自：即将进入职场的我</div> | | |
| <div>计算机操作系统(第四版)之输入输出系统要点梳理</div> <div>输入输出系统 输入输出系统</div> <div>来自</div> | | |
| <div>操作系统读书笔记（汤子瀛 思维导图）</div> <div>自己整理的操作系统知识点，Xmind编辑的。</div> <div>来</div> | | |
| <div>高新路竟有一位51岁股市奇才，炒股2年靠1个铁律，存款惊哭老婆</div> <div>潮望 · 爍爍</div> | | |
| <div>操作系统最全知识点总结（找工作，考研必备）</div> <div>操作系统知识点总结，供找工作、考研的小伙伴们拿来作为操作系统复习提纲...</div> <div>来自</div> | | |
| <div>C++：strcmp与strncmp的比较</div> <div>strcmp与strncmp都是用来比较字符串的，区别在于能否比较指定长度字符串。 strcmp C/C++函数，比较两个字符串 设这两个字符串为str1...</div> <div>来自</div> | | |
| <div>strcmp</div> <div>C/C++函数，比较两个字符串 设这两个字符串为str1，str2，若str1=str2，则返回零；若str1若str1>str2，则返回正数。 中文名strcmp外文...</div> | | |
| <div>线程与进程的区别</div> <div>1. 一个程序至少有一个进程，一个进程至少有一个线程 2. 线程的划分尺度小于进程，使得多线程程序的并发性高 3. 进程在执行过程中拥有独...</div> <div>来自：</div> | | |
| <div>操作系统知识点整理(持续更新)</div> <div>基础概念 操作系统的四个特性 1. 并发: 同一时间间隔内支持执行多个任务, 对于一个核则是交替执行； 2. 共享: 硬件资源或数据资源支持多个...</div> <div>来自：moxi</div> | | |
| <div>高新路竟有一位51岁股市奇才，炒股2年靠1个铁律，存款惊哭老婆</div> <div>潮望 · 爍爍</div> | | |
| <div>进程和线程关系及区别</div> <div>1.定义 进程是具有一定独立功能的程序关于某个数据集合上的一次运行活动,进程是系统进行资源分配和调度的一个独立单位. 线程是进程的...</div> <div>来自：yaosin</div> | | |
| <div>计算机操作系统 第三版</div> <div>计算机操作系统 第三版 汤子瀛 目录书签都有，可以学习！</div> | | |
| <div>计算机操作系统——知识点总结</div> <div>操作系统主要知识点总结 包括5个主要部分：概述，进程，处理机调度与死锁，存储器管理，虚拟存储器 另外的部分如文件管理，磁盘管理...</div> <div>来自：BigWater</div> | | |
| <div>操作系统知识点整理</div> <div>1. 进程的有哪几种状态，状态转换图，及导致转换的事件进程的状态：运行、阻塞、就绪三种状态。运行和就绪之间的转换由调度程序引导...</div> <div>来自：Ail</div> | | |
| <div>计算机四级操作系统题目总结（4-8）</div> <div>1.异步环境下的一组并发的进程因为直接制约而相互发送消息，进行相互合作，相互等待，使得各进程按照一定的速度执行的过程称为进程...</div> <div>来自：</div> | | |
| <div>原来血糖高真正的元凶是它！你知道吗？</div> <div>鑫丰杰 · 顶新</div> | | |
| <div>下载</div> | <div>计算机操作系统第四版的PPT</div> | <div>06-18</div> |
| <div>献给那些正在学操作系统的！</div> | | |
| <div>计算机操作系统（第四版）第二章 进程的描述与控制（后四节+清晰的总结+图示）</div> <div>1 引言 好久没有整理操作系统的内容，这篇文章整理的是后四节的内容，分别是经典进程的同步问题、进程通信、线程的基本概念、线程...</div> <div>来自：</div> | | |
| <div>下载</div> | <div>计算机操作系统(第4版)汤小丹课件</div> | <div>03-29</div> |
| <div>计算机操作系统(第4版)汤小丹编写，全部课件</div> | | |
| <div>下载</div> | <div>计算机操作系统知识结构思维导图</div> | <div>07-11</div> |
| <div>用mindmanage软件做的关于大学计算机操作系统知识结构的思维导图，包括《计算机操作系统（第四版）》的全部知识点，整理不易，谢谢。ps：因为是mmap格式，所以需要安...</div> | | |

| | | |
|--|--|----------------------------|
| <div><div>下载</div><div>计算机操作系统-第4版课后习题答案-张尧学</div></div> | | 06-19 |
| 第一章 | 1. 什么是操作系统的基本功能? 答：操作系统的职能是管理和控制计算机系统中的所有硬、软件资源，合理地组织计算 | 机工作流程，并为用户提供一个良好的... |
| <div><div>高新路52岁股民9年悟出股市投资铁律，1指标选股战法，震惊投资界</div><div>意欧迅 · 熯熯</div></div> | | |
| <div><div>下载</div><div>计算机操作系统学习指导与题解.pdf</div></div> | | 09-09 |
| 本书为《计算机操作系统》一书的配套辅导书。书中全面介绍了计算机操作系统的基本原理。全书共分为10章，每一章首先扼要阐述该章的基本内容，然后给出重点和难点的学习提... | | |
| <div><div>下载</div><div>计算机操作系统第三版.pdf</div></div> | | 02-07 |
| 计算机操作系统第三版，内容讲解的很是详细，非常好的一本书。 | | |
| <div><div>2019考研专业课《计算机操作系统原理》精讲视频课程</div><div>-</div></div> | | |
| <div><div>下载</div><div>计算机操作系统-汤子瀛-第3版 PDF完整版</div></div> | | 01-24 |
| 目 录 第一章 操作系统引论 1.1 操作系统的目标和作用.....1 1.1.1 操作系统的目标..... | | |
| <div><div>《计算机操作系统》总结五（死锁）</div><div>转载自:http://blog.csdn.net/bigpudding24/article/details/48608579 2.17 死锁的概念以及产生死锁的原因死锁的定义 在多道程序系统中，由...</div></div> | | 来自： yongc |
| <div><div></div><div>中科院在职研究生招生简章</div><div>百度广告</div></div> | | |
| <div><div>操作系统复习要点</div><div>什么是虚拟内存，什么是物理内存 进程和线程 进程是具有一定功能的程序关于某个数据集合上的一次运行活动，进程是系统进行资源调度和...</div></div> | | 来自： c |
| <div><div>操作系统总结（一）</div><div>1.操作系统主要特性：并发性 共享性 异步性 虚拟性 2.操作系统的功能：处理器管理 存储管理 设备管理 文件管理 作业管理 ...</div></div> | | 来自： xir |
| <div><div>进程和线程的本质和区别</div><div>进程是什么？ 程序并不能单独运行，只有将程序装载到内存中，系统为它分配资源才能运行，而这种执行的程序就称之为进程。程序和进程...</div></div> | | 来自： ITL |
| <div><div>深入理解计算机操作系统(一)</div><div>目录 1.1 信息就是位加上上下文 1.2 程序被其他程序翻译成不同的格式 1.3 了解编译系统如何工作是大有益处的 1.4 处理器读并解释储存在内存...</div></div> | | 来自： 读万 |
| <div><div>strcmp 函数实现</div><div>strcmp int strcmp(constchar*str1,constchar*str2); str1和str2可以是字符串常量或者字符串变量，返回值为整形。返回结果如下规定： ①...</div></div> | | 来自： f |
| <div><div>荣泉-纸箱加工专用机械</div><div>荣泉纸箱机械 · 顶新</div></div> | | |
| <div><div>进程和线程的区别</div><div>操作系统面试题 1、进程和线程的区别？ 解析：（1）进程是资源的分配和调度的一个独立单元，而线程是CPU调度的基本单元 （2）...</div></div> | | 来自： zhou75 |
| <div><div>进程和线程的主要区别（总结）</div><div>根本区别：进程是操作系统资源分配的基本单位，而线程是任务调度和执行的基本单位在开销方面：每个进程都有独立的代码和数据空间（ ...</div></div> | | 来自： kuang |
| <div><div>线程与进程的区别与联系</div><div>线程，有时被称为轻量级进程(Lightweight Process，LWP)，是程序执行流的最小单元。一个标准的线程由线程ID，当前指令指针(PC)，...</div></div> | | 来自： sh |
| <div><div>进程和线程的定义、区别与联系</div><div>一、进程 进程：指在系统中能独立运行并作为资源分配的基本单位，它是由一组机器指令、数据和堆栈等组成的，是一个能独立运行的活动...</div></div> | | 来I |
| <div><div>操作系统（第四版）期末复习总结（中）</div><div>衔接我的上一篇博文，这片从第三章开始第三章：处理机调度与死锁</div></div> | | 来自： qq_4 |

| | |
|--|-------|
| <div><div>下载</div><div>计算机操作系统第四版（最新版）汤小丹第1~6章 ppt</div></div> <div>计算机操作系统第四版（最新版）汤小丹第1~6章.zip</div> | 08-13 |
| <div><div>计算机操作系统第四版答案===转载</div><div>http://blog.csdn.net/zhang_hongli_li/article/details/48623285</div></div> <div>来自： fll</div> | |
| <div><div>下载</div><div>计算机操作系统课后习题答案(第四版)</div></div> <div>计算机操作系统课后习题答案(第四版) (2).docx</div> | 05-10 |
| <div><div>下载</div><div>计算机操作系统教程（第四版）</div></div> <div>计算机系统原理教程第四版，大学教材，有利于理解计算机系统原理，方便大家学习，有书签，方便学习。</div> | 09-29 |
| <div><div>下载</div><div>计算机操作系统(第4版)汤小丹完整PPT</div></div> <div>计算机操作系统(第4版)汤小丹完整PPT</div> | 10-08 |
| | |
| <div><div>现代操作系统教程</div><div>操作系统</div><div></div></div> | |
| <div><div>CSDN 2018博客之星活动报名开始了！</div><div>如期而至，CSDN 2018 博客之星评选活动又跟大家见面了。正如我们一直坚持的主题“星星之火可以燎原”一样，点滴记录，精彩人生。一篇...</div></div> <div>来自： </div> | |
| <div><div>SQL提升（二）</div><div>Sql表操作提升 一、前言 Sql是最重要的关系数据库操作语言，现在基本上任何与数据库相关的操作都离不开sql。所以说sql功能是很强...</div></div> <div>来自： 愤怒</div> | |
| <div><div>Swagger教程二</div><div>Swagger搭建Restful接口教程二 一、前言 上一章节我们说的是swagger-ui也就是swagger1,接下来我们说的是swagger升级版swagger2 ...</div></div> <div>来自： 愤怒</div> | |
| <div><div>PyQt5+VSCode简单配置附pyuic编译的附加选项说明...</div><div>PyQt5+VSCode简单配置附pyuic编译的附加选项说明Python IDE 之 VSCode首先第一步第二步第三步第四步如果你比较懒- -！，不想每次都...</div></div> <div>来自： weixin_4</div> | |
| <div><div>MyBatis——mapper.xml提升指南</div><div>MyBatis常见细节问题 一、前言 MyBatis 是一款优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。MyBatis 避免了几乎...</div></div> <div>来自： 愤怒</div> | |
| <div><div>【Linux从入门到放弃】Linux开发第二天&用户的添加&安装软件&更改ssh登录</div><div>Linux第二天 1.用户的添加 useradd iceFrog passwd iceFrog 更改密码：echo "dota" passwd -stdin iceFrog 切换用户：su ...</div></div> <div>来自： 唐</div> | |
| <div><div>webstorm 2018 激活破解方法大全</div><div>webstorm 作为最近最火的前端开发工具,也确实对得起那个价格,但是秉着勤俭节约的传统美德,我们肯定是能省则省啊。方法一：（更新时间...</div></div> <div>来自： 唐</div> | |
| <div><div>pyCharm最新2018激活码</div><div>本教程对jetbrains全系列可用例：IDEA、WebStorm、phpstorm、clion等 因公司的需求，需要做一个爬取最近上映的电影、列车号、航班号...</div></div> | |
| <div><div>很黄很暴力的十个网站</div><div>13岁的北京学生张某，在去年12月27日19时新闻联播一则关于净化网络视听的新闻里，接受采访时说的话激起了轩然大波：“上次我上网查...</div></div> <div>来自： Kinb_l</div> | |
| <div><div>最新迅雷“应版权方要求，文件无法下载”的解决办法</div><div>迅雷下载有的电影电视剧的时候会出现：应版权方要求，文件无法下载，或者显示迅雷任务包含违规内容 无法继续下载。这个是因为版权方...</div></div> <div>来自： </div> | |
| <div><div>整理了10个干净、好用的BT、磁力链搜索网站给大家</div><div>现在越来越流行在线看视频了，但是对于我得收藏癖爱好者，还是希望可以有比较好的资源网站的，尤其是种子、磁力链网站。所以就整理...</div></div> <div>来自： YX</div> | |

| | |
|---|----------------------------|
| <div>史上最简单的 SpringCloud 教程 终章</div> <div>转载请标明出处： http://blog.csdn.net/forezp/article/details/70148833 本文出自方志朋的博客 错过了这一篇，你可能再也学不会 Sp...</div> | 来自 |
| <div>批处理添加字段和删除字段</div> <div>在平时的工作中你或许会遇到这样的问题，给某个文件夹下面的所有的图片添加同一个字符串，如果一个一个去F2再CTRL+V这样就会烦得很...</div> | 来自： pyf_91 |
| <div>ElementUI离线使用-分页功能</div> <div>Element离线使用 -分页功能 引入element后，由于网络速度较慢，发现页面运行起来，比较卡顿，于是将其下载到本地 1.下载静态资源 elem...</div> | |
| <div>vue-cli3脚手架/单文件/环境搭建</div> <div>vue-cli3脚手架/单文件/环境搭建。vue3.x版本相对于2.x版本做了许多地方的优化,个人总结主要还是使用上使开发者更加一目了然,配置起来...</div> | 来自： 欢迎来到 |
| <div>vue3修改link标签默认icon</div> <div>vue3修改link中标签默认icon,vue3初次使用的时候不好好阅读配置难免会遇到一些坑,本人在项目完结的时候打算把浏览器的导航小icon图标...</div> | 来自： 欢迎来到 |
| <div>jsp四大作用域和九大对象</div> <div>常用于服务器间同一请求不同页面之间的参数传递，常应用于表单的控件值传递。...</div> | 来自 |
| <div>架构模式之MVVM</div> <div>架构模式之MVVM 前言 MVVM是一种架构模式，本文会涉及一小点vue代码，以及一篇简单的springboot的代码，建议在阅读本文档前先对...</div> | |
| <div>SpringBoot整合ActiveMQ消息队列</div> <div>首先要讲什么是ActiveMQ：AciveMQ是Apache出品的目前最流行，能力强劲的开源消息总线 主要功能： 1、 解决服务之间代码耦合 2、 使...</div> | 来自 |
| <div>常见缓存集锦</div> <div>缓存技术，五花八门,有 memcache、redis、squid、varnish、web cache、CDN等等。缓存技术主要分为三个大类： 一：数据库缓存 二：...</div> | 来自 |
| <div>赛风 Psiphon3下载</div> <div>来自蚂蚁小草 下载地址https://s3.amazonaws.com/psiphon/web/mjr4-p23r-puwl/zh/download.html#direct ...</div> | 来自： www. |
| <div>Vue项目搭建 + 引入ElementUI</div> <div>初始化单页系统 在学习Vue的过程中，官方网站都是给了非常详细的介绍，所以初始化大型单页应用，官网给的参考资料地址：https://cn.vu...</div> | |
| <div>最新最详细的黑苹果10.13.4安装教程</div> <div>最新最全求详细的13.4的黑苹果安装教程，带安装工具和镜像！安装10.13的过程中，在论坛High Sierra版能查到的资料很少，安装教程帖子...</div> | 来自： qq_2 |
| <div>SQL提升（一）</div> <div>Sql不常见关键字提升 一、前言 Sql是最重要的关系数据库操作语言，现在基本上任何与数据库相关的操作都离不开sql。所以说sql功能是很...</div> | 来自： 愤怒 |
| <div>Proxyee-down的下载与安装教程</div> <div>Proxyee-down是monkeyWie在Github上的一个开源项目，向作者致敬。 最新版的Proxyee-down为3.12（2018.10更新），因为作者在3.x后的...</div> | 来自： shade |
| <div>军事理论课答案（西安交大版）</div> <div>1.11 【单选题】我国陆地领土面积排名世界第几？（ C ） A、1 B、2 C、3 D、4 2 【单选题】以下哪个国家不属于金砖五国（ BRICS ）？（ B...</div> | 来自： li |
| <div>【《Unity Shader入门精要》 提炼总结】(十三)第十三章·透明度测试的Shader实现&透明度混合的Shader实现</div> <div>本文由@唐三十胖子出品，转载请注明出处。 文章链接：https://blog.csdn.net/iceSony/article/details/84667551 ...</div> | 来自： 唐 |
| <div>关于PHP在Windows IIS上传的图片401无法访问的解决方法</div> <div>今天项目上线发现上传图片无法正常显示，报401权限问题在这里记录下解决办法 问题总结： 在Windows平台下如果PHP使用的是IIS的话那...</div> | 来自 |
| <div>selenium +java 多个类公用driver问题</div> <div>问题点：太久没有写selenium代码，居然把driver公用的问题忘记了，即：每写一个测试类，执行过程中都会新建一个窗口，这样应该说是非...</div> | 来自： u01 |
| <div>C#Winform窗口移动</div> <div>在我们将Winform自带的边框隐藏之后，我们需要自己编写窗口的移动。 思路就是1.获得点击左键时当前鼠标的坐标 2.获得移动后鼠标的坐...</div> | 来自： N |
| <div>迅雷因版权方无法下载解决方法，附带破解不限速方法</div> <div>迅雷下载有的电影电视剧的时候会出现：应版权方要求，文件无法下载，或者显示迅雷任务包含违规内容 无法继续下载。这个是因为版权方...</div> | 来自： |

| | | |
|---|---|--------------|
| 2018/12/9 | 《计算机操作系统（第四版）》知识点归纳 - JimruEaster的大锅 - CSDN博客 | |
| 李飞飞为什么会离职 Google ? | | 来 |
| ... | | |
| 三分钟小课堂-----docker(二)创建容器 | | |
| 1 打开centos官网 https://www.centos.org/download/ 根据下图所示，查询以前版本，为什么不用最新版？你可以想想！ | 滑动到第二张... | 来自： u01 |
| 原生js模仿jquery Scroll动画 | | |
| 有时候我们在vue项目中需要用到有关scroll的动画,但是又没jquery那样的animate简单的动画方法,又要避免引入jquery的包导致过大,于是就用... | | 来自： 欢迎来到 |
| Redis底层详解（四）整数集合 | | |
| 一、集合概述 对于集合，STL 的 set 相信大家都不陌生，它的底层实现是红黑树。无论插入、删除、查找都是 O(log n) 的时间复杂度。... | | 来自： WhereIsT |
| 一个奇鸽免费版- 船新体验的小工具软件 | | |
| 从名字就不难看出，这是一款有着船新体验，门槛低，需要安卓手机就能正常运行。但相对应的，功能也比同类应用要强大，也更新了多个... | | 来自： 一个奇鸽 |
| Python 凭什么碾压Java、C++等老牌语言？ | | |
| 随着计算机语言的发展，Python也跻身于语言排行的常青树。要是说Python是最目前最火爆的语言，应该没有人反驳吧。在当下的人工智能... | | 来 |
| 教大家一个可以用迅雷全速下载百度网盘文件的方法 | | |
| 本帖最后由 古道吹西风 于 2017-8-28 22:11 编辑 百度的各种限制，相信大家都是苦难言，每年赚那么多的钱，还这样小气，开通会员也一样... | | 来自： |
| 从Microsoft Teams技术栈看前端技术发展趋势 | | |
| 在前不久的微软美国一年一度的Ignite大会上，微软宣布Microsoft Teams是微软历史上发展最快的应用。它将取代Skype for business，成为... | | |
| 安卓手机怎么打开Pornhub看视频 | | |
| 下面是操作方法：第一步：下一个工具（梯，子）：第二步：安装好，如图第三步：选择节点第四步：链接成功，如图第五步：连上之后就... | | 来自： c |