

# 一、操作系统的基本概念

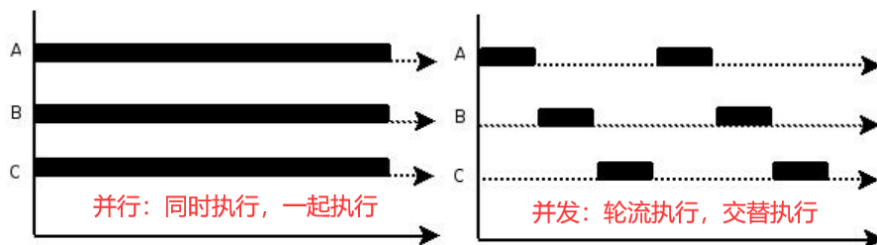
## 1.1 操作系统概念

控制和管理整个计算机系统的硬件与软件资源。合理地组织、调度计算机的工作与资源。为用户和其他软件提供方便接口与环境的程序集合。

## 1.2 操作系统的特征

**特征：并发，共享，虚拟，异步**

**并发：**两个或多个事件在**同一时间间隔内**发生。使得系统具有处理和调度多个程序同时执行的能力。操作系统的并发是通过分时实现的。注意：并发是指在一个时间段，并行是指在同一段。并行是指系统具有同时执行或操作（硬件支持：多流水线或者多处理机）。对于单机处理来说，并行宏观上程序是同时在运行的，微观上程序是交替执行的，多道程序轮流占有CPU，交替执行。



**重要考点：**单核CPU同一时刻只能执行一个程序，各个程序只能并发地执行。多核CPU同一时刻可以同时执行多个程序，多个程序可以并行地执行。

**共享：**互斥共享方式。例如打印机、磁带，同一时刻**只能供一个进程对资源进行访问**。这种资源称作：临界资源或者独占资源。

**同时访问方式：**一段时间内允许多个进程对资源进行访问。典型代表：磁盘设备，重入码编写的文档。

**虚拟：**一个物理上的实体变为若干逻辑上的对应物，这种技术被称为虚拟技术。虚拟处理器：采用多道程序并发的方式，让每个终端用户感觉到有多个处理器。时分复用技术。

**虚拟存储器：**将物理存储变为虚拟存储器，逻辑上扩充存储器用量。空分复用技术。

也可以将一台IO设备虚拟为多台逻辑上的IO设备，并允许每个用户占用一台逻辑上的IO设备。

**异步：**多道程序走走停停，进程以不可预知的速度向前进。

## 1.3 操作系统的运行环境

**程序运行：**程序运行的过程是执行CPU一条条的机器指令的过程。

**操作系统的运行机制：**

CPU执行的两种性质程序，操作系统内核程序，用户自编程序。

内核：

时钟管理：操作系统为用户提供标准时间，根据时钟对进程进行管理，实现进程切换。

中断机制：初衷是为了提高多道程序运行环境中的CPU利用率。保护和回复中断线程的信息，转移控制权到相关程序。

**原语：处于系统的最底层，最接近硬件。运行具有原子性，只能一气呵成。系统控制的数据结构及处理。**

中断与异常：

为了进行内核态和用户态两种状态的切换，引入了中断机制。内核态可以执行用户态无法执行的特权指令。访管指令是在用户态使用，将用户态转换为内核态，所以访管指令不是特权指令。

中断是让操作系统内核夺回CPU使用权的唯一途径。

中断（外中断）：来自于CPU指令之外的事件发生。I/O中断：输入输出已经完成。时钟中断：固定时间片已到，让处理机处理。

**异常（内中断）：源自于CPU执行指令内部的事件。非法操作码，除零，地址越界，算术溢出。陷入指令：用户自行设置，执行陷入后，用户态转换为内核态。异常不能被屏蔽。**

系统调用：用户在程序中调用操作系统提供的一些子功能。设备功能：完成设备的请求或者释放，设备启动等功能。文档管理：完成文档的读、写、创建以及删除功能。进程控制：完成进程的创建、撤销、阻塞以及唤醒功能。进程通信：完成进程之间的消息传递和信号传递功能。内存管理：完成内存的分配、回收以及获取作业占有内存区大小及始址等功能。

用户态与内核态：**处于内核态时，说明此时正在运行的是内核程序，此时可以执行特权指令。**

处于用户态时，说明此时正在执行的时应用程序，此时只能运行非特权指令。

cpu中有一个寄存器叫程序状态字寄存器（psw），其中有个二进制位，1表示内核态，0表示用户态。

**别名：内核态=内核态=管态。用户态=目态。**

内核态->用户态。执行一条特权指令——修改psw的标志位为用户态，意味着操作系统主动让出CPU使用权。

**用户态->内核态。由中断引发，硬件自动完成变态过程，触发中断信号意味着操作系统强行夺回CPU使用权。**

## 1.4 大内核与微内核

大内核：将操作系统的主要功能模块进行集中，从而用以提供高性能的系统服务。优点：各个管理模块间共享信息，能够有效利用相互间的有效特性，有巨大性能优势。缺点：层次交互关系复杂，层次接口难定义，层次间界限模糊。

微内核：计算机体系结构发展，操作系统提供服务越来越多，接口形式越来越复杂。将内核中最基本的功能保留在内核，将不需要在内核态执行的功能转移到用户态执行，降低内核设计的复杂性。优点：有效分离内核与服务、服务与服务、使得它们间的接口更清晰，维护代价大大降低。各部分可以独立的优化和演进。缺点：性能问题，需要频繁的在用户态和内核态之间切换。

## 二、进程

### 2.1 进程的概念和特征

进程的概念：

进程是程序一次执行过程。进程是一次程序及其数据在处理机上顺序执行时发生的活动。进程时具有独立功能的程序在一个数据集上运行的过程，是资源分配和调度的独立单位。是为了更好描述和控制程序的并发执行，实现操作系统的并发性和共享性。

**进程控制块PCB：为了更好的描述进程的基本情况和运行状态，进而控制和管理进程。PCB是进程存在的唯一标志。**

进程的特征：

动态性、并发性、独立性、异步性、结构性。

动态性：进程具有生命周期，有创建、活动、暂停、终止等过程。

并发性：多个进程同时存在于内存中，使得程序与程序之间可以并发执行。

独立性：进程能独立获得资源和接受调度。

异步性：进程相互制约，以不可预知的速度向前推进。操作系统中配置有相应的进程同步机制。

结构性：每个进程都配置一个PCB用于描述。

### 2.2 进程的状态与转换

**状态：创建态、就绪态、运行态、阻塞态、结束态。**

创建态：进程正在被创建，尚未进入就绪态。

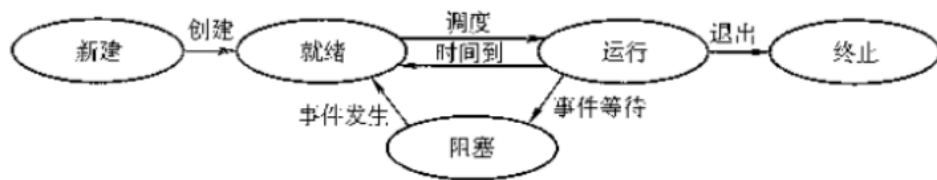
就绪态：进程已处于准备运行状态。

运行态：进程在处理机上运行。

阻塞态：又称等待态，进程正在等待某个时间而暂停运行。

结束态：进程正在从系统中消失（正常结束或异常终止）

相互转换：



就绪态->运行态：处于就绪态的进程获得处理机进入运行态。

运行态->就绪态：处于运行态的进程时间片用完后，让出处理机进入就绪态。

运行态->阻塞态：进程请求除处理机外的其他资源，此时运行态进入阻塞态。

阻塞态->就绪态：进程获得所等待的资源，如IO资源，或中断结束。

## 2.3 进程控制

进程的创建：

分配进程标识号，申请PCB（PCB有限）。为进程分配资源，为程序和数据以及用户栈分配必要的内存空间。初始化PCB，包括初始化标志信息、初始化处理机状态信息、初始化处理机控制信息、设置进程的优先级。若进程就绪队列可以接纳新进程，进程就进入就绪态。

进程的终止：

结束分类：正常结束：进程的任务已经完成并且准备退出运行。异常结束：进程正在运行，出现了某些异常事件，导致程序无法继续运行。外界干预：进程应外界请求终止运行。

**结束过程：**根据被终止进程的标识符，检索PCB，读取进程状态。若进程处于运行态，终止运行，剥夺处理机。终止进程之下的子进程。该进程拥有的全部资源还给父进程或者操作系统。将PCB从队列中删除。

进程的阻塞和唤醒：

阻塞原语执行过程：找到要被阻塞进程标识号对应的PCB。若该进程处于运行态，保护其现场，将其状态转换为阻塞态，停止运行。将PCB插入相应时间的等待队列。

唤醒原语的执行过程：找到等待队列中进程相应的PCB。将其从等待队列中移除，置其状态为就绪态。将PCB插入就绪队列，等待调度程序调度。

进程切换：

进程切换是在内核态下完成的。

过程：保存处理机上下文，包括程序计数器和其它寄存器。更新PCB信息。把进程的PCB移入相应的队列，如就绪、阻塞等队列。选择另一个进程执行，更新其PCB。更新内存管理的数据结构。恢复处理机上下文。

进程的组织：

进程是一个独立的运行单位，也是操作系统进行资源分配和调度的基本单位，由以下三部分组成：

进程控制块：

进程描述信息：进程标识符：标志进程。用户标识符：标识进程归属的用户，主要为了共享和保护服务。

进程控制和管理信息：进程当前状态：描述进程状态信息。进程优先级：描述进程抢占处理机的优先级。代码运行入口地址。程序外存地址。进入内存时间。处理机占用时间。信号量使用。

资源分配清单：

用以说明有关内存地址空间或者虚拟地址空间状况，所打开的文档的列表和所使用的输入/输出设备信息。代码段指针、数据段指针、堆栈段指针、文档描述符、键盘、鼠标。

处理机相关信息：

处理机中各寄存器的值。通用寄存器，地址寄存器，控制寄存器，标志寄存器值，状态字。

程序段：能被进程调度程序调度到CPU执行的程序代码段。

数据段：进程对应的程序加工处理的原始数据或者程序执行时产生的中间或最终结果。

进程的组织方式：

链接方式：按照进程状态将PCB分为多个队列，操作系统由指向各个队列的指针。

索引方式：根据进程状态不同，创建几张索引表。操作系统持有指向各个索引表的指针。

## **2.4 进程的通信**

共享存储：通信进程之间存在一块可以被直接访问的共享空间。

低级方式：基于数据结构共享。

高级方式：基于存储区共享。

消息传递：进程间的数据交换是以格式化的消息为单位的，进程通过系统提供的发送消息和接收消息的两个原语进行数据交换。

直接通信方式：发送进程直接发消息给接收进程，并将它挂在接收进程的消息缓冲队列上，接收进程从消息缓冲队列中取得消息。

间接通信方式：发送进程把消息发给某个中间实体，接收进程从中间实体获得消息。

**信道通信：**发送进程以字符流形式将大量数据送入写管道，接收进程从管道中接收数据。当管道写满时，写进程的write()系统调用将被阻塞，等待读进程将数据取走。当读进程将数据全部取走后，管道变空，此时读进程的read()系统调用将被阻塞。

信道通信是半双工通信，不可以同时读和写。要求互斥和同步，确定对方存在。

## 2.5 进程同步

临界资源：一次只允许一个进程使用的资源（打印机，特殊变量，数据）。

临界资源访问过程：进入区：检查进程是否可以进入临界区。临界区：可以访问临界资源的代码。退出区：将正在访问临界区的标志清楚。剩余区：代码中的其余部分。

互斥：间接制约关系，当一个进程访问临界资源，其它进程不能访问。

同步：直接制约关系，为了完成某个任务而创建多个进程，相互合作，要相互通信同步。

互斥遵循4个原则：

**空闲让进：**临界区空闲时，可以允许一个请求进入临界区的进程进入临界区。

**忙则等待：**已有进程进入临界区后，其他试图进入临界区的进程必须等待。

**有限等待：**对于请求访问临界区的进程，在有限时间内进入临界区。

**让权等待：**进程不能进入临界区时，应当立即释放处理机。

实现临界区互斥的基本方法：软件实现法，硬件实现法。

软件实现法：

单标志法（利用信号量实现同步）：设置公用整型变量turn，用于指示允许进入临界区的进程编号，turn为0，则允许P0进程进入临界区；turn为1，允许P1进程进入临界区。

缺点：可能未被空闲让进（比如turn为0，但P0不需要资源，P1需要资源），违背忙则等待。

双标志法先检查（利用信号量实现互斥）：程序进入临界资源前，先检查临界资源是否被访问，如果空闲才能进入。

缺点：2个进程可能同时进入临界区（同时检测到临界区未被使用），违背忙则等待。

双标志后检查：先设置自己的标志，表明自己想要进入，检查对方标志，如果对方也要进入，那么就等待，否则进入。

缺点：双方可能互相谦让，导致饥饿。

皮特森算法：防止两个进程无限期等待，在算法三的基础上增加一个标志位，从而防止饥饿。

优点：解决了饥饿现象。

硬件实现方法：

中断屏蔽法：对中断进行屏蔽，关中断。

缺点：限制了处理机交替执行程序的能力。

硬件指令法：读出指定标志后，将该标志置为真。

信号量：

整形信号量：wait：资源-1，signal=资源+1。缺点：没有遵循让权等待机制，会导致处于忙等待状态。

记录型信号量：需要一个表示资源数目的value，还需要一个进程链表，用于连接所有等待该资源的进程。

还有利用信号量实现同步和利用信号量实现互斥，以及利用信号量实现前驱关系。

管程：一组数据以及定义在这组数据之上的对这组数据的操作组成的软件模块，这组操作能初始化并改变管程中的数据和同步进程。

## 三、线程

线程的基本概念：

减少程序在并发执行时付出的时空开销，提高操作系统的并发性能。

引入线程后，进程只作为系统资源的分配单元，线程作为处理机的分配单元。

线程与进程的比较：

调度：进程是独立调度的基本单位，线程是资源的基本单位。不同线程切换回引起进程切换。

拥有资源：进程是资源分配的基本单位。

并发性：引入线程后，进程可以并发执行，多个线程间也可以并发执行，提高系统吞吐量。

系统开销：同一进程的线程切换要比进程切换开销小。

**地址空间和其它资源：进程的地址空间相互独立，同一进程的各线程之间共享进程资源（包括地址空间），某进程的线程对其它进程不可见。**

通信方面：进程间通信需要进程同步和互斥手段的辅助，保证数据的一致性。线程间可以直接读/写进程程序段来通信。

线程属性：不拥有系统资源，拥有唯一标识符和线程控制块。不同线程可以执行相同的程序，同一服务程序被不同用户调用时，操作系统为其创建不同线程。同一进程的线程共享该进程全部资源。线程时处理机的独立调度单位。线程也有生命周期，如阻塞、就绪、、运行等状态。多CPU计算机中，各个线程可占用不同的CPU。每个线程都有一个线程ID、线程控制块（TCB）。切换同进程中的线程开销小，切换进程开销大。由于共享内存地址空间，同一进程中的线程间通信甚至无需系统干预。

## 四、死锁

死锁定义：各进程互相等待对方手里的资源，导致各进程都阻塞，无法向前推进。

死锁产生原因：系统资源竞争；进程推进顺序非法。

死锁产生的必要条件：

**互斥条件：**进程对分配的资源进行排他性控制。（只能一个进程使用资源，其它进程不能同时用）

**不可剥夺条件：**进程获得资源在未使用之前，不能被其他进程强行夺走。（假如有进程获得资源，其它进程不能剥夺）

**请求并保持条件：**进程持有至少一个资源，提出新的资源请求，而资源已经被其他进程占用，此时进程被阻塞，但自己已经获得的资源保持不放。（自己占用资源不释放，还请求新的资源）

**循环等待条件：**你等我释放，我等你释放。

死锁的处理策略：

**死锁预防：**破坏四个必要条件其中之一。一次性请求所有资源，资源剥夺，按顺序分配。缺点：效率低不灵活。

破坏互斥条件（一般资源具有独占性，这个较难破坏）。破坏不可剥夺条件，释放已经占用的资源。破坏请求并保持状态，一次性申请完所需的全部资源（如此进程能一次性获取所需的资源）。破坏循环等待条件，采用顺序资源法，对进程进行顺序推荐。

**避免死锁：**在资源动态分配时，用某种方法防止系统进入不安全状态。寻找可能的安全串行，检测分配资源是否会死锁。

系统安全状态，按照某种方式分配资源后，是否会导致死锁，如果死锁，就是不安全。

银行家算法：计算当前资源不同分配方式，看是否会进入不安全状态。

**死锁的检测及解除：**允许进程死锁，及时判断是否死锁，对其进行解除。

资源分配图：圆圈是进程，框是资源。进程到资源称为请求边，资源到进程是分配边。在资源分配图中找到分配满足的进程，消去其请求边与分配。如果所有边可消去，则不存在死锁。

死锁解除：资源剥夺法，挂起某些死锁进程（最后还会恢复），将资源分给其它进程。撤销进程法，强制撤销（直接杀灭，不恢复）部分甚至全部死锁进程，剥夺资源。进程回退法，让一个或多个进程回退到能避免死锁的地步。

## 五、处理机调度

调度的概念：合理的对进程进行处理机分配。

调度的层次：



**作业调度（高级调度）：**从外存中选择作业送入内存，每个作业只调入一次，调出一次。

**内存调度（中级调度）：**将暂时不运行的进程调至外存，使其进入挂起态。或者将已经具备运行条件的进程调入内存，修改其状态为就绪态。

辨析：挂起状态是将进程调到外存进行等待；阻塞状态下进程映像还在内存中。

**进程调度（低级调度）：**按照某种策略或方法从就绪队列中选取一个进程，将处理机分配给它。

三者联系：作业调度为进程活动做准备，进程调度使进程正常活动起来，中级调度将暂时不能运行的进程挂起。

**进程调度方式：**

非剥夺调度方式：如果想将处理机分配给一个更高优先级进程，必须要等待当前占用处理机的进程释放处理机才能将处理机分配给更高优先级进程。

优点：适合批处理系统。缺点：不适合分时系统。

剥夺调度方式：如果有更高级进程请求处理机，暂停正在执行的进程，将处理机分配给更高级进程。

优点：提高系统吞吐率和响应效率。

调度的基本准则：CPU利用率（CPU尽可能处于忙碌状态）；系统吞吐量（单位时间内CPU完成作业的数量）；周转时间（作业提交到作业完成的时间）；等待时间（作业等待处理机的时间）；响应时间（从用户提交请求到首次产生响应的时间）。

**处理机调度算法：**

**先来先服务算法：**先来的先分配处理机。

优点：对长作业有利。

缺点：短作业容易饥饿。

**短作业优先算法：**优先选择运行时间最短的进程。

优先：平均等待时间和周转时间短。

缺点：对长作业不利，容易造成长作业饥饿。

**优先级调度算法：**优先级高的先执行。

可以分为：剥夺型和非剥夺型。

优先级可分为：静态优先级和动态优先级。系统优先级高于用户，交互优先级高于非交互。I/O优先级高于计算。

**高响应比调度算法：**响应比高的先执行。

响应比=（等待时间+要求服务时间）/要求服务时间。

等待时间相同，要求服务时间越短，响应比越高（有利于短作业）。要求服务时间相同，等待时间越长响应比越高（防止饥饿）。

对于长作业，随着等待时间延长，也能具有较高响应比。

特点：不会导致饥饿，非抢占式算法。

**时间片轮转算法：**就绪进程按照到达先后排成队列，依次在时间片内占用处理机，时间片到达时释放处理机。

时间片选择很重要，受进程数、系统处理能力等因素影响。

特点：非抢占式，不会导致饥饿。

**多级反馈队列调度算法：**设置多个就绪队列，每个队列设置不同优先级，优先级递减。每个队列中时间片不同，时间片递减。每个队列按先来先服务原则排队，若规定时间没完成，则放入下一级队列。只有高级队列为空，低等队列才能调度。

特点：产生饥饿现象，是抢占式的。

## 六、虚拟内存管理

**局部性原理：**

**时间局部性：**一条指令被执行或者数据被访问后，不久之后，可能会被再次执行或者访问。（原因：程序中存在大量循环操作）。

**空间局部性：**一旦程序访问了某个存储单元，不久之后附近的存储单元也会被访问到。（原因：指令通常是顺序存放、顺序执行的，数据一般也是以数组等形式存放的）。

虚拟存储器的定义和特征：基于局部性原理，一部分程序装入内存，一部分留在外存，需要的时候再将外存内容调入内存。特征：多次性（作业在运行时，分多次调入内存运行）、对换性（作业不必一直驻留内存，允许作业在运行过程中进行换进换出）、虚拟性（从逻辑上扩充内存容量，使得看到的内存容量大于实际内存容量）。

虚拟内存最大容量由CPU寻址范围决定。虚拟内存的实际容量由min（内存+外存容量之和，CPU寻址范围）决定。

页表有：页号，物理块号，状态位（页是否调入内存），访问字段（本页在一段时间内被访问次数），修改位（记录本页是否被修改过），外存地址（指出该页在外存上的位置）。

**页面置换算法：**

**最佳置换算法（OPT）：**选择永不使用或者最长时间不再访问的页面进行淘汰。

**先进先出页面置换算法（FIFO）：**淘汰最先进入的页面。

**最近最久未使用（LRU）置换算法：**选择最近最长时间没有被访问的页面进行淘汰，每个页面设置一个访问字段，用来标识上次被访问到现在经历的时间。

**时钟（CLOCK）置换算法：**像一个时钟一样转圈，每个页面设置一个使用位，遇到没有被使用的就会将页面换出，使用位置0，如果遇到使用的就会将使用位置零，然后扫描下一个。

优点：性能接近于最佳置换算法。缺点：实现复杂，开销大。

**改进型CLOCK算法：**访问位基础上增加修改位。扫描过程：第1步：扫描缓冲区，先选择未被使用也未被修改的页面换出。第2步：如果未找到，则寻找未被使用，但已修改的页面换出，对每个走过的帧，都将使用位置0。如果还未找到，指针会回到起始地点，然后重复第1步。

分配策略：所谓全局置换，是繁忙进程能够从空闲进程中“借”得物理块。局部置换，是只能从系统预留的物理块中得到物理块。全局置换能够“借”的范围更广（空闲进程+系统）。

**固定分配局部置换：**每个进程分配固定物理块数，缺页时进行换页。

**可变分配全局置换：**进程分配一定物理块，系统保留一定闲置物理块，如果缺页，就对

**可变分配局部置换：**根据进程的缺页情况，对物理块进行动态分配。

**驻留集：**给一个进程分配的物理页框的集合就是进程的驻留集。

**抖动：**刚换出（到磁盘）的页面又要换入内存。原因是：分配的物理页帧数不足或者置换算法不当。

**工作集：**某段时间内，进程要访问的页面集合。

---

## 一、文档系统专题

题1：【2012统考真题】某文档系统空间的最大容量为 4TB（ $1\text{TB}=2^{40}\text{B}$ ），以磁盘块为基本分配单位。磁盘块大小为 1KB。文档控制块（FCB）包含一个 512B 的索引表区。请回答下列问题。

1）假设索引表区仅采用直接索引结构，索引表区存放文档占用的磁盘块号，索引表项中块号最少占多少字节？可支持的单个文档最大长度是多少字节？

2）假设索引表区采用如下结构：第 0 ~ 7 字节采用 < 起始块号，块数 > 格式表示文档创建时预分配的连续存储空间，其中起始块号占 6B，块数占 2B；剩余 504 字节采用直接索引结构，一个索引项占 6B，则可支持的单个文档最大长度是多少字节？为了使单个文档的长度达到最大，请指出起始块号和块数分别所占字节数的合理值并说明理由。

解答：（1）总容量是 $2^{42}\text{B}$ ，磁盘块大小是 $2^{10}\text{B}$ ，所以一共有 $2^{32}$ 个磁盘块。每个磁盘块都可以用一个32位的二进制数表示（比如0000 0000 0000 0000 0000 0000 0010 代表第2个磁盘块）。问：块号占多少字节？因为每个字节8位，所以是 $32/8=4\text{B}$ ，所以占4字节。（可以用4个字节来代表一个磁盘块）

现在有一个512B的索引表区，用 $512\text{B}/4\text{B}=128$ ，求得有128个索引表项，（因为4个字节代表一个磁盘块，4字节等于1个索引表项）每个索引表项代表一个磁盘块，每个磁盘块大小1KB，所以单个文档最大长度位 $128\times 1\text{KB}=128\text{KB}$ 。

（2） $6\text{B}=48\text{位}$ 。 $2\text{B}=16\text{位}$ ，16位表示的是连续的内存块数= $2^{16}$ ，又因为每个内存块是 $1\text{KB}=2^{10}\text{B}$ ，所以预分配的连续内存空间是 $2^{26}\text{B}$ 。

因为1个索引项占6B，所以504字节=84个索引项。由第1问推理可知，1个索引项=1个磁盘块， $84\times 1\text{KB}=84\text{KB}$ 。

由题目第2问开头可知，索引表表示的空间大小可以=预分配的存储空间+直接索引的存储空间。所以可支持的单个文档的最大长度是 $2^{26}\text{B}+84\text{KB}$ 。

在第2问的框架下，索引表结构是不变，即前8字节表示预分配的空间，所以只能在这里作文章。因为根据第1问可知，该系统容量为 $2^{32}$ 个磁盘块的大小。所以我们可以让表示连续分配块数占32位，即4字节，这样最大能分配大小为4TB的连续块作为预分配空间。起始块号也占4字节，刚好能寻址完全 $2^{32}$ 块磁盘块。很完美的分配方法。

题2：【2016 统考真题】某磁盘文档系统使用链接分配方式组织文档，簇大小为4KB。目录文档的每个目录项包括文档名和文档的第一个簇号，其他簇号存放在文档分配表FAT中。

1) 假定目录树如下图所示，各文档占用的簇号及顺序如下表所示，其中dir, dir1是目录，file1, file2是用户文档。请给出所有目录文档的内容。

2) 若FAT的每个表项仅存放簇号，占2B，则FAT的最大长度为多少字节？该文档系统支持的文档长度最大是多少？

3) 系统通过目录文档和FAT实现对文档的按名存取，说明file1的106，108两个簇号分别存放在FAT的哪个表项中



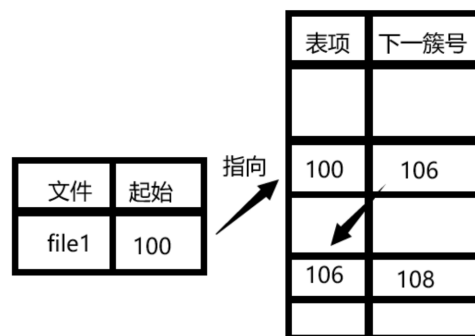
4) 假设仅FAT和dir目录文档已读入内存，若需将文档dir / dir1 / file1的第5000个字节读入内存，则要访问哪几个簇？

解答：（1）可以这么看，题目中只有2个目录，分别为dir和dir1。dir的目录项是dir1（不用去考虑下一级），dir1的目录项是file1和file2。由定义：目录文档的每个目录项包括文档名和文档的第一个簇号，所以dir的目录名为：dir1 48。dir1的目录名为：file1 100 file2 200。

(2)  $2B=16$ 位，任何1个簇能用16位二进制数组成的簇号表示，所以一共能表示 $2^{16}$ 个簇，又因为1个表项对应1个簇，所以一共有 $2^{16}$ 个表项，因为一个表项大小 $2B$ ，所以FAT的最大长度=表项的数量 $\times$ 表项大小 $=2^{16} \times 2 = 2^{17}B=128KB$ 。

最大文档长度=簇的数量 $\times$ 每个簇的大小。已知有 $2^{16}$ 个簇，题目中知道每个簇的大小为 $4KB=2^{12}B$ 。所以所能表示的文档最大大小为： $2^{28}B=256MB$ 。

(3) 由第1问可知，dir1目录文档的第1个目录项为file1 100，这里的100是file1的第1个簇号，它需要通过指针与后面2个簇号106和108连接。file1的目录项首先会指向第1个簇号对应FAT中的表项（如下图），也就是第100表项，因为在FAT表中每个表项的构成结构是：表项+下一簇号，所以100簇号对应的下一个簇号是106，通过106簇号可以找到表项106，108簇号存储在106表项中。由此可以找到file1文档中的所有簇号。所以答案是：file1的簇号106存放在FAT的100号表项中，簇号108存放在FAT的106号表项中。



(4) 由题目知道，dir目录已经读入内存，接下来需要将dir1读入内存，所以需要访问簇号为48的簇。下面需要把file1读入内存，因为每个簇的大小为 $4KB=4096B$ ，系统知道file1的目录项最大只能访问到4096个字节，所以它会到FAT的第100表项去找，所以会访问簇号为106的簇（这个簇是第2个簇，对应的内存是第4097~8192个字节，第5000个字节在这个范围内）。

题3：文档F由200条记录组成，记录从1开始编号。用户打开文档后，欲将内存中的一条记录插入到文档F中，作为其第30条记录。请回答下列问题，并说明理由。

(1)若文档系统采用连续分配方式，每个磁盘块存放一条记录，文档F存储区域前后均有足够的空闲磁盘空间，则完成上述插入操作最少需要访问多少次磁盘块？F的文档控制块内容会发生哪些改变？(2)若文档系统采用链接分配方式，每个磁盘块存放一条记录和一个链接指针，则完成上述插入操作需要访问多少次磁盘块？若每个磁盘块大小为1 KB，其中4个字节存放链接指针，则该文档系统支持的文档最大长度是多少？

解答：（1）明确一下题意，文档F已经写在磁盘上，总共有200条数据，前后均有足够的空间，因此可以移动数据。这里要注意了，是插入数据，不是修改。首先要知道：读、写文档一次都需要访问一次磁盘。

现在的操作是先读第1个数据，然后往前写1位（写到第0位），然后读第2个数据，写到第1个数据的位置上...总共重复了29次。然后把想要插入的数据，写在原本第29个数据所在的位置上（由于之前的前移操作，在该数前面有了29个数，所以该数是第30条数据，符合题意）。

总共进行了29次读写+1次写，所以访问磁盘的次数是 $29 \times 2 + 1 = 59$ 。

(2) 需要访问31次磁盘 (29+2)。前面的29次比较好理解，需要从链表头的第1个结点往后读每个结点的链接指针，因为链接指针指向了后续结点，所以一共读了29次，找到了第29个结点。

然后需要生成一个结点Node，让Node的链接指针指向原先第30个结点（插入后是第31个结点），然后把Node写入到磁盘中（写1次），最后要修改第29个结点的链接指针（写1次），使其指向Node。

所以总共是31次。

题4：某文档系统采用索引结点存放文档的属性和地址信息，簇大小为 4KB。每个文档索引结点占 64B，有 11 个地址项，其中直接地址项 8 个，一级、二级和三级间接地址项各 1 个，每个地址项长度为 4B。请回答下列问题。

(1) 该文档系统能支持的最大文档长度是多少？（给出计算表达式即可。）

(2) 文档系统用 1M ( $1\text{M} = 2^{20}\text{B}$ ) 个簇存放文档索引结点，用 512M 个簇存放文档数据。若一个图像文档的大小为 5600B，则该文档系统最多能存放多少个图像文档？

(3) 若文档 F1 的大小为 6KB，文档 F2 的大小为 40KB，则该文档系统获取 F1 和 F2 最后一个簇的簇号需要的时间是否相同？为什么？

解答：(1) 1个直接地址项指向1个存放数据的簇。1个一级间接地址项指向1个簇，这个簇里存放有地址项，每个地址项指向1个直接地址项。1个二级间接地址项同样也指向1个簇，这个簇里存放有地址项，只是每个地址项指向的是1个一级地址项。

大概关系是：三级间接地址项 包含 二级间接地址项 包含 一级间接地址项 包含 直接间接地址项，1个直接间接地址项指向1个包含数据的簇。

所以推理关系是这样的：每个簇大小4KB，每个地址项长度4B，所以一个簇能包含1024个地址项。先从三级间接地址项开始，1个三级间接地址项=1024个二级间接地址项=1024x1024个一级间接地址项=1024x1024x1024个直接间接地址项。

同理1个二级间接地址项=1024x1024个直接间接地址项。1个一级间接地址项=1024个直接间接地址项。

计算思路：把一级、二级、三级间接地址项全部转化为直接地址项，然后乘以簇的大小（因为1个直接地址项对应1个存放数据的簇）

计算公式：  $(8 + 1024 + 1024 \times 1024 + 1024 \times 1024 \times 1024) \times 4\text{KB}$ 。

(2) 一个簇能存放的索引结点数： $4\text{KB} / 64\text{B} = 2^{12}\text{B} / 2^6\text{B} = 2^6$ 。存放文档结点簇的数量是  $1\text{M} = 2^{20}\text{B}$ 。所以总计是  $2^6 = 64\text{M}$  个索引结点。

簇是文档系统分配空间的最小单位。一个图像文档5600B，不能占满2个簇，但仍要按2个簇算。因为用 512M 个簇存放文档数据，所以能存放256M个图像文档。

但这里要注意了，题目中说的是图像文档，它是一个文档，一个文档必须对应于一个文档索引结点，现在只有64M个索引结点，却有256M个图像文档，因此能存储的文档必须受限于索引结点的个数。

所以只能存储64M个图像文档。

(3) 首先要注意，f1和f2都是文档，是由文档索引结点进行索引，一般索引的顺序是按照：直接地址项索引->一级地址项索引->二级地址项索引...的次序来索引的。

因为题目中直接地址项索引有8个，每个直接地址项索引对应一个存放数据的簇，直接地址项索引到一级间接地址索引的临界值是 $8 \times 4\text{KB} = 32\text{KB}$ ，也就是当索引文档大小大于32KB时就需要使用间接索引。

f1可以通过直接地址索引找到，f2则需要由一级间接地址索引找到。所以f2的搜索会比f1深一层，因此时间不相同。

## 二、内存管理专题

题1：【2009统考真题】请求分页管理系统中，假设某进程的页表内容如下表所示。

页号	页框 (Page Frame) 号	有效位 (存在位)
0	101H	1
1	—	0
2	254H	1

页面大小为4KB,一次内存的访问时间是100ns,一次快表 (TLB) 的访问时间是10 ns, 处理一次缺页的平均时间为 $10^8\text{ns}$  (已含更新TLB和页表的时间), 进程的驻留集大小固定为2, 采用最近最少使用置换算法 (LRU) 和局部淘汰策略。假设TLB初始为空, 地址转换时先访问TLB,若TLB未命中, 再访问页表 (忽略访问页表之后的TLB更新时间)。有效位为0表示页面不在内存, 产生缺页中断, 缺页中断处理后, 返回到产生缺页中断的指令处重新执行。设有虚地址访问串行2362H、1565H 25A5H,请问:

- 1) 依次访问上述三个虚地址, 各需多少时间? 给出计算过程。
- 2) 基于上述访问串行, 虚地址1565H的物理地址是多少?

解本题需要有如下前置知识:

- 1.页表存储在内存中, 存储了虚拟地址和物理地址的映射关系, 通过页号找到页框号, 最后还要将页框号与页内偏移地址拼接得到物理地址, 然后重新访问内存。
- 2.页框是内存中存储数据的基本单元;
- 3.页号是逻辑地址空间中的标识, 作用是将逻辑地址映射到物理地址;
- 4.虚地址由页号 (高位)、页内偏移 (低位) 组成, 页内偏移的位数 (n) 与页面的大小相关 (是2的n次方的n) 。

5.2362H,1565H,25A5H这些都是十六进制数，一个数代表4位。

6.快表一般存储在硬件中，是高速缓存。

7.查找顺序是：快表->页表->外存，快表初始为空

解答：（1）因为页面大小4KB= $2^{12}$ B，根据前置知识第4点，后12位为页内偏移，前4位为页号。

2362H = 0010 0011 0110 0010B，所以页内偏移是0011 0110 0010，页号是0010=2。首先查快表，花费10ns，因页表初始为空，未命中（下面访问内存时会载入）；然后访问内存（页表），花费100ns，页号为2时有效位为1，命中；命中后可以获取页框号，将页框号与页内偏移拼接，得到物理地址，重新访问内存，花费100ns。开销：210ns。

1565H按照上面分析，页号为1，页内偏移565H。首先查找快表，花费10ns，页号为1的项未载入，未命中。访问内存（页表），花费100ns，页号为1时有效位为0，未命中。不得已进行缺页中断，访问外存，开销 $10^8$ ns（题目中说包含载入内存、TLB的时间）。得到页框号后与页内偏移地址拼接，得到物理地址，重新访问内存，开销100ns。开销（ $210+10^8$ ）ns。

25A5H按照上面分析，页号为2，页内偏移5A5H。首先查找快表，花费10ns，页号为2的项在第1次载入了，所以命中；得到了页框号后与页内偏移拼接，然后访问内存，花费100ns。开销110ns。

（2）首先你要看题目中说到，进程的驻留集大小固定为2，所以在内存中最多只能驻留2个页面，所以页表只能有2个页表项，因为页表中原来已有0和2号页表项，在前面访问虚地址1565H时，页号为1的页表项也被载入页表。此时需要淘汰一个页表项，因为2号被访问1次，0号被访问0次，所以淘汰0号页表项，0号的页框101H变成了1号页表项的对应页框。

接下来对1565H按照上面分析，页号为1，页内偏移为565H，页框101H。物理地址：页框+页内偏移=101565H。

题2：【2010统考真题】设某计算机的逻辑地址空间和物理地址空间均为64KB，按字节编址。若某进程最多需要6页(Page)数据存储空间，页的大小为1KB，操作系统采用固定分配局部置换策略为此进程分配4个页框(Page Frame)。见下表，在装入时刻260前，该进程的访问情况也见下表（访问位即使用位）。

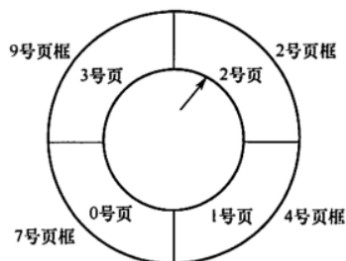
页号	页框号	装入时刻	访问位
0	7	130	1
1	4	230	1
2	2	200	1
3	9	160	1

当该进程执行到时刻260时，要访问逻辑地址为17ACH的数据，请回答下列问题：



(1) 该逻辑地址对应的页号是多少?

(2) 若采用先进先出 (FIFO) 置换算法, 则该逻辑地址对应的物理地址是多少? 要求给出计算过程。若采用时钟(Clock)置换算法, 则该逻辑地址对应的物理地址是多少? 要求给出计算过程。设搜索下一页的指针沿着顺时针方向移动, 且当前指向2号页框, 示意图如下。



解答: (1) 因为页大小是1KB, 所以后10位代表的是页偏移量,  $17CAH = 000101\ 11\ 1100\ 1010B$ , 页号是5, 页内偏移是11 1100 1010。

(2) 先进先出:  $0 \rightarrow 3 \rightarrow 2 \rightarrow 1$ , 置换0号, 所以页框号变成7=0111, 物理地址=页框号+页内偏移=01 1111 1100 1010=1FCA H。(这里要小心, 一定要把页框号和页内偏移化成二进制后再拼接)

首先回顾clock算法, 会用到访问位, 在选择是否换出某个页面的时候, 会看访问位是否为1, 如果是1就不换出这个页面, 但是会把访问位置为0 (清0), 然后检查下一个页面, 直到检查到第1个访问位不为0的页面才将其换出。

在时刻260时, 页号0,1,2,3的访问位都为1, 从2号页开始顺时针访问,  $2 \rightarrow 1 \rightarrow 0 \rightarrow 3$ , 由于这些访问位都为1, 所以都会在访问后置为0, 所以第1次访问到访问位为0的页号是2。此时页框号2被赋予页号5, 物理地址=页框号+页内偏移=00 1011 1100 1010=0BCA H。

题3: 【2012统考真题】某请求分页系统的局部页面置换策略如下: 从0时刻开始扫描, 每隔5个时间单位扫描一轮驻留集(扫描时间忽略不计)且本轮没有被访问过的页框将被系统回收, 并放入到空闲页框链尾, 其中内容在下一次被分配之前不被清空。当发生缺页时, 如果该页曾被使用过且还在空闲页框链表中, 则重新放回进程的驻留集中; 否则, 从空闲页框链表头部取出一个页框。

忽略其他进程的影响和系统开销, 初始时进程驻留集为空。目前系统空闲页框链表中页框号依次为32、15、21、41。进程P依次访问的<虚拟页号, 访问时刻>是:  $\langle 1, 1 \rangle$ 、 $\langle 3, 2 \rangle$ 、 $\langle 0, 4 \rangle$ 、 $\langle 0, 6 \rangle$ 、 $\langle 1, 11 \rangle$ 、 $\langle 0, 13 \rangle$ 、 $\langle 2, 14 \rangle$ 。请回答下列问题:

- 1) 访问 $\langle 0, 4 \rangle$ 时, 对应的页框号是什么?
- 2) 访问 $\langle 1, 11 \rangle$ 时, 对应的页框号是什么? 说明理由。
- 3) 访问 $\langle 2, 14 \rangle$ 时, 对应的页框号是什么? 说明理由。
- 4) 这种方法是否适合于时间局部性好的程序? 说明理由。

解答:

(1) 对应页框：21。(2) 对应页框：32。(3) 对应页框：41。(4) 适合。如果程序的时间局部性越好，从空闲页框链表中重新取回的机会越大，该策略的优势越明显。

下面是推理过程，答案用红色标出。使用<虚拟页号，访问时刻，页框号>的形式来表示，比如<0，4，21>表示虚拟页号为0，访问时刻为4，分配的页框号为21。

初始时空闲页的页框号依次为：32，15，21，41。分配先后顺序是从左到右。

第一轮：1~5 时刻：<1，1，32>，<3，2，15>，<0，4，21>

扫描。

此时空闲页的页框号为：41。

第二轮：6~10时刻：<0，6，21>

扫描。因为虚拟页号1,3在本轮未出现，页框32和15被回收放入到空闲页链表尾（注意题目中说的放入空闲页框链尾），虚拟页号0因为再次出现，页框号没被回收仍是21。

此时空闲页的页框号依次为：41，32，15。

第三轮：11~15：<1，11，32>，<0，13，21>，<2，14，41>

之所以<1，11，32>是因为题目中说：若该页曾被使用过且还在空闲链表中，则重新放回进程的驻留集，因为页号1在第一轮使用过，第二轮未出现，进入空闲页，此时第三轮还未结束进行扫描清算（题目中说：其中内容在下一次分配之前不清空），所以没被清空，沿用32号页框。

之所以<2，14，41>是因为题目中说：否则（指的是该页号没被使用过），从空闲页框链表头部取出一个页框。根据第二轮结束时的排序，链头是41号页框。

题4：某计算机系统按字节编址，采用二级页表的分页存储管理方式，虚拟地址格式如下所示：

10 位	10 位	12 位
页目录号	页表索引	页内偏移量

请回答下列问题：

1) 页和页框的大小各为多少字节？进程的虚拟地址空间大小为多少页？ 2) 假定页目录项和页表项均占 4 个字节，则进程的页目录和页表共占多少页？要求写出计算过程。 3) 若某指令周期内访问的虚拟地址为 0100 0000H 和 0111 2048H，则进行地址转换时共访问多少个二级页表？要求说明理由。

解答本题要有如下前置知识：1.页目录号，用于索引页目录表，页目录表中每个项对应一个页表起始地址。2.页表索引，用于索引页表，页表中每一项对应一个页的起始地址。

解答：

(1) 页框和页大小都是4KB（原因：页框是物理内存的基本单位，而页是虚拟内存的基本单位，页和页框大小是相等的）。

虚拟地址空间大小为 $2^{20}$ 页（原因：虚拟地址空间大小 $2^{10} \times 2^{10} \times 2^{12} = 2^{32}B$ ，每页大小 $2^{12}B$ ，所以地址空间大小为 $2^{32}/2^{12}=2^{20}$ ）。

(2) 答案1025页。页目录号10位，可以表示 $2^{10}$ 个页目录，一个页目录4B，所以页目录大小是： $2^{10} \times 2^2 = 2^{12}B$ ，一个页也是 $2^{12}B$ ，所以页目录占1页。

根据前置知识1首先要知道，页目录表是包含页表的。页目录号10位，页表索引10位，所以页表个数是 $2^{10} \times 2^{10} = 2^{20}$ 个，页表总大小是 $2^{20} \times 2^2 = 2^{22}B$ ，再除以一页的大小 $2^{22} / 2^{12} = 2^{10}=1024$ 页。

总计：1+1024=1025页。

(3) 只需访问一个二级页表，我们可以计算得到，0100 0000H的高10位为转化为十进制是4，0111 2048H的高10位为转化为十进制同样是4。（根据前置知识1，页目录号是用于索引页目录表的，页目录表中每个项对应一个页表的起始地址，比如页目录号如果为4，就能找到页目录表中第4项对应的页表，所以访问的是同一个页表）

**21. 【2020 统考真题】**某 32 位系统采用基于二级页表的请求分页存储管理方式，按字节编址，页目录项和页表项长度均为 4 字节，虚拟地址结构如下所示。

页目录号（10 位）	页号（10 位）	页内偏移量（12 位）
------------	----------	-------------

某 C 程序中数组 a[1024][1024]的起始虚拟地址为 1080 0000H，数组元素占 4 字节，该程序运行时，其进程的页目录起始物理地址为 0020 1000H，请回答下列问题。

- 1) 数组元素 a[1][2]的虚拟地址是什么？对应的页目录号和页号分别是什么？对应的页目录项的物理地址是什么？若该目录项中存放的页框号为 00301H，则 a[1][2]所在页对应的页表项的物理地址是什么？
- 2) 数组 a 在虚拟地址空间中所占的区域是否必须连续？在物理地址空间中所占的区域是否必须连续？
- 3) 已知数组 a 按行优先方式存放，若对数组 a 分别按行遍历和按列遍历，则哪种遍历方式的局部性更好？

(1)  $4 \times 1024 + 2 \times 4 = 4096 + 8 = 5004$

虚拟地址：0001 0000 1000 0000 0001 0000 0000 1000 = 10801008H

页目录号：042H

页号：001H

物理地址：

00301H = 0100 0011 0000 0001 = 4301H

(2) 在虚拟地址不一定连续，在物理地址必须连续

### (3) 行遍历

## 三、同步与互斥

解本题首先需要有如下前置知识：

1.P操作用来请求一个资源或者进入一个临界区域。执行P操作时是将信号量的值减1。如果减1后的值大于等于0则可以继续执行。

2.V操作用来释放一个资源或者离开一个临界资源。执行V操作时是将信号量的值加1。如果有其他进程或线程因为P操作而被阻塞，那么系统会选择一个被阻塞的进程或线程来唤醒，允许它继续执行。

3.Semaohore是信号量的英文单词。

【2011统考真题】某银行提供了 1 个服务窗口和 10 个供顾客等待时使用的座位。顾客到达银行时，若有空座位，则到取号机上领取一个号，等待叫号。取号机每次仅允许一位顾客使用。当营业员空闲时，通过叫号选取一位顾客，并为其服务。顾客和营业员的活动过程描述如下。

```
cobegin
{
    process 顾客 i
    {
        从取号机获取一个号码;
        等待叫号;
        获取服务;
    }
    process 营业员
    {
        While(TRUE)
        {
            叫号;
            为客户服务;
        }
    }
}coend
```

请添加必要的信号量和 P、V 操作或 wait()、signal() 操作，实现上述过程中的互斥与同步。要求写出完整的过程，说明信号量的含义并赋初值。

解答：

Semaphore mutex1=1; //取号机互斥

Semaphore empty=10; //空座位数，最初是10。

Semaphore full=0; //坐了人的座位数，最初是0。

Semaphore mutex2=1; //窗口互斥

```

Cobegin {
    process 顾客 i {
        P(empty);
        P(mutex1);
        从取号机获取一个号码;
        V(mutex1);
        V(full);
        等待叫号;
        P(mutex2);
        获得服务
    }
    process 营业员 {
        while(TRUE){
            P(full);
            叫号;
            V(mutex2);
            为客户服务;
            V(empty);
        }
    }
}
}coend

```

**08.** 已知系统为 32 位实地址，采用 48 位虚拟地址，页面大小为 4KB，页表项大小为 8B，每段最大为 4GB。

- 1) 假设系统使用纯页式存储，则要采用多少级页表？页内偏移多少位？
- 2) 假设系统采用一级页表，TLB 命中率为 98%，TLB 访问时间为 10ns，内存访问时间为 100ns，并假设当 TLB 访问失败时才开始访问内存，问平均页面访问时间是多少？
- 3) 若是二级页表，页面平均访问时间是多少？
- 4) 上题中，若要满足访问时间小于 120ns，则命中率至少需要为多少？
- 5) 若系统采用段页式存储，则每用户最多可以有多少个段？段内采用几级页表？

#### 08. 【解答】

1) 页面大小为 4KB, 因此页内偏移为 12 位。系统采用 48 位虚拟地址, 因此虚页号为  $48 - 12 = 36$  位。采用多级页表时, 最高级页表项不能超出一页大小; 每页能容纳的页表项数为  $4\text{KB}/8\text{B} = 512 = 2^9$ ,  $36/9 = 4$ , 因此应采用 4 级页表, 最高级页表项正好占据一页空间。

2) 系统进行页面访问操作时, 首先读取页面对应的页表项, 有 98% 的概率可以在 TLB 中直接读取到, 然后进行地址转换, 访问内存读取页面; 若 TLB 未命中, 则要通过一次内存访问来读取页表项。页面平均访问时间为

$$98\% \times (10 + 100) + (1 - 98\%) \times (10 + 100 + 100) = 112\text{ns}$$

3) 二级页表的平均访问时间计算同理:

$$98\% \times (10 + 100) + (1 - 98\%) \times (10 + 100 + 100 + 100) = 114\text{ns}$$

4) 设快表命中率为  $p$ , 则应满足

$$p \times (10 + 100) + (1 - p) \times (10 + 100 + 100 + 100) \leq 120\text{ns}$$

解得  $p \geq 95\%$ 。

5) 系统采用 48 位虚拟地址, 每段最大为 4GB, 因此段内地址为 32 位, 段号为  $48 - 32 = 16$  位。每个用户最多可以有  $2^{16}$  段。段内采用页式地址, 与 1) 中计算同理,  $(32 - 12)/9$ , 取上整为 3, 因此段内应采用 3 级页表。

14. Gribble 公司正在开发一款 64 位的计算机体系结构, 也就是说, 在访问内存时, 最多可以使用 64 位的地址。假设采用的是虚拟页式存储管理, 现在要为这款机器设计相应的地址映射机制。

1) 假设页面的大小是 4KB, 每个页表项的长度是 4B, 而且必须采用三级页表结构, 每级页表结构中的每个页表都必须正好存放在一个物理页面中, 请问在这种情形下, 如何实现地址的映射? 具体来说, 对于给定的一个虚拟地址, 应该把它划分为几部分, 每部分的长度分别是多少, 功能是什么? 另外, 采用这种地址映射机制后, 可以访问的虚拟地址空间有多大? (提示: 64 位地址并不一定全部用上。)

2) 假设每个页表项的长度变成了 8B, 而且必须采用四级页表结构, 每级页表结构中的页表都必须正好存放在一个物理页面中, 请问在这种情形下, 系统能够支持的最大的页面大小是多少? 此时, 虚拟地址应该如何划分?

#### 14. 【解答】

- 1) 页面大小为 4KB, 每个页表项大小为 4B, 因此在每个页表当中, 共有 1024 个页表项, 对于每个层次的页表来说, 都满足这一点, 这样每级页表的索引均为 10 位, 由于页面大小为 4KB, 所以页内偏移地址为 12 位。逻辑地址被划分为 5 个部分:

22 位	10 位	10 位	10 位	12 位
空闲	一级索引	二级索引	三级索引	页内偏移

可访问的虚拟地址空间大小为  $2^{42}\text{B} = 4\text{TB}$ 。

- 2) 假定一个页面的大小为  $2^Y$ , 即页内偏移地址为  $Y$  位, 每个页面可以包含  $2^Y/8 = 2^{(Y-3)}$  个页表项, 因此每级页表的索引位为  $Y-3$  位, 共有 4 级页表, 所以  $4(Y-3) + Y \leq 64$ ,  $Y \leq 15.2$ , 因此  $Y = 15$ 。所以最大的页面大小为  $2^{15}\text{B} = 32\text{KB}$ 。

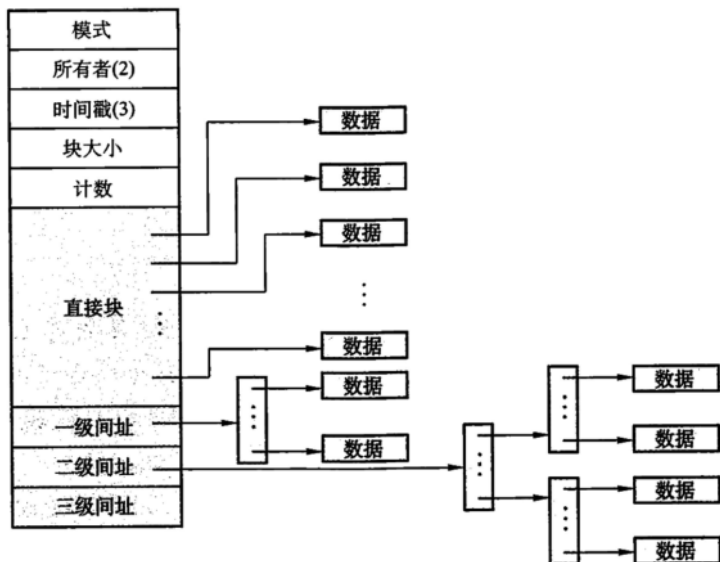
总结: 求解这类题目的关键是清楚地划分逻辑地址, 清楚地划分了逻辑地址的每个部分, 这类题目就很容易求解。

04. 在 UNIX 操作系统中, 给文件分配外存空间采用的是混合索引分配方式, 如下图所示。UNIX 系统中的某个文件的索引结点指示出了为该文件分配的外存的物理块的寻找方法。在该索引结点中, 有 10 个直接块 (每个直接块都直接指向一个数据块), 有 1 个一级间接块、1 个二级间接块及 1 个三级间接块, 间接块指向的是一个索引块, 每个索引块和数据块的大小均为 4KB, 而 UNIX 系统中地址所占空间为 4B (指针大小为 4B), 假设以下问题都建立在该索引结点已在内存中的前提下。

现请回答:

- 1) 文件的大小为多大时可以只用到索引结点的直接块?

- 2) 该索引结点能访问到的地址空间大小总共为多大 (小数点后保留 2 位)?  
 3) 若要读取一个文件的第 10000B 的内容, 需要访问磁盘多少次?  
 4) 若要读取一个文件的第 10MB 的内容, 需要访问磁盘多少次?



#### 04. 【解答】

- 1) 要想只用到索引结点的直接块, 这个文件应能全部在 10 个直接块指向的数据块中放下, 而数据块的大小为 4KB, 所以该文件的大小应小于或等于  $4\text{KB} \times 10 = 40\text{KB}$ , 即文件的大小不超过 40KB 时可以只用到索引结点的直接块。
- 2) 只需要算出索引结点指向的所有数据块的块数, 再乘以数据块的大小即可。直接块指向的数据块数 = 10 块。  
一级间接块指向的索引块里的指针数为  $4\text{KB}/4\text{B} = 1024$ , 所以一级间接块指向的数据块数为 1024 块。  
二级间接块指向的索引块里的指针数为  $4\text{KB}/4\text{B} = 1024$ , 指向的索引块里再拥有  $4\text{KB}/4\text{B} = 1024$  个指针数。所以二级间接块指向的数据块数为  $(4\text{KB}/4\text{B})^2 = 1024^2$ 。

三级间接块指向的数据块数为  $(4\text{KB}/4\text{B})^3 = 1024^3$ 。所以, 该索引结点能访问到的地址空间大小为

$$\left[ 10 + 1 \times \frac{4\text{KB}}{4\text{B}} + 1 \times \left( \frac{4\text{KB}}{4\text{B}} \right)^2 + 1 \times \left( \frac{4\text{KB}}{4\text{B}} \right)^3 \right] \times 4\text{KB} \approx 4100.00\text{GB} = 4.00\text{TB}$$

- 3) 因为  $10000\text{B}/4\text{KB} = 2.44$ , 所以第 10000B 的内容存放在第 3 个直接块中, 若要读取一个文件的第 10000B 的内容, 需要访问磁盘 1 次。
  - 4) 因为 10MB 的内容需要数据块数为  $10\text{MB}/4\text{KB} = 2.5 \times 1024$ , 直接块和一级间接块指向的数据块数 =  $10 + (4\text{KB}/4\text{B}) = 1034 < 2.5 \times 1024$ , 直接块和一级间接块及二级间接块的数据块数 =  $10 + (4\text{KB}/4\text{B}) + (4\text{KB}/4\text{B})^2 > 1 \times 1024^2 > 2.5 \times 1024$ , 所以第 10MB 数据应该在二级间接块下属的某个数据块中, 若要读取一个文件的第 10MB 的内容, 需要访问磁盘 3 次。
05. 有一个文件系统如图 A 所示。图中的方框表示目录, 圆圈表示普通文件。根目录常驻内存, 目录文件组织成链接文件, 不设 FCB, 普通文件组织成索引文件。目录表指示下一级文件名及其磁盘地址 (各占 2B, 共 4B)。下级文件是目录文件时, 指示其第一个磁盘块地址。下级文件是普通文件时, 指示其 FCB 的磁盘地址。每个目录的文件磁盘块的最后 4B 供拉链使用。下级文件在上级目录文件中的次序在图中为从左至右。每个磁盘块有 512B, 与普通文件的一页等长。



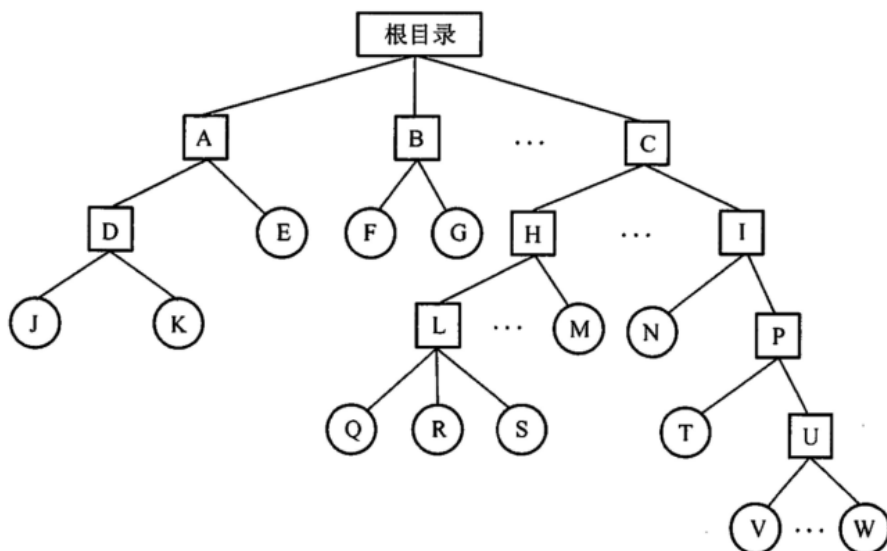


图 A 某树形结构文件系统框图

普通文件的 FCB 组织如图 B 所示。其中，每个磁盘地址占 2B，前 10 个地址直接指示该文件前 10 页的地址。第 11 个地址指示一级索引表地址，一级索引表中的每个磁盘地址指示一个文件页地址；第 12 个地址指示二级索引表地址，二级索引表中的每个地址指示一个一级索引表地址；第 13 个地址指示三级索引表地址，三级索引表中的每个地址指示一个二级索引表地址。请问：

- 1) 一个普通文件最多可有多少个文件页？
- 2) 若要读文件 J 中的某一页，最多启动磁盘多少次？
- 3) 若要读文件 W 中的某一页，最少启动磁盘多少次？
- 4) 根据 3)，为最大限度地减少启动磁盘的次数，可采用什么方法？此时，磁盘最多启动多少次？

该文件的有关描述信息	
1	磁盘地址
2	磁盘地址
3	磁盘地址
⋮	⋮
11	磁盘地址
12	磁盘地址
13	磁盘地址

图 B FCB 组织

#### 05.【解答】

- 1) 因为磁盘块大小为 512B, 所以索引块大小也为 512B, 每个磁盘地址大小为 2B。因此, 一个一级索引表可容纳 256 个磁盘地址。同样, 一个二级索引表可容纳 256 个一级索引表地址, 一个三级索引表可容纳 256 个二级索引表地址。这样, 一个普通文件最多可有文件页数为  $10 + 256 + 256 \times 256 + 256 \times 256 \times 256 = 16843018$ 。
- 2) 由图可知, 目录文件 A 和 D 中的目录项都只有两个, 因此这两个目录文件都只占用一个物理块。要读文件 J 中的某一页, 先从内存的根目录中找到目录文件 A 的磁盘地址, 将其读入内存 (已访问磁盘 1 次)。然后从目录 A 中找出目录文件 D 的磁盘地址读入内存 (已访问磁盘 2 次)。再从目录 D 中找出文件 J 的 FCB 地址读入内存 (已访问磁盘 3 次)。在最坏情况下, 该访问页存放在三级索引下, 这时候需要一级级地读三级索引块才能得到文件 J 的地址 (已访问磁盘 6 次)。最后读入文件 J 中的相应页 (共访问磁盘 7 次)。所以, 若要读文件 J 中的某一页, 最多启动磁盘 7 次。
- 3) 由图可知, 目录文件 C 和 U 的目录项较多, 可能存放在多个链接在一起的磁盘块中。在最好情况下, 所需的目录项都在目录文件的第一个磁盘块中。先从内存的根目录中找到目录文件 C 的磁盘地址并读入内存 (已访问磁盘 1 次)。在 C 中找出目录文件 I 的磁盘地址并读入内存 (已访问磁盘 2 次)。在 I 中找出目录文件 P 的磁盘地址并读入内存 (已访问磁盘 3 次)。从 P 中找到目录文件 U 的磁盘地址并读入内存 (已访问磁盘 4 次)。从 U 的第一个磁盘块中找出文件 W 的 FCB 地址并读入内存 (已访问磁盘 5 次)。在最好情况下, 要访问的页在 FCB 的前 10 个直接块中, 按照直接块指示的地址读文件 W 的相应页 (已访问磁盘 6 次)。所以, 若要读文件 W 中的某页, 最少启动磁盘 6 次。
- 4) 为了减少启动磁盘的次数, 可以将需要访问的 W 文件挂在根目录的最前面的目录项中。此时, 只需读内存中的根目录就可找到 W 的 FCB, 将 FCB 读入内存 (已访问磁盘 1 次), 最差情况下, 需要的 W 文件的那个页挂在 FCB 的三级索引下, 因此读 3 个索引块需要访问磁盘 3 次 (已访问磁盘 4 次) 得到该页的物理地址, 再去读这个页即可 (已访问磁盘 5 次)。此时, 磁盘最多启动 5 次。

#### 01.【解答】

用于设备分配的数据结构有系统设备表 (SDT)、设备控制表 (DCT)、控制器控制表 (COCT) 和通道控制表 (CHCT)。

SDT 整个系统中只有一张, 它记录系统中全部设备的情况, 是系统范围的数据结构。每个设备有一张 DCT, 系统为每个设备配置一张 DCT, 以记录本设备的情况。每个控制器有一张 COCT, 系统为每个控制器都设置一张用于记录本控制器情况的 COCT。系统为每个通道配置一张 CHCT, 以记录通道情况。SDT 中每个表目有一个指向 DCT 的指针, DCT 中的每个表目有一个指向 COCT 的指针, COCT 中有一个 CHCT 指针, CHCT 中有一个 COCT 指针。