哈尔滨工业大学计算学部

2020年春季学期

操作系统设计与实现课程设计

**操作系统核心功能动态运行时分析与设计**

姓名：贺宗磊

学号：11703004421

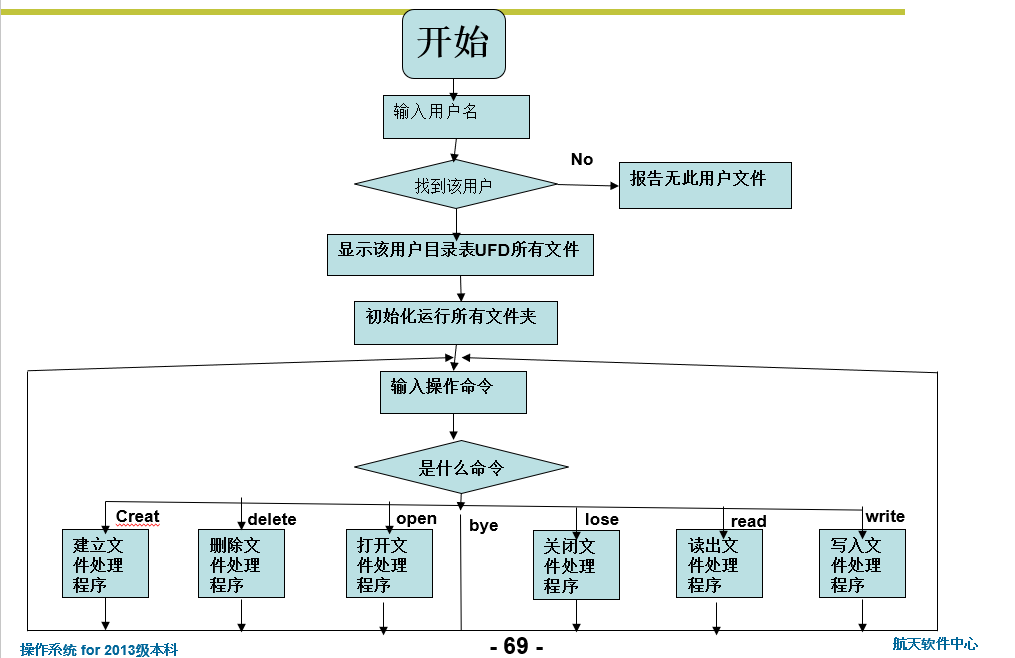
摘要

文件读取是操作系统的必备部分，用户通过对文件的存取来完成对文件的修改、追加和搜索等操作。在一台多核处理器电脑，内存较小，硬盘上存储多个可执行的程序（采用多线程设计），每个可执行程序文件都很大，运行时都需要读写文件过程中，在这个过程当中，操作系统的动态运行时核心过程是调度进程，决定哪个进程运行、挂起、交换等；分配内存，哪个进程得到内存；管理和控制文件系统；检查“许可证”、修改目录、路径等；处理系统调用：由用户的进程发“请求”，系统根据资源的充分利用，统筹安排；处理输入输出的请求和工作。本文简要设计核心流程，并对其中的关键技术，如这文件与文件系统、虚拟内存、进程管理、和简单IO知识等进行简要介绍和设计。

**正文：**

**一：核心流程及设计：**

操作系统的动态运行时核心过程是调度进程，决定哪个进程运行、挂起、交换等；分配内存，哪个进程得到内存；管理和控制文件系统；检查“许可证”、修改目录、路径等；处理系统调用：由用户的进程发“请求”，系统根据资源的充分利用，统筹安排；处理输入输出的请求和工作。因为本次是对文件进行读取。常用的存取方法有顺序存取法、随机存取法（直接存取法）和按键存取法三种。顺序存取法是按照文件的逻辑地址顺序存取，在记录式文件中，这反映为按记录的排列顺序来存取，在无结构的字符流文件中，顺序存取反映当前读写指针的变化。随机存取法允许用户根据记录的编号来存取文件的任一记录，或者是根据存取命令把读写指针移到欲读写处来读写。按键存取法是一种用在复杂文件系统，特别是数据库管理系统中的存取方法，文件的存取是根据给定的键或记录名进行的。在此，我们按如下核心流程读写文件。



**二：关键技术——文件与文件系统**

先说文件定义，文件是记录在外存上的相关信息具有名称的集合。这里有一点，文件是存放在外存的，这是由于内存的断电易失性质导致的，虽然内存速度快，但是仍然需要文件来稳定地存放文件。而对于用户来说，文件是逻辑外存地最小分配单元，即数据只能通过文件的方式写到外存。

文件系统的结构包括不同的层次，I/O控制为最底层，基本文件系统负责向设备驱动程序发送一般命令，文件组织模块则知道文件机器逻辑和物理块，可以将逻辑块地址转化为物理块地址，逻辑文件系统管理元数据，包括文件系统的所有结构数据。通过分层结构管理，能减少重复代码。例如相同的I/O控制模块可以被多个文件系统采用。

文件有一定的属性，一般来说包括：名称，标识符（唯一标签，通常为数字），类型，位置，大小，保护，时间、日期和用户标识等。

文件作为一种抽象数据类型，显然需要定义一些基本的操作，具体应当包括：创建，写，读，重定位，删除，截短6个基本操作。这些文件操作总是涉及为给定文件搜索相关目录条目，这种情况下，操作系统维护一个包含所有打开文件的信息表——打开文件表，就会避免一些不必要的搜索操作。这个表的建立主要是通过在首次打开文件时，使用系统调用open()。

 open()系统调用。可以分为五步：（1）进程向os提供文件路径名（2）os查询硬盘上的文件属性（3）从硬盘返回文件属性（4）os把一个文件描述符和这个文件属性联系起来，一般来说是一个number（5）os把文件描述符返回给进程，这时进程就取得了这个文件，可以通过这个描述符访问文件具体内容。同样，进程同样维护一个内部的打开文件表，并指向系统表的相应条目。应当注意到，这个过程中实际上只涉及了路径名、文件属性，并没有具体的文件内容。

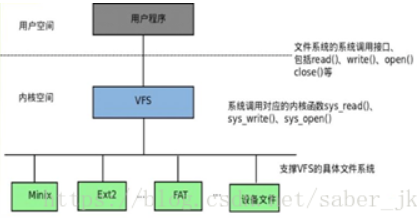
  文件是用来存储信息的，既然有了文件的属性信息、基本操作、类型结构等，下面就要谈及文件的访问，否则文件没有存在的意义。这里主要从访问方法的层面来分析。在支持多种访问方式的系统中，针对不同应用访问方法的选择尤为重要。（1）顺序访问（2）直接访问（3）其他访问，一般建立在直接访问方式之上，通常设计文件索引，包含各块指针的结构称为索引包。针对大文件，索引较大难以存放在内存中，可以创建多级索引，即为索引文件再创建索引。

因为当今计算机文件系统十分庞大，文件数目繁多，故而产生了目录，目的是更好地组织文件并加以管理。目录应包含相关操作操作：搜索文件、删除文件、创建文件、遍历目录、重命名文件、跟踪文件系统，这些操作在设计目录逻辑结构时要着重考虑。定义目录结构地常用方案有：单层结构目录，双层结构目录，树状结构目录，无环图目录，通用图目录。前三种方案可以看作逐渐拓展，本质都属于树状结构。

文件在计算机系统中，需要有一定的方式保护其安全，包括可靠性和防止非法访问两方面。前者通常通过文件备份来实现。后者对不同的访问类型加以区分并进行限制。访问类型有：读、写、执行、添加、删除、列表清单。实现访问控制的手段最普通的方案就是为每个目录、文件增加一个访问控制表，针对每一个用户设置权限。这样比较没有效率，精简方法就是分组，这实际上在前面文件共享部分有所提及。如下最左侧的属性实际就是访问权限的设置。r是读，w是写，x是执行，三个为一组，从左到左右依次是拥有者、组成员、其他访问者的权限。

在磁盘上，文件系统可能包含的信息有：如何启动存储的操作系统、总的块数、空闲块的数目和位置、目录结构和具体文件。

把多个文件系统整合成一个目录结构是需要解决的问题，如果采用为每个类型编写目录和文件程序是一种显然的方法但是不能令人满意。我们可以采用一种方法，将基本系统调用的功能和实现细节分开，这就是虚拟文件系统技术。在这种技术下，文件系统实现分为三层，第一层为文件。如下图：



系统接口，第二层是虚拟文件系统层，其目的有二：（1）定义一个清晰的VFS结构，分开文件系统的通用操作和具体实现（2）提供了在网络上唯一标识一个文件的机制。第三层实现文件系统类型或远程文件系统协议。

目录分配和管理是文件系统中重要的一环。在本次中我们使用存储文件名和数据块指针的线性列表，这种方法有致命的缺点是需要线性搜索，访问速度会比较慢。除此之外，另一个用于文件目录的结构是哈希表，根据文件名得到一个值，并返回一个指向线性列表中元素的指针，这就大大减少了目录搜索时间。

**三：关键技术——虚拟内存**

虚拟内存是计算机系统内存管理的一种技术。它使得应用程序认为它拥有连续的可用的内存（一个连续完整的地址空间），而实际上，它通常是被分隔成多个物理内存碎片，还有部分暂时存储在外部磁盘存储器上，在需要时进行数据交换。目前，大多数操作系统都使用了虚拟内存，如Windows家族的“虚拟内存”；Linux的“交换空间”等。设计的工作原理如下：

工作原理

虚拟存储器是由硬件和操作系统自动实现存储信息调度和管理的。它的工作过程包括6个步骤：

①中央处理器访问主存的逻辑地址分解成组号a和组内地址b，并对组号a进行地址变换，即将逻辑组号a作为索引，查地址变换表，以确定该组信息是否存放在主存内。

②如该组号已在主存内，则转而执行④；如果该组号不在主存内，则检查主存中是否有空闲区，如果没有，便将某个暂时不用的组调出送往辅存，以便将这组信息调入主存。

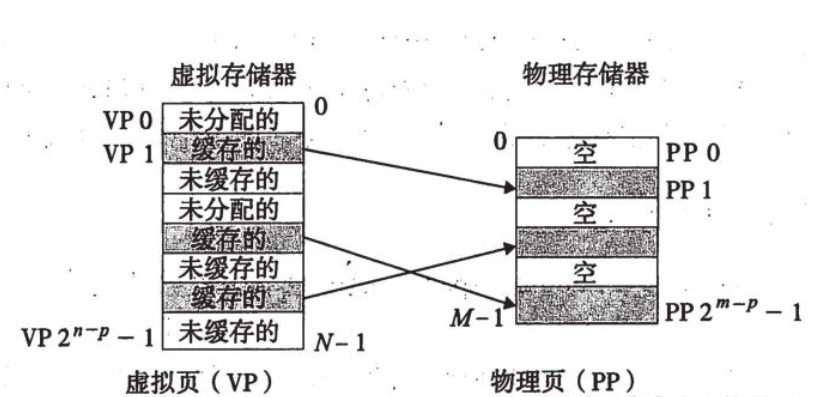
③从辅存读出所要的组，并送到主存空闲区，然后将那个空闲的物理组号a和逻辑组号a登录在地址变换表中。 [3]

④从地址变换表读出与逻辑组号a对应的物理组号a。

⑤从物理组号a和组内字节地址b得到物理地址。

⑥根据物理地址从主存中存取必要的信息

一个应用示例：



在访问过程中虚存空间的用户程序按照虚地址编程并存放在辅存中。程序运行时，由地址变换机构依据当时分配给该程序的实地址空间把程序的一部分调入实存。每次访存时，首先判断该虚地址所对应的部分是否在实存中：如果是，则进行地址转换并用实地址访问主存；否则，按照某种算法将辅存中的部分程序调度进内存，再按同样的方法访问主存。由此可见，每个程序的虚地址空间可以远大于实地址空间，也可以远小于实地址空间。后一种情况通常出现在多用户或多任务系统中：实存空间较大，而单个任务并不需要很大的地址空间，较小的虚存空间则可以缩短指令中地址字段的长度。有了虚存的机制后，应用程序就可以透明地使用整个虚存空间。对应用程序而言，如果主存的命中率很高，虚存的访问时间就接近于主存访问时间，而虚存的大小仅仅依赖于辅存的大小。

**四：关键技术——进程管理**

进程是正在运行的程序实体，并且包括这个运行的程序中占据的所有系统资源，比如说CPU（寄存器），IO,内存，网络资源等。很多人在回答进程的概念的时候，往往只会说它是一个运行的实体，而会忽略掉进程所占据的资源。比如说，同样一个程序，同一时刻被两次运行了，那么他们就是两个独立的进程。linux下查看系统进程的命令是ps

并发程序和顺序程序有本质上的差别，为了能更好地描述程序的并发执行，实现操作系统的并发性和共享性，引入“进程”的概念。

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，是系统进行资源分配和调度的一个独立单位。

处理器是计算机系统中最重要的资源。在现代计算机系统中，为了提高系统的资源利用率，CPU将为某一程序独占。通常采用多道程序设计技术，即允许多个程序同时进入计算机系统的内存并运行。在本系统中进程流程：Csrss.exe：这是子系统服务器进程，负责控制Windows创建或删除线程以及16位的虚拟DOS环境。System Idle Process：这个进程是作为单线程运行在每个处理器上，并在系统不处理其它线程的时候分派处理器的时间。Smss.exe：这是一个会话管理子系统，负责启动用户会话。Services.exe：系统服务的管理工具。Lsass.exe：本地的安全授权服务。Explorer.exe：资源管理器。Spoolsv.exe：管理缓冲区中的打印和传真作业。Svchost.exe：这个进程要着重说明一下，有不少朋友都有这种错觉：若是在“任务管理器”中看到多个Svchost.exe在运行，就觉得是有病毒了。其实并不一定，系统启动的时候，Svchost.exe将检查注册表中的位置来创建需要加载的服务列表，如果多个Svchost.exe同时运行，则表明当前有多组服务处于活动状态；多个DLL文件正在调用它。

**五：关键技术——I/O 缓冲区**

IO缓冲区概念与如高速缓存（cache）产生的原理类似，在I/O过程中，读取磁盘的速度相对内存读取速度要慢的多。因此为了能够加快处理数据的速度，需要将读取过的数据缓存在内存里。而这些缓存在内存里的数据就是高速缓冲区下面简称为“buffer”。

具体来说，buffer（缓冲区）是一个用于存储速度不同步的设备或优先级不同的设备之间传输数据的区域。一方面，通过缓冲区，可以使进程之间的相互等待变少，从而使从速度慢的设备读入数据时，速度快的设备的操作进程不发生间断。另一方面，可以保护硬盘或减少网络传输的次数。

buffer和cache是两个不同的概念：cache是高速缓存，用于CPU和内存之间的缓冲；buffer是I/O缓存，用于内存和硬盘的缓冲；简单的说，cache是加速“读”，而buffer是缓冲“写”，前者解决读的问题，保存从磁盘上读出的数据，后者是解决写的问题，保存即将要写入到磁盘上的数据。

buffer cache和page cache都是为了处理设备和内存交互时高速访问的问题。buffer cache可称为块缓冲器，page cache可称为页缓冲器。在linux不支持虚拟内存机制之前，还没有页的概念，因此缓冲区以块为单位对设备进行。在linux采用虚拟内存的机制来管理内存后，页是虚拟内存管理的最小单位，开始采用页缓冲的机制来缓冲内存。Linux2.6之后内核将这两个缓存整合，页和块可以相互映射，同时，页缓存page cache面向的是虚拟内存，块I/O缓存Buffer cache是面向块设备。需要强调的是，页缓存和块缓存对进程来说就是一个存储系统，进程不需要关注底层的设备的读写。

buffer cache和page cache两者最大的区别是缓存的粒度。buffer cache面向的是文件系统的块。而内核的内存管理组件采用了比文件系统的块更高级别的抽象：页page，其处理的性能更高。因此和内存管理交互的缓存组件，都使用页缓存。