**计算机系统结构-存储系统章节报告**

1. 题目背景
   1. 存储器的概念

现代计算机系统都以存储器为中心,这与冯诺依曼计算机以运算器为中心不同.一个程序要想运行,必须将程序主体部分和数据部分都加载到存储器之后,才能正常执行.在程序的执行过程中, 中央处理器需要从存储器获取程序指令来执行, 运算得到的中间结果和最终结果也会存储到存储器中, 而且各种输入输出设备所需的数据也需要直接从存储器中获取, 所以存储器是计算机体系结构中的中心部分.

* 1. 存储系统的概念

存储系统是指包含了两个或两个以上速度, 价格和容量不相同的存储器并且使用硬件, 软件或硬件和软件相结合的方式连接起来的一个完整的系统.在一个存储系统中, 内部的各个存储器对外部是透明的, 站在应用程序员的角度来看, 整个存储系统就是一个存储器, 这个存储器的访问速度近似等于各个存储器中存储周期最小的那一个, 存储容量近似的等于所有存储器中容量最大的那一个, 价格近似地等于所有存储器中最便宜的那一个.

* 1. 存储器和存储系统的关系

存储器是指单个用于存储数据的硬件设备, 而存储系统是指使用了硬件,软件或软硬结合的方式对多个存储器进行集成的一个系统.

* 1. 存储系统的分类

存储系统可分为两类: 一是由Cache和主存储器构成的Cache存储系统, 另外一种是由主存储器和硬盘存储器构成的虚拟存储系统.这两种存储系统的用途各不相同. Cache存储系统主要的功能是提高存储速度, 虚拟存储系统的功能主要是增大存储容量.从价格上来看, 由于Cache存储器采用和存储器基本相同额结构, 所以Cache存储器的价格昂贵, 而硬盘存储器本身存储速率很低, 所以价格低廉.

* 1. 存储器的层次结构

在计算机的存储体系结构中, 可根据存储速度, 访问方式,存储容量等因素将不同的存储器分为不同的层次. 根据访问速度从高到低, 可将存储器分为通用寄存器, 指令和数据缓冲栈, Cache, 主存储器, 外部存储器. 因为存储速率的不同, 这些存储器的存储容量和单位价格也各不相同. 因此, 在对计算机的存储系统进行设计时, 应当综合各方面因素, 并结合实际的应用场景, 给出符合定位的存储系统设计方案.

* 1. 虚拟存储器

虚拟存储器又被称为虚拟存储系统, 虚拟存储器主要由主存储器和联机工作的外部存储器构成. 其中主存储器通常使用动态随机存储器实现, 联机工作的外部存储器通常是硬盘存储器. 虚拟存储器的主要实现方式是将数据存储到联机工作的外部存储器中, 然后在主存储器中存储数据在联机工作的外部存储器中的存储地址, 当需要访问数据时, 根据映射的地址取数据.

1. 涉及到的问题及其解决方案

2.1 如何提高存储器的存储速度?

2.1.1 使用并行访问存储器

要想在存储器的一个存储周期中访问到更多的数据, 最直接的方法就是增加存储器的步长, 即增长存储器每次访问数据的字长, 这样就可以使得存储在一个存储周期内访问更多的数据, 从而提高存储速率.

这种方式的优点是实现容易,操作简单.

缺点有如下:

1. 容易发生取指令冲突. 在遇到程序转移而且转移成功时, 一个存储周期中读出的n条指令中,后面的指令将会无效.
2. 容易发生读操作数冲突. 一次读出的操作数, 并不一定都是有用的.
3. 写数据冲突. 这种访问数据的方式必须凑齐了规定步长的数据之后才会写入存储器, 如果只写一个字, 就大大增加了存储过程的操作步骤.
4. 读写冲突. 当要读出的一个字和要写入的一个字处于同一个存储字内时, 无法在同一个存储周期内完成.

解决以上可能发生的问题, 需要将存储系统设计为多套独立的地址寄存器和读写控制逻辑.

2.1.2 交叉访问存储器

交叉访问存储器分为高位交叉访问存储器和低位交叉访问存储器.

高位交叉存储器即交叉使用多个存储器的高位存储空间, 从而达到扩大存储容量的目的, 同时也可以通过将不同任务分配给不同的存储器中来提高访问器的存储速度.

低位交叉存储器即交叉使用多个存储器的低位存储地址, 该设计方法主要的目的就是提高存储速度.通过共享指令的方式使得多个存储器可协同工作.

2.2 在虚拟存储器结构中, 如何加快内部地址变换?

在虚拟存储器中, 如果不采取有效的措施, 对主存储器的数据访问速度将会降低几倍, 这并不符合存储系统的设计要求, 因为在存储系统的设计中, 要求整个存储系统的存储速度要接近各个存储器中存储速度最高的那一个.

造成虚拟存储器速度降低的原因:

1. 在段式或页式虚拟存储器中, 要想访问主存储器就要先访问段表或页表, 在段页式虚拟存储器中, 既要查看段表, 也要查看页表, 这就大大降低了数据的访问速度.
2. 当页表或段表的容量超过了一个页面的大小时, 它们就可能被映像到主存储器不连续的页面位置上, 这样在对页表和段表查询时, 效率就会大打折扣.

解决方案:

1. 采用目录表的存储方式

采用目录表的存储方式即压缩页表的大小, 然后使用一个高速存储器来存放页表, 从而加快查询页表的速度.该方法的缺点为如果页表过大, 那么使用高速存储器的方式会使得成本急剧上升.

1. 采用快慢表的存储方式

快慢表的存储方式是指将页表分为小容量且查询速度快的快表和大容但查询速度慢的慢表, 然后将页表中经常被查询的数据存储到块表中, 而慢表则存储页表中的所有数据, 当对页表数据进行查询时, 先去快表中查询, 如果查询不到则转到慢表查询.

2.3 在虚拟存储器结构中,使用什么样的规则进行页面替换?

2.3.1 随机替换算法

利用软件或硬件的随机数发生器来确定主存储器中被替换的页面, 这种方式实现简单, 但是由于是完全随机, 所有并不能反映页面调度的历史信息, 也不能提高页面的命中率.

2.3.2 先进先出替换算法

这种算法在进行页面替换时, 采用的是先进入主存储器的页面将会被先替换掉的方式, 这种方式实现简单, 而且能够反映页面的历史调用情况, 但是不能根据页面的使用频率而选择更优的替换方式.

2.3.3 近期使用最少替换算法

这种算法会将最近一定时间段内对页面的调用次数进行统计, 在对页面进行替换时, 会优先选择最近时间段内使用次数最少的方式, 这种方式能更好的反映页面的使用频率情况, 并根据该情况替换页面. 但是这种方式实现复杂, 并且需要选择合适的时间长度作为统计的时间段.

2.3.4 最久没有使用页面替换算法

这种替换算法会将近期一定时间段内最久没有访问过的页面进行替换, 与近期使用最少算法的区别在于该算法以没有使用过作为判断依据, 因此实现较近期使用最少算法相对简单.

2.3.5 最优替换算法

这种方式是选择将来最久不会被访问的页面进行替换, 因为替换掉的是将来最久不会被访问的页面, 所以这种方式最能够提高命中率.

2.3.6 堆栈型替换算法

这种替换页面的方式不同于先进先出的替换结构, 而是采用堆栈的替换结构, 即先进后出, 最近加载的页面将会被最先替换.

2.4 如何提高主存储器的命中率?

因为主存储器的命中率与存储系统的存储速度息息相关, 所以提高主存储器的命中率是一个关键的问题.

影响主存储器命中率的因素:

1. 程序在执行过程中的页面地址流分布的情况
2. 所采用的页面替换算法
3. 页面大小
4. 主存储器的容量
5. 所采用的页面调度方法

2.4.1 采用合适的页面大小

页面大小与命中率并不是线性的, 页面大小还与主存储器的容量和程序的页地址流分布情况等因素相关. 因此在设计页面大小时, 要通过对典型程序的模拟实验结果来进行确定.

2.4.2 调整合适的主存容量

通过实验得出, 主存命中率随着分配给程序的主存容量的增加而单调增加, 但是当分配的主存达到某一个阈值时, 命中率的增加会变得微乎其微, 因此选择合适的分配主存容量能够得到命中率与成本的权衡.

2.4.3 使用预取式页面调度方式

预取式的页面调度方式是根据程序的特点, 在程序被挂起而未重新开始运行之前, 把上次停止前用到的页面事先加载到主存储器中, 此时如何程序重新开始运行, 因为主存储器中已经包含一定数量的页面, 就可以提高页面的命中率.

1. 结论

存储系统作为计算机体系结构中的核心部分, 其工作效率和准确率是提升整个计算机性能的主要因素. 而对于不同的应用场景, 我们需要选择不同的存储系统结构.对于不同的实际问题, 还需要具体选择不同的存储方法.

参考文献:

Ⅰ: 计算机系统结构（清华大学第二版）

Ⅱ: Fujitsu Technology Solutions Intellectual Property GmbH; Patent Issued for Method Of Improving Access To A Main Memory Of A Computer System, A Corresponding Computer System And A Computer Program Product (USPTO 10,248,329) ---- Computers, Networks & Communications

Ⅲ: 计算机存储系统的管理技术 ---- 王延军