计算机组成
(2022秋)

计算机组成课程组

（刘旭东、高小鹏、肖利民、栾钟治、万寒）

北京航空航天大学计算机学院中德所

栾钟治

➢1

习题7——Cache与虚存

- 已发布
  - Spoc平台
- 12月09日截止
  - 23:55
- 在sopc提交
  - 电子版，可手写

➢2

回顾：虚拟存储

- 虚拟存储的动机
  - 多道程序（进程）同时运行时如何共享存储器？
  - 如何消除小的主存容量对编程的限制？
  - 在多任务系统中，虚拟存储为每个进程提供了一个大的、私有的、统一的内存空间幻象
- 解决之道
  - 把主存当做辅助存储器的高速缓存，称为虚拟存储
  - 命名和保护
    - 每个进程都看到一个大的、连续的地址空间（为了方便）
    - 每个进程的内存都是私有的，即受保护不被其他进程访问（为了共享）
  - Trade-off
    - 辅助存储的容量(磁盘上的交换空间)
    - 主存储的速度（DRAM）

➢3

回顾：虚拟存储基础

- 虚存空间与物理空间
  - 用户编程空间：虚地址或逻辑地址，对应的存储空间在辅存中
  - 物理内存空间：实地址或物理地址，对应的存储空间在主存中
- 虚拟存储器要解决的问题
  - 虚存空间与物理空间之间的数据交换
  - 虚地址与实地址的转换问题
  - 缺失处理和替换策略
- 基准：地址翻译
  - 通过地址翻译实现大的、私有的、统一的抽象
    - 在每次内存引用时硬件将有效地址（EA）翻译成物理地址（基和界）
  - 通过地址翻译
    - 控制进程可以引用哪些物理位置(DRAM和/或交换磁盘)
    - 允许动态分配和重新定位物理存储(在DRAM和/或交换磁盘中)
    - 地址转换的硬件和策略由操作系统控制，受用户保护

➢4

回顾：虚拟存储的调度

❖虚拟存储器的调度方式

- 段式调度：按程序的逻辑结构将程序空间划分为若干段，段的长度是随意的，虚存空间和物理空间按段进行交换
- 页式调度：将虚存空间和物理地址空间都分成固定大小的页。主存按页顺序编号；每个独立编址的程序空间也按自己的页顺序编号。虚存空间和物理空间按页进行交换
- 段页式调度：两种方法的结合。在段页式调度中把物理空间分成页，程序按模块分段，每个段再分成与物理空间页同样大小的页面。虚存空间和物理空间按页进行交换

回顾：页式虚拟存储器

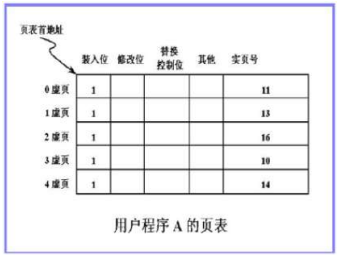
❖页式虚拟存储器

- 虚存空间和主存空间按固定大小分成若干页：虚页vs. 实页
- 虚地址格式（逻辑地址格式）：虚页号 ＋ 页内地址
- 实地址格式（物理地址格式）：实页号 ＋ 页内地址
- 页表：记录虚页与实页的映射关系，实现虚实地址的转换，页表建立在内存中，操作系统为每道程序建立一个页表。页表用虚页号作为索引，页表项包括虚页对应的实页号和有效位
- 页表寄存器：保存页表在内存中的首地址

2.2 页式虚拟存储器

■ 页表

- 每个进程有一个页表，页表项数取决于虚拟地址的结构。
- 页表项包括：实页号、装入位（有效位）、修改位（脏位）等
- 页表在内存中的首地址记录在页表基址寄存器中。



用户程序 A 的页表

■页表空间问题

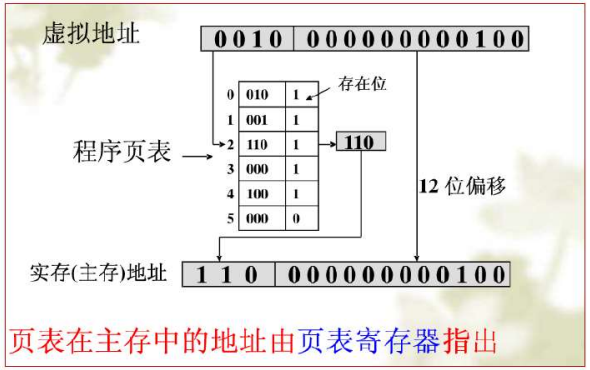
■页表可能很大。如VAX系统中，用户程序（进程）虚拟存储空间最大可达2GB，按512B分页，有2^{22}页，页表可以包括2^{22}个页表项。显然，这么大的页表需要占用很大的内存空间。

■多进程运行，多个页表同时都在内存。多个同时运行的进程的页表空间超过内存可分配空间的可能性是存在的。

■采用多级页表机制：将页表分页，当前使用的页的页表项所在页在内存，其余在外存，页表也采用按页交换机制。

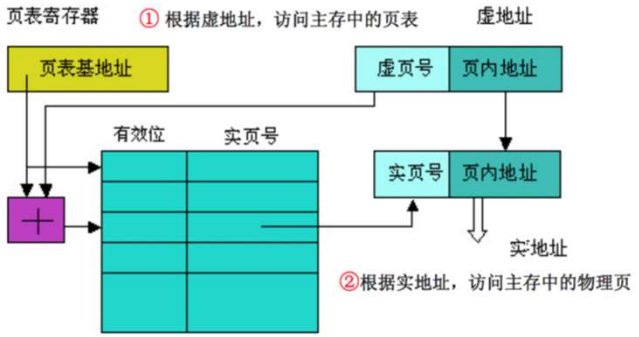
2.2 页式虚拟存储器

■ 虚拟地址到物理地址的转换



2.2 页式虚拟存储器

❖虚实地址的转换 —— 访存次数问题



2.2 页式虚拟存储器

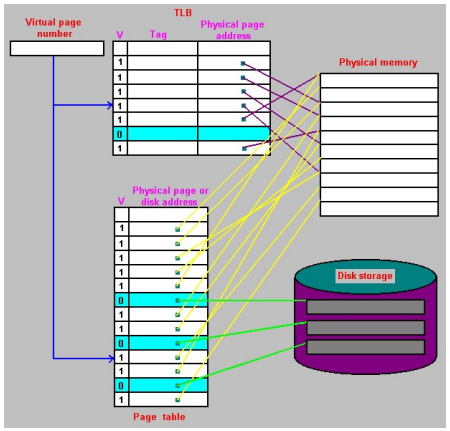
- **快表TLB（Translation Lookaside Buffer，转换后备缓冲器，旁路转换缓冲器）**
  - **问题：每次虚拟存储器的访问带来两次存储器访问，一次访问页表，一次访问所需的数据（或指令），简单的虚拟存储器速度太慢**
  - **解决办法： 使用Cache存储部分活跃的页表项，称为TLB（快表），它包含了最近使用的那些页表项**
  - **TLB内容：虚页号（标记）、对应实页号（数据）、有效位、修改位**
  - **TLB一般采用全相联模式**

有效位	修改位	标记（tag）	数据
		虚页号	实页号
		虚页号	实页号
		虚页号	实页号
		虚页号	实页号

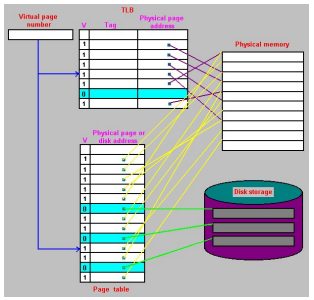
快表（TLB）

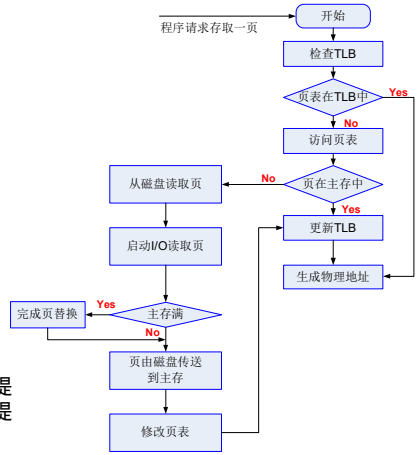
2.2 页式虚拟存储器

❖**页式虚拟存储器工作原理**



2.2 页式虚拟存储器



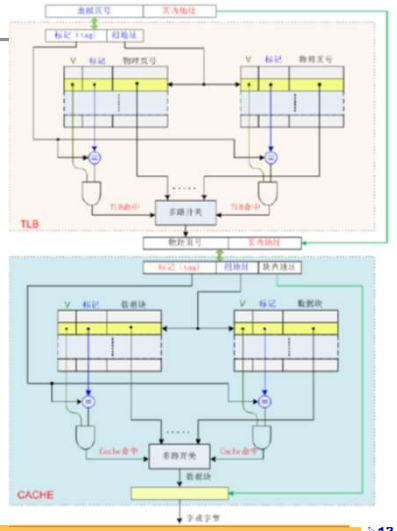


- 当虚页不在内存时(装入位为0)，页表提供保存该虚页的磁盘地址，以便进行磁盘读取。
- 有些系统采用双表结构：页表只提供虚页到实页的映射；由外页表提供虚页到磁盘地址的映射。

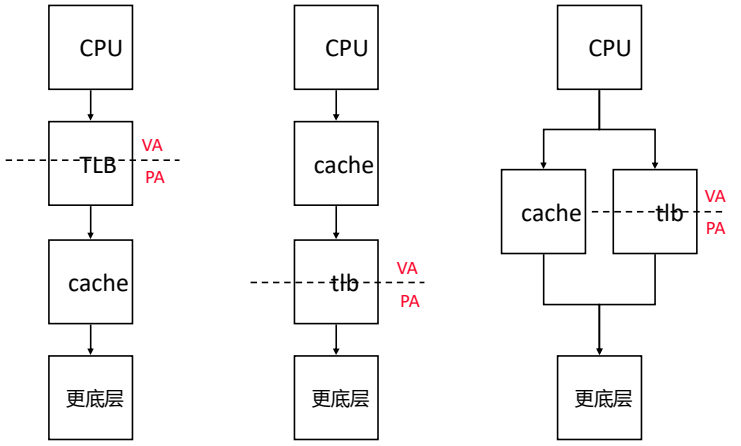
2.2 页式虚拟存储器

❖CPU通过TLB和Cache访问的过程

- TLB和Cache都采用组相联，并假设TLB和Cache均命中
- 依据虚页号，访问TLB，获取实页号，得到物理地址
- 依据物理地址，访问Cache，获取最终数据



2.2 页式虚拟存储器



2.2 页式虚拟存储器

❖TLB，页表，Cache三种缺失的可能性

TLB	Page table	Cache	Possible? If so, under what circumstance?
hit	hit	miss	可能，TLB命中则页表一定命中，但实际上不会查页表
miss	hit	hit	可能，TLB缺失但页表可能命中，信息在主存，就可能在Cache
miss	hit	miss	可能，TLB缺失但页表可能命中，信息在主存，但可能不在Cache
miss	miss	miss	可能，TLB缺失页表可能缺失，信息不在主存，一定也不在Cache
hit	miss	miss	不可能，页表缺失，说明信息不在主存，TLB中一定没有该页表项
hit	miss	hit	同上
miss	miss	hit	不可能，页表缺失，说明信息不在主存，Cache中一定也没有该信息

最好的情况应该是hit、hit、hit，此时，访问主存几次？ 不需要访问主存！

以上组合中，最好的情况是什么？hit、hit、miss和miss、hit、hit 只需访问主存1次

以上组合中，最坏的情况是什么？ miss、miss、miss 需访问磁盘、并访存至少2次

介于最坏和最好之间的是什么？ miss、hit、miss 不需访问磁盘、但访存至少2次

举例

❖假定页式虚拟存储系统按字节编址，逻辑地址36位，页大小16KB，物理地址32位，页表中包括有效位和修改位各1位、使用位和存取方式位各2位，且所有虚拟页都在使用中。请问：

- （1）每个进程的页表大小为多少？
- （2）如果所使用的快表（TLB）总表项数为256项，且采用2路组相联Cache实现，则快表大小至少为多少？

解答（1）

页面大小：16KB=2^{14}，页内偏移14位

虚地址36位：虚页号=36-14=22位

实地址32位：实页号=32-14=18位

每个进程最多可有：2^{22}个虚页

每个页表项：1+1+2 +2+ 18=24位

每个页表所占空间：$2^{22}\times24$=12MB

（2）

TLB：256个表项，2路组相联，所以共有128组

22位虚页号：7位组地址，15位Tag

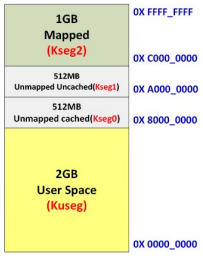
TLB每个表项：15+24=39位

TLB容量：39×256=9984位=1248字节

MIPS的存储空间管理

❖MIPS CPU运行模式：用户态和核心态

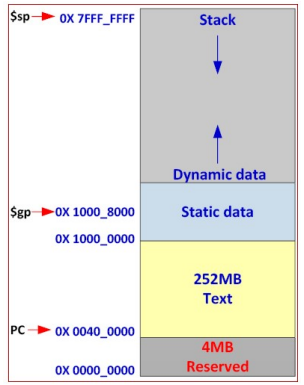
- Kuseg：0000000 -7FFF FFFF (2G)，用户模式下可用，即MIPS规范约定用户空间为2G在带有MMU的机器里，这些地址都将由MMU转换。
- KSeg0：80000000 -9FFFFFFF(512M)，通过将最高位清零映射到物理地址低端512M(00000000 -1FFF FFFF)空间。这种映射简单，无需MMU转换（Unmapped）。这段地址的存取都会通过高速缓存(cached)，对于有MMU 的系统，操作系统内核会存放在该区域。
- KSeg1：A0000000 -BFFFFFFF(512M)，将最高3位清零映射到物理地址低端512M(00000000 -1FFFFFFF)空间，无需MMU转换（Unmapped），kseg1不使用缓存（Uncached）。kseg1是系统重启时能正常工作的内存映射地址空间，重新启动时的入口向量是BFC00000，这个向量对应的物理地址是1FC00000。因此你可以使用这段地址空间来访问你的初始化程序的ROM。
- KSeg2：C0000000 -FFFFFFFF (1G)，只能在核心态下使用，并且要经过MMU转换，一般情况下你不需要使用这段地址空间。



MIPS的存储空间管理

❖MIPS按如下约定为程序分配空间

- 堆栈在高地址区，从高到低增长。
- 过程调用时，生成当前“栈帧”，返回后退回当前栈帧
- 程序的动态数据（如：C中的malloc申请区域、链表）在堆(heap)中从低向高进行存放和释放（free时）栈区位于堆栈高端，堆区位于堆栈低端，静态数据区上方。
- 全局指针$gp固定设为0x10008000，其16位偏移量的访问范围为0x10000000 到0x1000FFFF
- 静态数据区从固定的0x10000000处开始存放
- 程序代码从固定的0x00400000处开始存放，故PC的初始值为0x00400000。



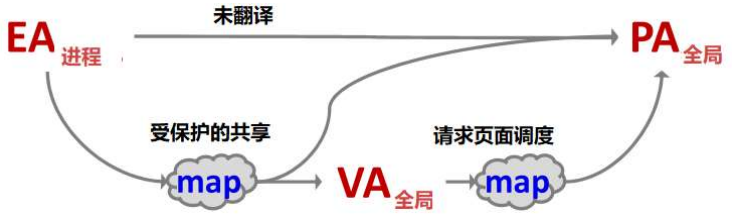
MIPS每个进程的虚拟（逻辑）地址空间

虚拟存储小结

❖虚拟存储的特征

- 程序员在比实际主存空间大得多的逻辑地址空间中编程
- 程序执行时，把当前需要的程序段和相应的数据块调入主存，其他暂不用的部分存放在磁盘上
- 指令执行时，通过硬件（MMU）将逻辑地址（也称虚拟地址或虚地址）转化为物理地址（也称主存地址或实地址）
- 在发生程序或数据访问失效时，操作系统进行主存和磁盘之间的信息交换

“虚存”不代表一切



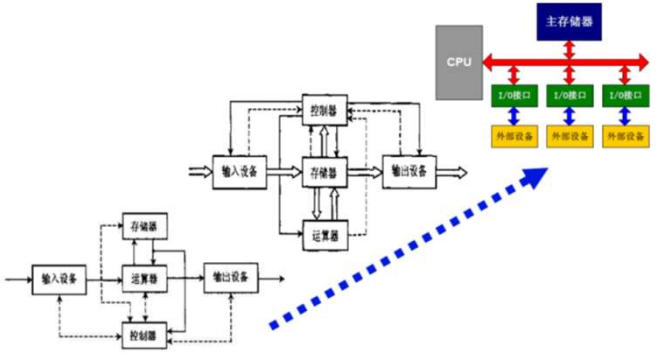
第九部分 总线与I/O

一、总线
二、I/O接口
三、程序查询I/O方式
四、中断与中断I/O方式
五、DMA I/O方式
六、I/O通道

1.1 总线的一般概念

❖总线的由来

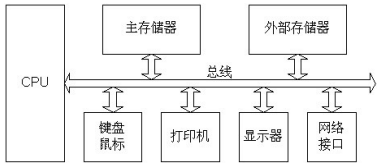
计算机部件的互连方式：分散连接 → 总线连接



1.1 总线的一般概念

❖总线：连接两个或多个功能部件的一组公共的信号传输线

- 连接多个部件的信息传输线，各部件共享的传输介质
- 通过总线，计算机在各部件之间实现地址、数据和控制信息的交换
- 在某一时刻，只允许有一个部件向总线发送消息，而多个部件可同时从总线上接收相同的信息
- 总线实际上是由许多传输线或通路组成，每条线可传输一位二进制码，一串二进制码可以在一段时间内逐一传输，若干条传输线可同时传输若干位二进制代码



1.1 总线的一般概念

❖总线特性

- 机械特性：机械连接方式。如几何尺寸、引脚数量、插头标准。
  - 连接方式：电缆式、主板式、底板式
- 电气特性：信号传输方向、有效电平、电平逻辑等。
  - 电平方式：单端方式（一组信号线、一个公共接地信号）、差分方式
  - 电平逻辑：正逻辑、负逻辑
- 功能特性：信号功能定义。
- 时间特性：信号之间的时序关系。

1.1 总线的一般概念

❖总线的设计要素（性能指标）

- ➢类型：专用或复用
  - 总线复用：地址总线与数据总线是否复用（时分多路复用）
- ➢仲裁方式：集中式或分布式
  - 总线上各部件使用总线的仲裁方式。
- ➢时序：同步/异步方式
  - 总线上的数据与时钟同步工作的总线为同步总线，与时钟异步的总线为异步总线。
- ➢总线宽度：数据总线位数（根数），如32位，64位。
- ➢标准传输率：每秒传输的最大字节量。
- ➢信号线数：所有信号线的总数。

例子

❖设一个32位微处理器配有16位的外部数据总线，时钟频率为50MHz，若最短的总线传输周期为4个时钟周期，问处理器的最大数据传输率是多少？若想提高一倍数据传输率，可采用什么措施？

❖ 该总线的最短传输周期为：

$T = 4/50MHz = 80 \times 10^{-9}$ s

❖ 对于外部总线为16位的处理器，最大数据传输率为：

$2B/T = 25 \times 10^6$ Bps = 25MBps

❖ 若想提高一倍数据传输率，可采用以下两种措施：

(1)外部数据总线宽度改为32bits，则最大数据传输率为：

$4B/T = 50 \times 10^6$ Bps = 50 MBps

(2)时钟频率加倍，则总线传输周期为：$T1 = 4/100MHz = 40 \times 10^{-9}$ s，

最大数据传输率： $2B/T1 = 50 \times 10^6$ Bps = 50 MBps