

Project 8: 虚拟内存分配器

Chentao Wu 吴晨涛

Professor

Dept. of CSE, SJTU

wuct@cs.sjtu.edu.cn

课程目标

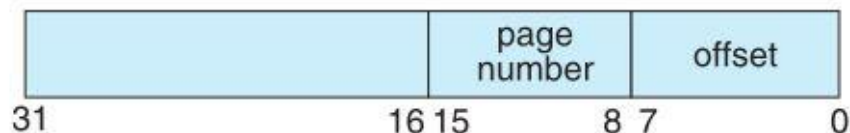
- 设计一个虚拟内存管理器，实现地址翻译，缺页异常处理，页面替换和分析。

项目介绍

- 本项目，你需要设计一个的虚拟内存管理器，将大小为 2^{16} bytes的虚拟地址空间的逻辑地址翻译成物理地址。程序基础功能包含以下五个：
 - （1分）从address.txt文件中读取逻辑地址
 - （3分）使用TLB和页表进行地址翻译得到物理地址
 - （1分）利用物理地址，从物理内存（可以分配一个数组模拟物理内存）中读取对应的字节
 - （2分）发生缺页异常时，从BACKING_STORE.bin文件中将页面加载到物理内存中，更新页表和TLB
 - （2分）统计并报告Page-fault rate和TLB hit rate

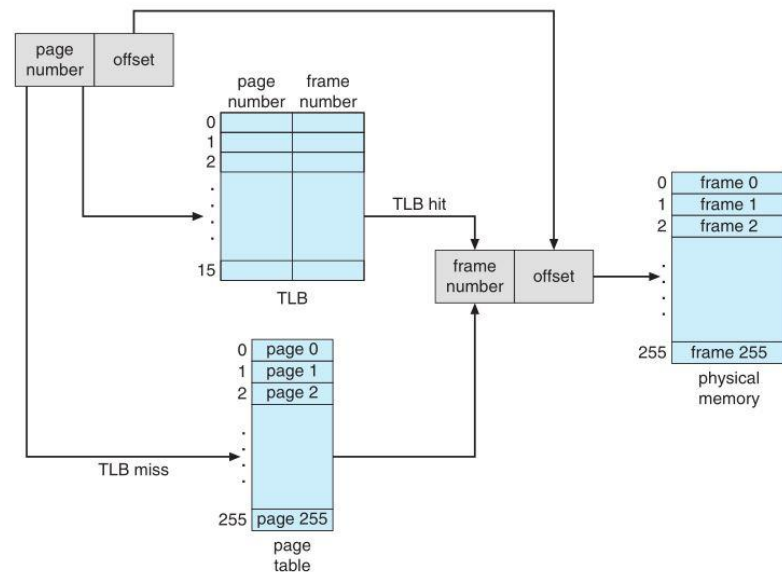
项目详情

- 你的程序将从address.txt文件中读取一些32-bit的虚拟地址。如左图所示，虚拟地址的高16位直接忽视，低16位将分为两部分，包括8-bit的页号和8-bit的偏移量。



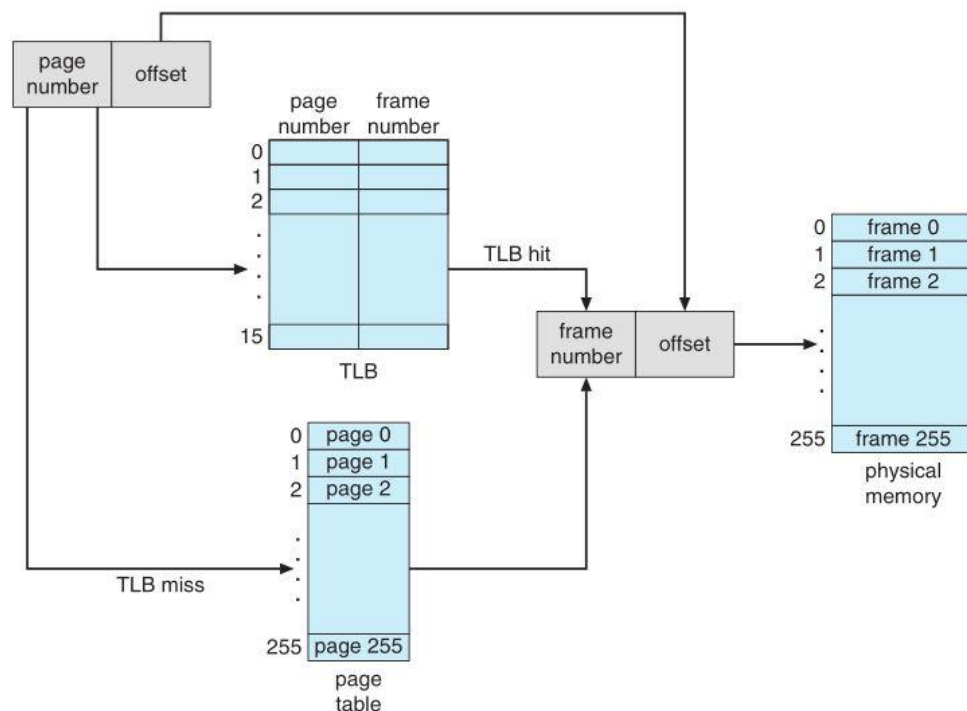
- 其他参数详情

- 一级页表
- 页表中有 2^8 个页表项
- 页大小为 2^8 bytes
- TLB有16个条目
- 页帧大小为 2^8 bytes
- 有256个页帧
- 物理内存大小为65536bytes ($256 \text{页帧数} \times 256 \text{bytes页大小}$)



地址翻译

- 地址翻译过程如右图所示，首先从逻辑地址提取页号，然后从TLB中查询是否命中。如果命中，则从TLB中获取帧号；否则，查询页表以获得帧号。如果页表中没有相应帧号，则触发缺页异常。得到帧号后，利用帧号和偏移量计算出物理地址，从物理内存中获取相应的字节值。



缺页异常处理

- 你的程序对页面实现按需分配，即当需要特定页面时，才将页面从辅助存储器中加载到主存储器。
- 本项目的辅助存储器由文件BACKING_STORE.bin表示，这是一个大小为65536bytes的二进制文件。当出现缺页异常时，你可以从该文件中读取大小为256bytes的页，将其加载到物理内存中。例如，如果页号为15的逻辑地址触发缺页异常，您的程序将从BACKING_STORE.bin中读取page15，并将其存储在物理内存中的页帧中，并更新页表和TLB。
- 为实现从BACKING_STORE.bin文件读取单个页面，你可以使用fopen(), fread(), fseek()和fclose()等标准C库函数来执行I/O操作。

作业及评分

自行阅读课本第十章的 Programming Projects 部分，完成以下任务，完成后共计12分。

- (base 9分) 根据前面的项目介绍，实现5个基础功能。
- (bonus 2分) 进阶功能：原先项目的物理内存和虚拟内存大小一致，现在你需要改变物理内存大小，将物理内存的页帧数从256个改为128个。此时，虚拟地址空间大小是物理地址空间大小的两倍，物理内存不足的情况可能发生。你需要使用FIFO或LRU进行页面替换来解决物理内存不足的情况。（注：淘汰页面后，需要flush TLB以及更新页表，flush TLB可以简单地将TLB所有内容清空）
- (report 1分) 做一个简单的报告解释你的代码，报告要求重点突出，内容精炼。