Lab05：设计虚拟内存管理器

截止日期

请参阅实验室分配要求。

目标

该项目的目标是实践虚拟内存管理的地址转换。

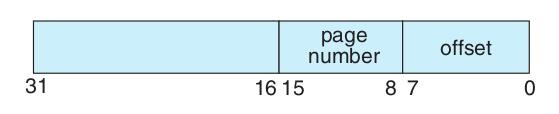
细节

该项目包括编写一个程序，该程序将逻辑地址转换为物理地址，用于大小为2 ^ 16 = 65,536字节的虚拟地址空间。您的程序逻辑将从包含地址的文件中读取，并使用TLB和页表将每个逻辑地址转换为其对应的物理地址，并输出存储在转换后的物理地址中的字节值。

您的学习目标是使用模拟来理解将逻辑地址转换为物理地址所涉及的步骤。这将包括使用请求分页解决页面错误，管理TLB以及实现页面替换算法。

具体

您的程序将读取包含几个表示逻辑地址的32位整数的文件。但是，您只需要关注16位地址，因此必须屏蔽每个逻辑地址的最右边16位。这16位分为（1）8位页码和（2）8位页偏移。因此，地址的结构如下所示：



其他细节包括以下内容：

页面表中有2 ^ 8个条目

页面大小为2 ^ 8字节

TLB中有16个条目

帧大小为2 ^ 8字节

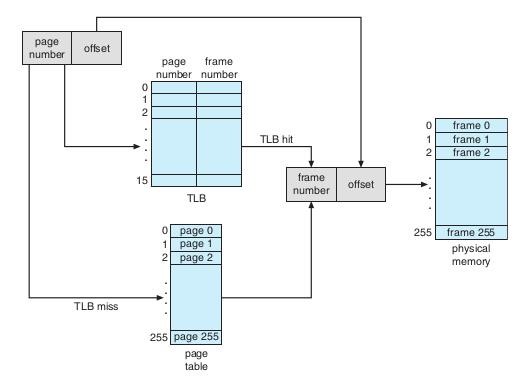
256帧

物理内存为65,536字节（256帧×256字节帧大小）

此外，您的程序只需要关注读取逻辑地址并将其转换为相应的物理地址。您不需要支持写入逻辑地址空间。

地址翻译

您的程序将使用教科书中概述的TLB和页表将逻辑地址转换为物理地址。首先，从逻辑地址中提取页码，并查阅TLB。在TLB命中的情况下，帧号从TLB获得。在TLB未命中的情况下，必须查阅页表。在后一种情况下，从页表获得帧号，或者发生页面错误。地址转换过程的直观表示是：



处理页面错误

您的程序将实现请求分页策略。后备存储由文件BACKING\_STORE.bin表示，该文件是大小为65,536字节的二进制文件。发生页面错误时，您将从文件BACKING\_STORE读取一个256字节的页面，并将其存储在物理内存中的可用页面框架中。例如，如果页码为15的逻辑地址导致页面错误，则程序将在第15页从BACKING\_STORE读取（请记住页面从0开始，大小为256字节）并将其存储在物理内存的页面框架中。一旦存储了该帧（并且更新了页表和TLB），TLB或页表将解析对页面15的后续访问。

您需要将BACKING\_STORE.bin视为随机访问文件，以便您可以随机搜索文件的某些位置以供阅读。我们建议使用标准C库函数来执行I / O，包括fopen（），fread（），fseek（）和fclose（）。

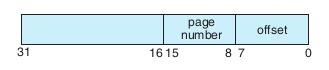
物理内存的大小与虚拟地址空间的大小相同--65,536字节 - 因此您不必在页面错误期间关注页面替换。后来，我们使用较少量的物理内存描述了对该项目的修改;此时，将需要一个页面替换策略。

测试文件

我们提供文件addresses.txt，其中包含表示逻辑地址的整数值，范围从0到65535（虚拟地址空间的大小）。您的程序将打开此文件，读取每个逻辑地址并将其转换为相应的物理地址，并在物理地址处输出有符号字节的值。

怎么开始

首先，编写一个简单的程序，根据以下内容提取页码和偏移量：



来自以下整数：

1,256,32768,32769,128,65534,33153

也许最简单的方法是使用运算符进行位屏蔽和位移。一旦您可以正确地建立页码和偏离整数，您就可以开始了。

最初，我们建议您绕过TLB并仅使用页表。一旦页表正常工作，您就可以集成TLB。请记住，地址转换可以在没有TLB的情况下工作; TLB只是让它变得更快。当您准备好实现TLB时，请记住它只有16个条目，因此在更新完整TLB时需要使用替换策略。

您可以使用FIFO或LRU策略来更新TLB。

如何运行程序

您的程序应运行如下：

./a.out addresses.txt

您的程序将读取文件addresses.txt，其中包含1,000个逻辑地址，范围从0到65535.您的程序是将每个逻辑地址转换为物理地址，并确定存储在正确物理地址的有符号字节的内容。 （回想一下，在C语言中，char数据类型占用一个字节的存储空间，因此我们建议使用char值。）您的程序将输出以下值：

1.正在转换的逻辑地址（从addresses.txt读取的整数值）。

2.相应的物理地址（程序将逻辑地址转换为）。

3.签名的字节值存储在翻译的物理地址的物理存储器中。我们还提供了文件correct.txt，其中包含文件addresses.txt的正确输出值。您应该使用此文件来确定您的程序是否正确地将逻辑转换为物理地址。

统计

完成后，您的程序将报告以下统计信息：

1.页面错误率 - 导致页面错误的地址引用的百分比。

2. TLB命中率 - 在TLB中解析的地址引用的百分比。由于addresses.txt中的逻辑地址是随机生成的，并且不反映任何内存访问位置，因此不要期望具有高TLB命中率。

提示：

您可以尝试生成具有局部性的逻辑地址序列，因此可以预期更高的命中率。

页面替换

到目前为止，该项目假设物理内存与虚拟地址空间大小相同。实际上，物理内存通常比虚拟地址空间小得多。该项目的这一阶段现在假设使用一个较小的物理地址空间，其中包含128个页面帧而不是256个。此更改将需要修改程序，以便跟踪可用页面帧以及使用FIFO或FIFO实现页面替换策略LRU在没有空闲内存时解决页面错误。不要忘记将以前的程序保留为版本1，并将其作为另一个版本启动。

服从

您的提交应包括（1）代码（需要makefile或其等效文件），（2）描述您的设计以及如何编译/使用代码的自述文件，以及（3）以下家庭作业的报告：

您的程序设计

实验结果（统计）的快照与分析

遇到的问题和您的解决方案

总结我们教科书中列出的不同内存管理方法。

参考资料

您的建议和意见

环境

Linux（推荐使用Ubuntu 18.04 / 16.04）和C / C ++，如果您愿意，也可以使用Java代码。

参考

教科书或您认为有用的任何其他文章。