实验 5：设计一个虚拟内存管理器

截止日期

请参考实验室作业要求。

目标

本项目的目标是练习虚拟内存管理的地址转换。

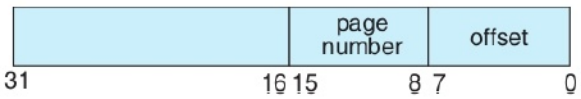
详情

该项目包括编写一个程序，用于将虚拟地址空间大小为 2^16 = 65536 字节的逻辑地址转换为物理地址。您的程序将从一个包含逻辑地址的文件中读取数据，并使用 TLB 和页面表将每个逻辑地址转换为相应的物理地址，然后输出存储在转换后的物理地址处的字节值。

您的学习目标是利用模拟来理解将逻辑地址转换为物理地址所涉及的步骤。这将包括使用按需分页解决页面错误、管理 TLB 以及实现页面置换算法。

具体

你的程序将读取一个包含几个32位整数的文件，这些整数表示逻辑地址。然而，你只需要关注16位地址，所以你必须对每个逻辑地址的最右16位进行掩码处理。这16位被分为（1）一个8位页面号和一个（2）一个8位页面偏移量。因此，地址的结构如下所示：

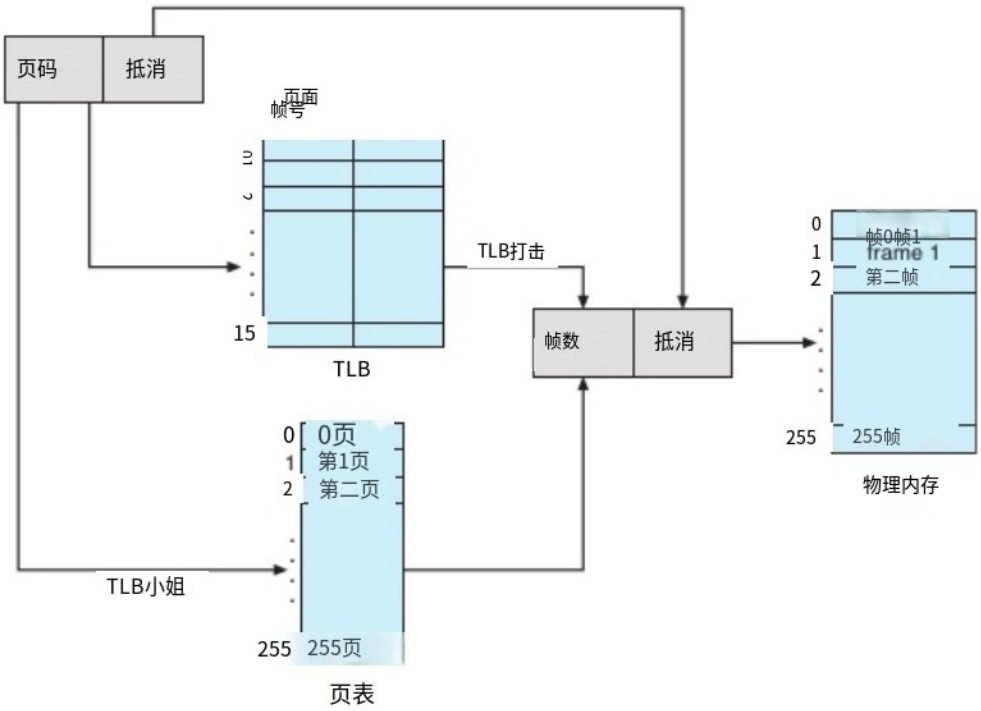


其他具体内容包括以下这些：
页面表中有 2^8 个条目
页面大小为 2^8 字节
☒ 16 个 TLB 中的条目
帧大小为 2^8 字节
256 帧
☒ 物理内存为 65536 字节（256 帧×256 字节帧大小）
此外，您的程序只需关注读取逻辑地址和转换即可。

将它们映射到相应的物理地址。您无需支持向逻辑地址空间写入操作。

地址转换

你的程序将使用 TLB 和页面表将逻辑地址转换为物理地址，如教科书中所述。首先，从逻辑地址中提取页面号，并查阅 TLB。在 TLB 命中时，从 TLB 中获取帧号。在 TLB 未命中时，必须查阅页面表。在后一种情况下，要么从页面表中获取帧号，要么发生页面错误。地址转换过程的可视化表示如下：



处理页面错误

您的程序将实现按需分页策略。后备存储由文件BACKING_STORE.bin表示，这是一个大小为65,536字节的二进制文件。当发生页面错误时，您将从文件BACKING_STORE中读取一个256字节的页面，并将其存储在物理内存中的可用页面帧中。例如，如果一个逻辑地址的页面号为15导致页面错误，您的程序将从BACKING_STORE中读取第15页（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中的页面帧中。一旦这个帧被存储（并且页面表和TLB被更新），后续对第15页的访问将通过TLB或页面表来解决。

您需要将 BACKING_STORE.bin 视为随机访问文件，以便您能够随机跳转到文件的某些位置进行读取。我们建议使用标准的 C 库函数。

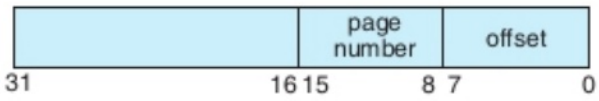
执行输入/输出操作，包括 fopen（） 、fread（） 、fseek（） 和 fclose（） 。
物理内存的大小与虚拟地址空间的大小相同——65536 字节——因此，在发生页面错误时，您无需担心页面替换。稍后，我们将描述对这个项目的修改，使用更少的物理内存；到那时，就需要一种页面替换策略了。

测试文件

我们提供文件 addresses.txt，其中包含从 0 到 65535（虚拟地址空间的大小）的整数值，表示逻辑地址。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，然后输出物理地址处有符号字节的值。

如何开始

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：



从以下整数中：

1、256、32768、32769、128、65534、33153
也许实现这一点最简单的方法是使用位掩码和位移运算符。一旦您能从一个整数中正确确定页面号和偏移量，就可以开始操作了。

最初，我们建议您绕过 TLB 仅使用页面表。一旦您的页面表正常工作，您就可以集成 TLB 了。请记住，地址转换在没有 TLB 的情况下也能工作；TLB 只是让它更快。当您准备实现 TLB 时，请记住它只有 16 个条目，所以当您更新完整的 TLB 时，您需要使用替换策略。您可以使用先进先出（FIFO）或最近最少使用（LRU）策略来更新您的 TLB 。

如何运行您的程序

您的程序应该按如下方式运行：

./a.out 地址.txt

您的程序将读取名为“addresses.txt”的文件，该文件包含 1000 个从 0 到 65535 的逻辑地址。您的程序要将每个逻辑地址转换为物理地址，并确定在正确物理地址存储的有符号字节的内容。（请记住，在 C 语言中，char 数据类型占用一个字节的存储空间，所以我们建议使用 char 类型的值。）
你的程序要输出以下数值：
1. 正在转换的逻辑地址（从 addresses.txt 中读取的整数值）
2. 相应的物理地址（您的程序将逻辑地址转换成的地址）

3. 在转换后的物理地址的物理内存中存储的有符号字节值。
我们还提供了文件“correct.txt”，其中包含了文件“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地址转换为物理地址。

统计数据

完成后，您的程序要报告以下统计数据:
1. 页面错误率 - 导致页面错误的地址引用所占的百分比。
2. TLB 命中率 - 在 TLB 中解决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的局部性，因此不要期望会有很高的 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，这样可能会有望获得稍高一些的命中率。

页面替换

到目前为止，该项目假设物理内存的大小与虚拟地址空间相同。但实际上，物理内存通常比虚拟地址空间小得多。在这个项目的这个阶段，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。这种变化将需要修改您的程序，以便它跟踪空闲页面帧，并实现一种页面替换策略，使用FIFO或LRU来解决当没有空闲内存时的页面错误。别忘了将您之前的程序作为版本1保留，并开始这个项目作为另一个版本。

提交

您的提交内容应包括：（1）代码（需要有一个 makefile 或其等效文件），（2）一个描述您的设计以及如何编译/使用您的代码的自述文件，以及（3）关于以下家庭作业的报告:
☒ 您对该程序的设计
实验结果（统计数据）的截图及分析
☒ 遇到的问题及您的解决方案
总结我们课本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

推荐使用 Linux 系统（Ubuntu 20.04/18.04 版本）以及 C/C++ 代码，如果您愿意，也可以使用 Java 代码。

参考文献

教科书或者任何你觉得有用的其他文章。