

## DIY地址翻译

本练习让您尝试分层分页中的地址转换。为了使事情易于管理,我们必须使用一个很小的逻辑和物理地址空间,比我们在课堂上学习的例子大一点。

这个例子将让我们看到一些与真正的分页内存系统相同的事情。

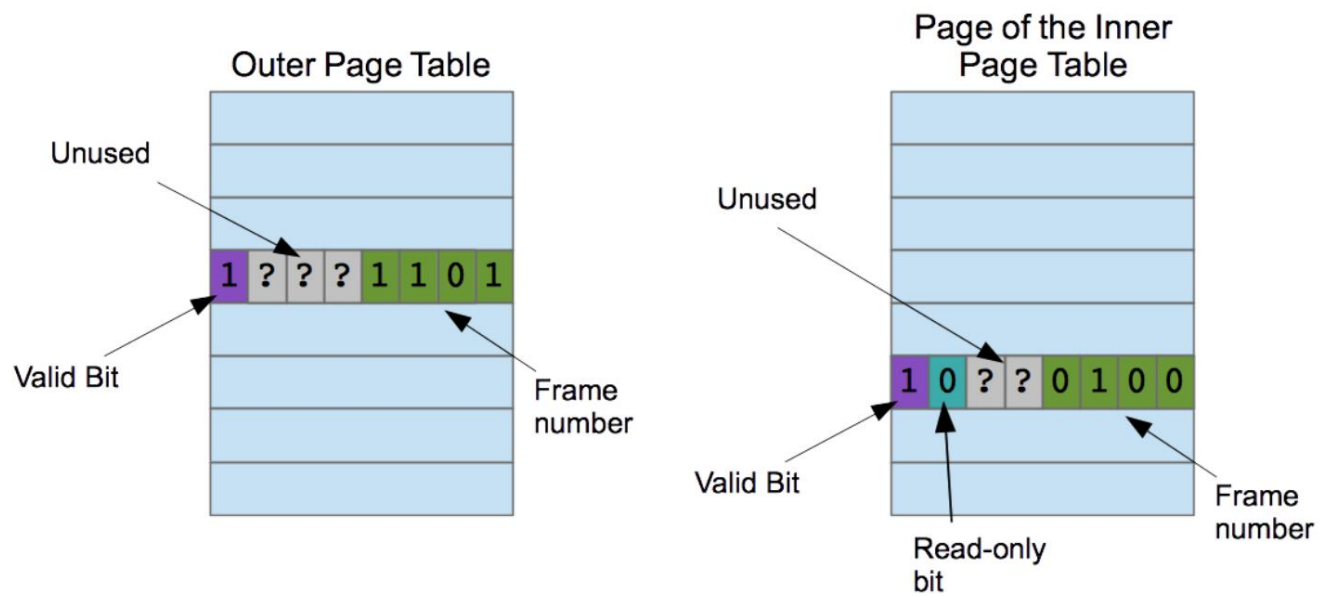
在 Moodle 的课程主页上,您会找到一个文件, memory.txt。该文件的一部分如下所示。这个文件给出了物理内存的内容,每行一个字节。物理地址是 7 位,因此内存的每个字节都显示 (以二进制形式),其 7 位物理地址位于左侧 (也以二进制形式给出)。内存被分成每帧 8 字节的帧。

框架十中的顶级页表。

实际地址	内存框内容编号
(二进制)	(二进制)
-----	
0000000	01111000 帧 0000 (0) 01100000
0000001	
0000010	11101010
0000011	00010011
0000100	01101101
0000101	01000011
0000110	10100010
0000111	10010100
-----	
0001000	00111100 帧 0001 (1) 10011000
0001001	
0001010	00111111
0001011	10100010
0001100	00100001
0001101	10110111
0001110	10101110
0001111	10111101
-----	
0010000	00011001 帧 0010 (2) 11001110
0010001	
0010010	01010100
0010011	01010101
0010100	11011011
0010101	01110010
0010110	00111010
0010111	11111010
-----	
0011000	10000111 帧 0011 (3) 11000011
0011001	
0011010	10011000
0011011	01010111
0011100	11101010
0011101	10000011
0011110	11000110
0011111	01100010

在这台计算机上,假设一个逻辑地址是 9 位。该系统使用两级寻呼。外页表填充一帧,每个条目存储在一个字节中 (因此,外页表有 8 个条目)。内页表的每一页占据一帧。内页表的条目只有一个字节,因此内页表的每一页也包含 8 个条目。

下图说明了外页表和内页表的格式。对于外表,该图显示了条目 3 可能包含的内容。高位为 1 表示该条目有效,进程可以访问通过该条目到达的页面。接下来的三位没有用到,低四位给出了内页表对应页所在的帧。例如,这里的内页表的第 3 页是有效的,它位于第 13 帧 (十进制为 1101)。



内页表的一页具有如右图所示的格式。每个条目都存储在一个字节中。条目的高位是有效位,下一位是只读位 (1表示只读,0表示可写)。接下来的两位未使用,低四位给出进程对应页所在的内存帧。假设这是页表第 3 页的插图。然后,该图中显示的行将告诉我们进程的第 29 页 (页表的第一页告诉我们进程中的第 0 .. 7 页,页表的下一页告诉我们第 8 .. 15,然后是下一个 16 .. 23,因此此页面告诉我们有关流程的第 24 .. 31 页)。根据该图,进程的第 29 页是有效的,它不是只读的,它位于第 4 帧 (0100 的十进制)。

您将通过完成名为练习 12 测验的 Moodle 在线测验来完成此练习。文件 memory.txt 显示您将使用的内存内容。在这个文件中,外页表在第 10 帧中。

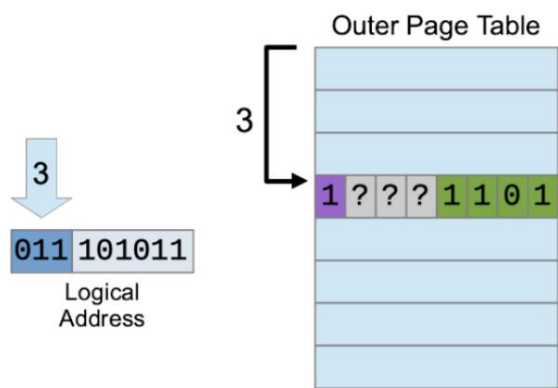
你的工作是报告在内存引用集合的地址转换过程中会发生什么。每个内存引用都是读取或写入,后跟一个 9 位逻辑地址。例如,给定以下内存引用:

· 写 011101011

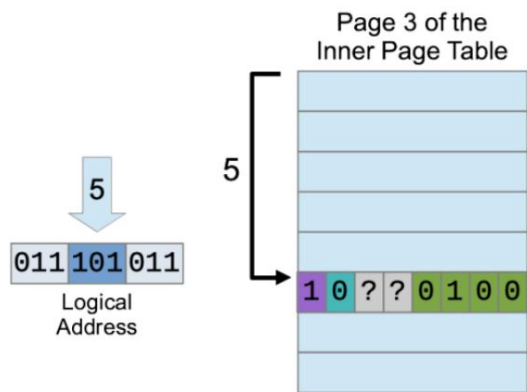
您会报告这转换为物理地址 0100011。

为什么这个地址会翻译成 0100011?请记住,在分层分页中,您会将逻辑地址分成多个字段。在这种架构中,页表被分成八个 (页大小的)部分,因此外页表包含八个条目。地址中的高三位告诉您需要页表的哪一部分。

下图显示了外部页表的一部分。对于逻辑地址 011101011,高三位的值为 3。外页表的第 3 行表示页表的这一部分是有效的,它在内存的第 13 帧中。



下面显示了 (内部)页表的第 3 页的一部分,来自内存的第 13 帧。它有 8 个条目,这是有道理的,因为页码中还剩下 3 位。逻辑地址的下三位表示我们需要页表这一部分的条目 5。查看条目 3,我们看到进程的这个页面是有效的,它不是只读的。它在内存的第 4 帧中。



使用帧号 (0100)作为高位,逻辑地址 (011)的剩余3位作为偏移量,我们得到物理地址0100011。

您必须翻译的某些给定内存访问可能不允许。下面两个例子都不好。一种是尝试访问无效页面,另一种是尝试写入只读页面。对于此类情况,您只需回答 X。

· 读取 111000111 · 写  
入 101111010