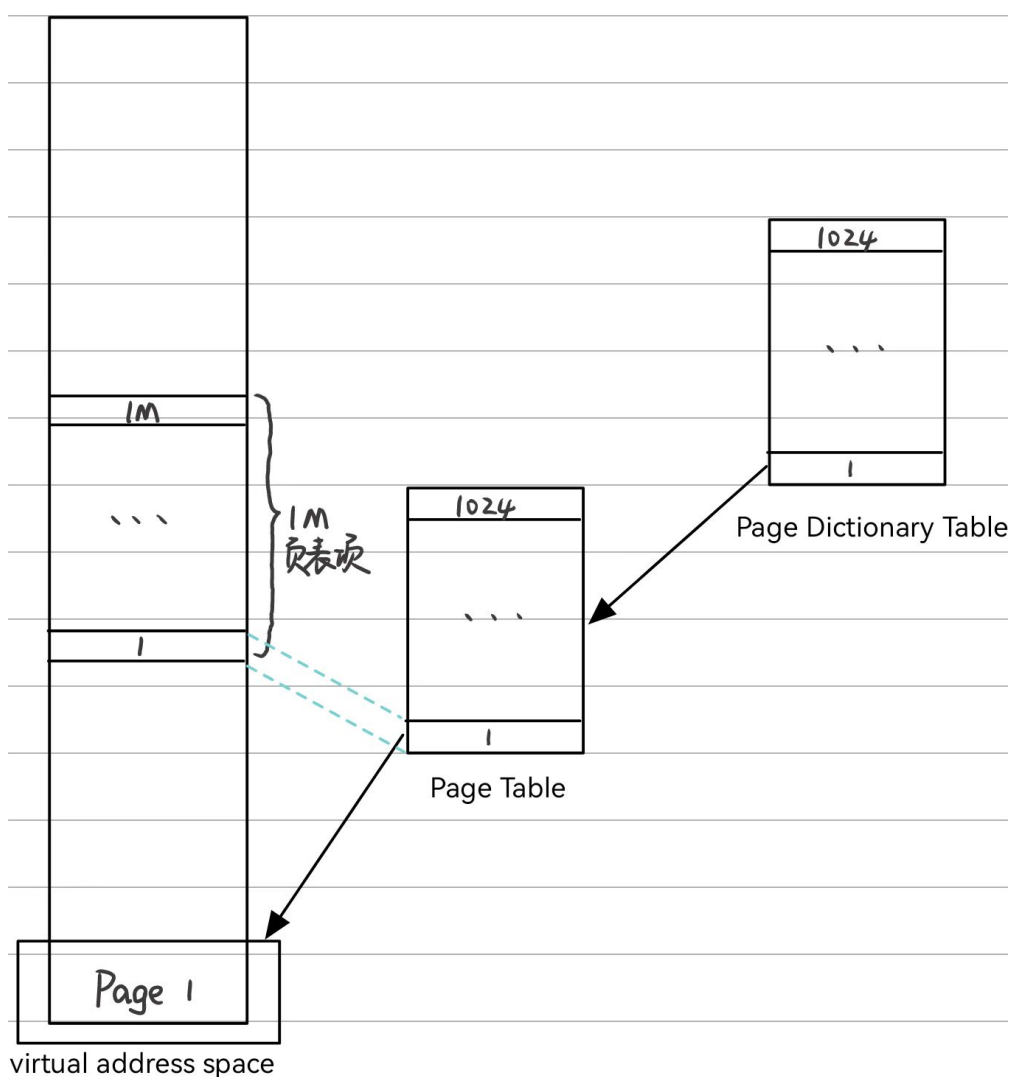


## 前提

- 在**39位**的三级页式存储系统中，虚拟空间大小为**512GB**，一个页表的大小是**4KB**，一个页表项是**8B**
- 给出的**PTbase**为虚拟存储空间的虚拟地址，只有这样页表才会在内存(虚拟)中连续排布。
- 页目录指的是一级页表。
- 多级页表的映射是按照顺序进行的，即内存中划分的第一个页块由上一级页表的第一个页表项映射，而这个页表项所在的页表是由上一个页表的第一个页表项映射，依次类推。

以二级页表为例



其中第一个**Page Table**的第一个页表项即为连续排布的页表项的第一个。

## 分析

## 二级页表

首先我们以一个二级页表引入，假设一个页表的大小是**4KB**，一个页表项的大小是**4B**

在虚拟地址空间中，这些页表项是连续排布的，我们暂且不谈这些页表项组成了多少页，我们现在知道的是，第**n**个页表项的映射的是虚拟地址空间中的第**n**个页。

首先我们以二级页表开始，给出了**PTbase**，其实也就是虚拟内存中第一个页表项的地址。这个页表项是处于某一个页表中的，可以计算得到页号为：

$$\text{PN}(\text{Page Number}) = \text{PTbase} \gg 12$$

由上文可知，这个页号对应的页是由第**PN**个页表项映射得到的，即第一个页表项所在的页表由第**PN**个页表项映射得到，由页表的映射可知，第**PN**个页表项是页目录的第一个页表项，因此，这个页目录第一个页表项的地址就是页目录的基地址：

$$\text{PDbase} = \text{PTbase} + \text{PN} * 4 = \text{PTbase} + \text{PTbase} \gg 10$$

那指向页目录的那一页表项的地址是什么呢？只需重复之前的动作，算出页目录所在的页号，得到映射到此页的页表项的项号，通过**PTbase**加上偏移量，即可得到映射到页目录自身的页目录项的地址：

$$(\text{PDbase} \gg 12) * 4 + \text{PTbase} = \text{PTbase} + \text{PTbase} \gg 10 + \text{PTbase} \gg 20$$

## 三级页表

和二级页表一样，我们现在知道的是，第**n**个页表项的映射的是虚拟地址空间中的第**n**个页。

虚拟地址空间可以划分为：**239/212=227** 页，即我们需要**227**个页表项进行对这些页的映射。

首先我们以三级页表开始，给出了**PTbase**，其实也就是虚拟内存中第一个页表项的地址。这个页表项是处于某一个页表中的，可以计算得到页号为：

$$\text{PN}(\text{Page Number}) = \text{PTbase} \gg 12$$

由上文可知，这个页号对应的页是由第**PN**个页表项映射得到的，即第一个页表项所在的页表由第**PN**个页表项映射得到，由页表的映射可知，第**PN**个页表项是第一个二级页表的第一个页表项，因此，这个二级页表第一个页表项的地址就是该二级页表的基地址：

页中间目录基地址(**page middle directory**)即二级页表的基地址**PMDbase**：

$$\text{PMDbase} = \text{PTbase} + \text{PN} * 8 = \text{PTbase} + \text{PTbase} \gg 9$$

这个二级页表是由一级页表即页全局目录项（**page global directory entry**）**PGDE**的第一个页表项映射得到的，现在我们只需得到**PMD**所在的页表号，即可得到一级页表的第一页表项的项号，由此我们可以得到一级页表的基地址：

$$\mathbf{PGDEbase = (PMDbase \gg 12) * 8 + PTbase = PTbase + PTbase \gg 18 + PTbase \gg 9}$$

重复上面的步骤，我们可以算出一级页表所处的页表号，由此得到映射到该页表的页表项项号，通过**PTbase**加上偏移量，即可得到映射到页目录自身的页目录项的地址：

$$\mathbf{(PGDEbase \gg 12) * 8 + PTbase = PTbase + PTbase \gg 27 + PTbase \gg 18 + PTbase \gg 9}$$

## 多级页表

根据以上的例子，我们就可以推广到多级页表中，只需要从最低一级的页表层层向上推进，就可以得到最终结果。其实看起来是有规律的。