

本节内容

两级页表

知识总览

单级页表存在什么问题？如何解决？

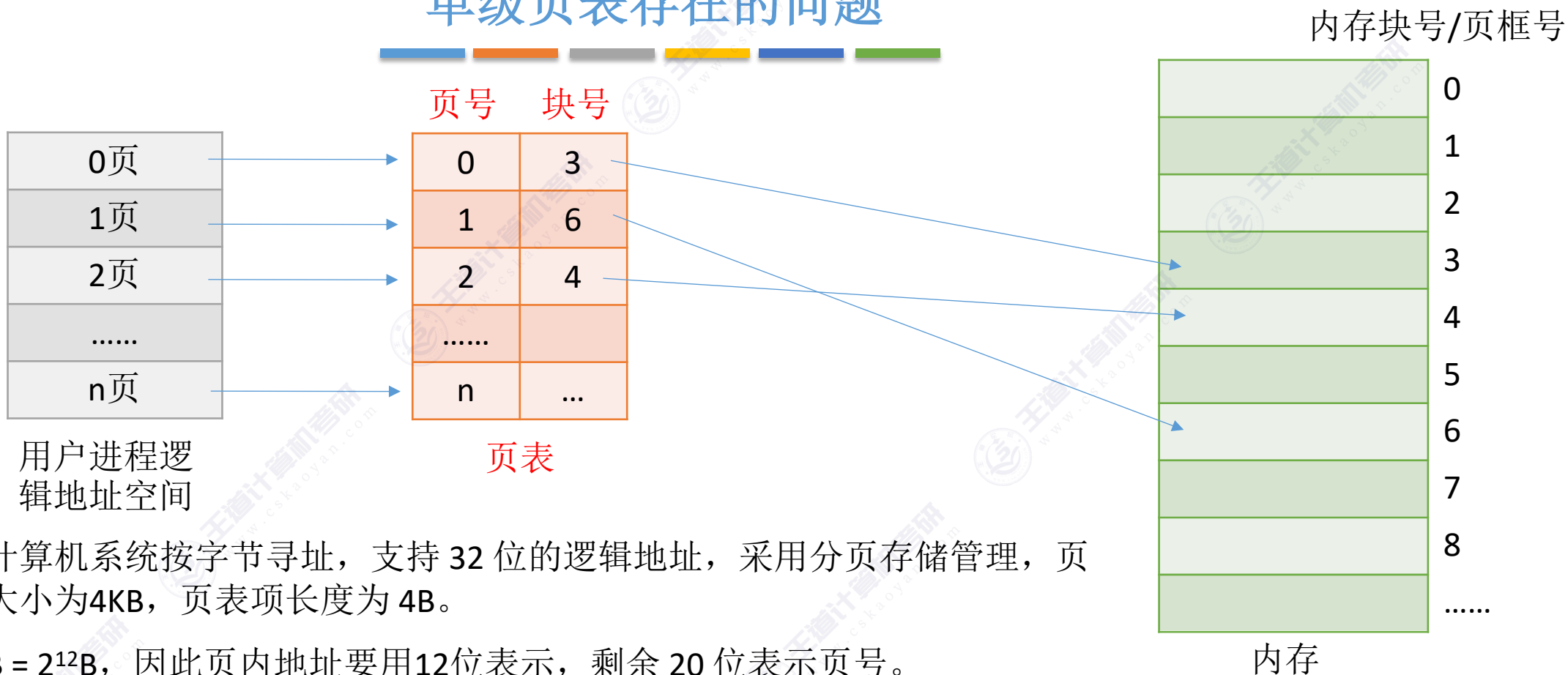
两级页表的原理、逻辑地址结构

如何实现地址变换？

两级页表问题需要注意的几个细节

两级页表

单级页表存在的问题



某计算机系统按字节寻址，支持 32 位的逻辑地址，采用分页存储管理，页面大小为 4KB，页表项长度为 4B。

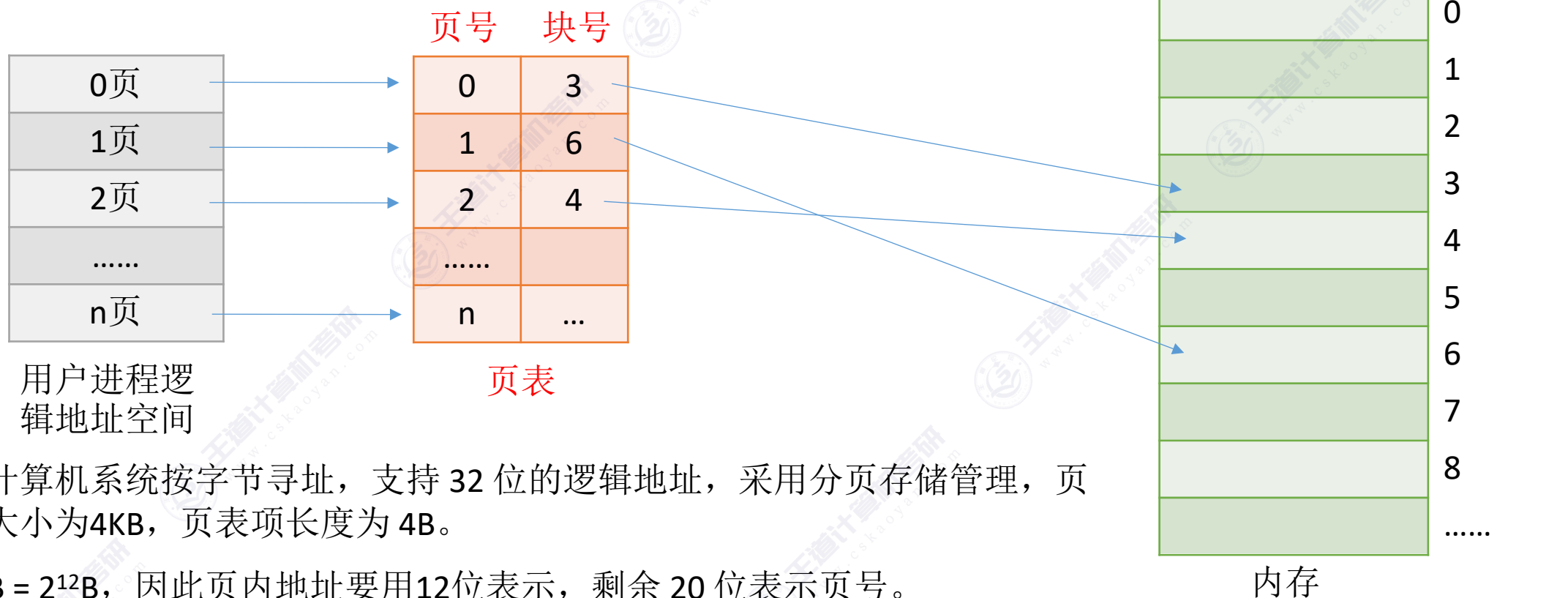
4KB = 2^{12} B，因此页内地址要用 12 位表示，剩余 20 位表示页号。

因此，该系统中用户进程最多有 2^{20} 页。相应的，一个进程的页表中，最多会有 $2^{20} = 1\text{M} = 1,048,576$ 个页表项，所以一个页表最大需要 $2^{20} * 4\text{B} = 2^{22}$ B，共需要 $2^{22}/2^{12} = 2^{10}$ 个页框存储该页表。

根据页号查询页表的方法：K 号页对应的页表项存放位置 = 页表始址 + K * 4
要在所有的页表项都连续存放的基础上才能用这种方法找到页表项

需要专门给进程分配 $2^{10} = 1024$ 个连续的页框来存放它的页表

单级页表存在的问题



某计算机系统按字节寻址，支持 32 位的逻辑地址，采用分页存储管理，页面大小为4KB，页表项长度为 4B。

4KB = 2^{12} B，因此页内地址要用12位表示，剩余 20 位表示页号。

因此，该系统中用户进程最多有 2^{20} 页。相应的，一个进程的页表中，最多会有 $2^{20} = 1\text{M} = 1,048,576$ 个页表项，所以一个页表最大需要 $2^{20} * 4\text{B} = 2^{22}$ B，共需要 $2^{22}/2^{12} = 2^{10}$ 个页框存储该页表。

根据局部性原理可知，很多时候，进程在一段时间内只需要访问某几个页面就可以正常运行了。因此没有必要让整个页表都常驻内存。

如何解决单级页表的问题？

问题一：页表必须连续存放，因此当页表很大时，需要占用很多个连续的页框。

问题二：没有必要让整个页表常驻内存，因为进程在一段时间内可能只需要访问某几个特定的页面。



把页表再分页并离散存储，然后再建立一张页表记录页表各个部分的存放位置，称为**页目录表**，或称**外层页表**，或称**顶层页表**

两级页表的原理、地址结构

32位逻辑地址空间，页表项大小为4B，页面大小为4KB，则页内地址占12位

31	12	11	0
页号			页内偏移量		

单级页表结构的逻辑地址结构

0# 页表

0	2
1	4
.....	
1023	...

1# 页表

0	762
.....	
1023	...

...

1023# 页表

0	...
.....	
1023	...

分为
1024
个部分

页号

块号

0	2
1	4
...	
1024	762
...	
1048575	...

进程最多有 2^{20} 个页面，
用 20 位二进制刚好可以
表示 $0 \sim 2^{20}-1$ 个页号。
每个页面可存放 $4K/4 = 1K = 2^{10} = 1024$ 个页表项。

内存块号/页框号

	0
	1
	2
	3
	4
	...
	762
	...

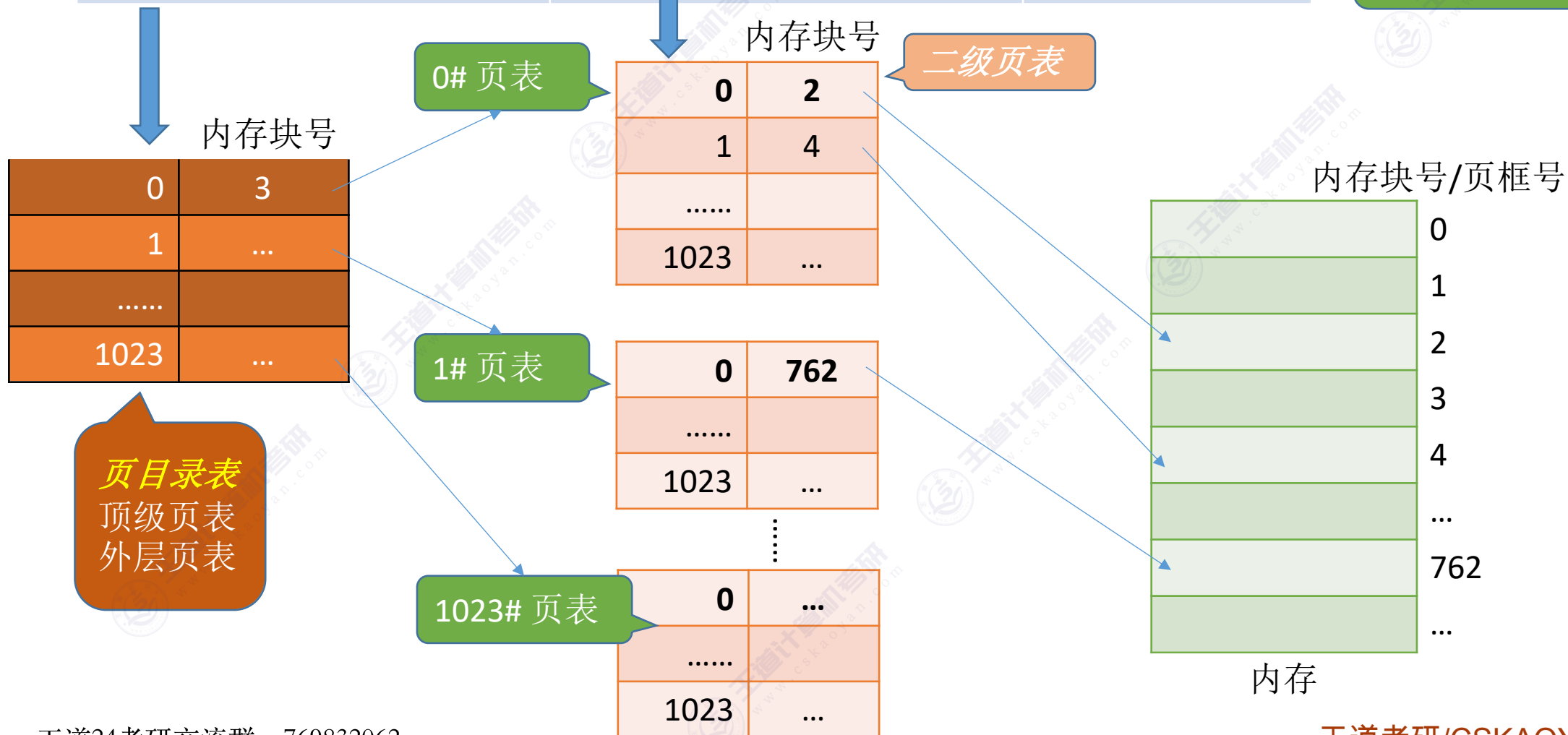
内存

10位一级页号刚好可表示 0~1023

两级页表的原理、地址结构

10位	10位	12位
一级页号（页目录号）	二级页号（页表索引）	页内偏移量

两级页表结构的逻辑地址结构



如何实现地址变换

31	22	21	12	11	0
一级页号（页目录号）			二级页号			页内偏移量		

两级页表结构的逻辑地址结构

例：将逻辑地址 (0000000000,0000000001,1111111111) 转换为物理地址（用十进制表示）。

内存块号	
0	3
1	...
.....	
1023	...

内存块号	
0	2
1	4
.....	
1023	...

内存块号/页框号	
	0
	1
	2
0# 页表存放位置	3
	4
	...
	762
	...
内存	

最终要访问的内存块号为4

该内存块的起始地址为 $4 * 4096 = 16384$

页内偏移量为 4095

最终的物理地址为 $16384 + 4095 = 20479$

- ①按照地址结构将逻辑地址拆分成三部分
- ②从PCB中读出页目录表始址，再根据一级页号查页目录表，找到下一级页表在内存中的存放位置
- ③根据二级页号查二级页表，找到最终想访问的内存块号
- ④结合页内偏移量得到物理地址

页表项存放位置：
 $3 * 4096 + 1 * 4 = 12,292$

如何解决单级页表的问题？

问题一：页表必须连续存放，因此当页表很大时，需要占用很多个连续的页框。

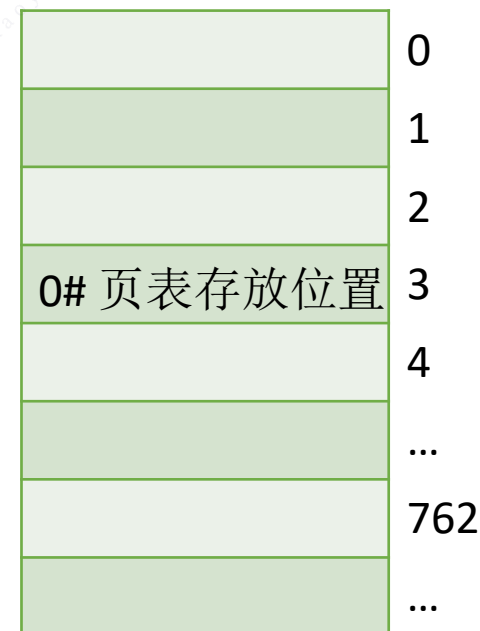
问题二：没有必要让整个页表常驻内存，因为进程在一段时间内可能只需要访问某几个特定的页面。

可以在需要访问页面时才把页面调入内存（虚拟存储技术）。可以在页表项中增加一个标志位，用于表示该页面是否已经调入内存

一级页号	内存块号	是否在内存中
0	3	是
1	无	否
.....		
1023	...	

二级页号	内存块号	是否在内存中
0	2	是
1	4	是
.....		
1023	...	

若想访问的页面不在内存中，则产生缺页中断（内中断/异常），然后将目标页面从外存调入内存



需要注意的几个细节

1. 若分为两级页表后，页表依然很长，则可以采用更多级页表，一般来说**各级页表的大小不能超过一个页面**

例：某系统按字节编址，采用 40 位逻辑地址，页面大小为 4KB，页表项大小为 4B，假设采用纯页式存储，则要采用（）级页表，页内偏移量为（）位？

页面大小 = 4KB = 2^{12} B，按字节编址，因此页内偏移量为12位

页号 = 40 - 12 = 28 位

页面大小 = 2^{12} B，页表项大小 = 4B，则每个页面可存放 $2^{12} / 4 = 2^{10}$ 个页表项

因此各级页表最多包含 2^{10} 个页表项，需要 10 位二进制位才能映射到 2^{10} 个页表项，因此每一级的页表对应页号应为10位。总共28位的页号至少要分为三级

逻辑地址：

页号 28位

页内偏移量 12位

逻辑地址：

一级页号 8位

二级页号 10位

三级页号 10位

页内偏移量 12位

2. 两级页表的**访存次数**分析（假设没有快表机构）

第一次访存：访问内存中的页目录表

第二次访存：访问内存中的二级页表

第三次访存：访问目标内存单元

如果只分为两级页表，则一级页号占 18 位，也就是说页目录表中最多可能有 2^{18} 个页表项，显然，一个页面是放不下这么多页表项的。

知识回顾与重要考点

所有页表项必须连续存放，页表过大时需要很大的连续空间

单级页表存在的问题

在一段时间内并非所有页面都用得到，因此没必要让整个页表常驻内存

将长长的页表再分页

两级页表

! 逻辑地址结构：（一级页号，二级页号，页内偏移量）

! 注意几个术语：页目录表/外层页表/顶级页表

要能根据逻辑地址位数、页面大小、页表项大小 确定多级页表的逻辑地址结构

两级页表

如何实现地址变换

①按照地址结构将逻辑地址拆分成三部分

②从PCB 中读出页目录表始址，根据一级页号查页目录表，找到下一级页表在内存中的存放位置

③根据二级页号查表，找到最终想访问的内存块号

④结合页内偏移量得到物理地址

几个细节

! 多级页表中，各级页表的大小不能超过一个页面。若两级页表不够，可以分更多级

! 多级页表的访存次数（假设没有快表机构）—— N 级页表访问一个逻辑地址需要 $N+1$ 次访存



公众号：王道在线



b站：王道计算机教育



抖音：王道计算机考研