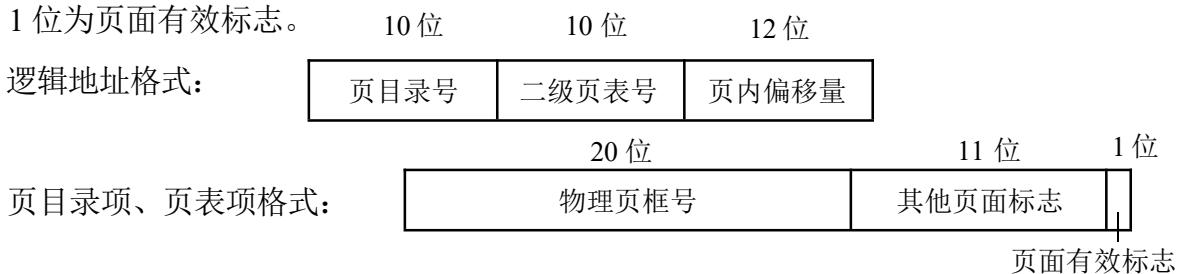


1. 一个 32 位的页式虚拟存储系统采用两级页表管理，其逻辑地址中，第 22 到 31 位是第一级页表（页目录）索引，第 12 位到 21 位是第二级页表索引，页内偏移占第 0 到 11 位。每个页表（目录）项包含 20 位物理页框号和 12 位标志位，其中最后

1 位为页面有效标志。

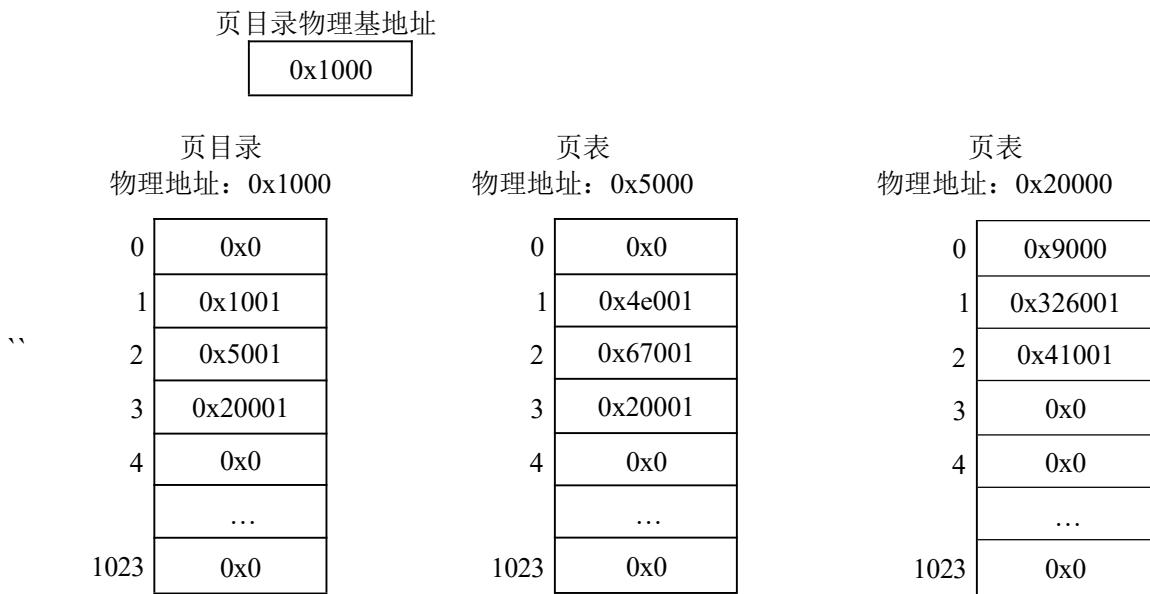


(1) 请问进程整个的地址空间有多少字节？一页有多少字节？

(2) 一个进程如果从 0x80000000 开始映射 4MB 大小页表空间，求第一级页表（页目录）的起始逻辑地址，并指出从哪个逻辑地址可以读出第一级页表（页目录）所在的物理页框号。说明理由。（注意 B 代表字节，一个 32 位地址占 4 字节）

(3) 如果当前进程的页目录物理地址、页目录和相应页表内容如图下所示，请描述访问以下逻辑地址时系统进行地址转换的过程，如可行，给出最终访存读取到的数据。逻辑地址：0x0、0x00803004、0x00402001

(4) 要想访问物理地址 0x326028，需要使用哪个逻辑地址？



- (1) 共有 $2^{32}B = 4GB$, 一页有 $2^{12} = 4KB$
- (2) 页目录基地址: $0x80000000 | (0x80000000 >> 10) = 0x80200000$, 即
`PDbase = 0x80200000`
- 页目录项: $PDbase|PTbase >> 10 = 0x80200800$, 即通过逻辑地址
`0x80200800` 即可读到页目录的物理页框号
- (3)
- 对于 `0x00000000`, `PDX = 0`, `PTX = 0`, 访问页目录的第 0 项得到二级页表物理地址为 `0x00000000`, 没有该物理地址的页表, 产生缺页中断, 需要申请内存建立页表
 - 对于 `0x00803004`, `PDX = 0x2`, `PTX = 0x3`, 访问页目录的第 2 项得到 `0x5001`, 页表有效, 位置在 `0x5000`, 访问第 3 项得到 `0x20001`, 物理页框有效, 为 `0x20000`, 故恰好访问到了另一个页表, 访问第 1 项, 为 `0x326001`, 访问第 `0x0` 个字节, 为 `0x00`
 - 对于 `0x00402001`, `PDX = 0x1`, `PTX = 0x2`, `offset = 0x1`, 访问页目录第 1 项, 得到页表有效, 为自身, 访问第 2 项, 得到页表 `0x5000` 处, 访问第 0 项, 得到 `0x0`, 访问第 1 个字节, 为 `0x00`
- (4) 物理页框号为 `0x326`, 该物理页面存在于页表 `0x20000` 处, 该页表又存在于页目录 `0x1000` 处的第 3 项, 故需要访问 `0x00c01028`

2. 一个 32 位的虚拟存储系统采用两级页表管理, 其逻辑地址形式如下:

第一级页表 (10 位)	第二级页表 (10 位)	页内偏移 (12 位)
--------------	--------------	-------------

物理地址为 32 位, 形式为:

物理页框号 (20 位)	页内偏移 (12 位)
--------------	-------------

页表项 (PTE) 格式为:

高 20 位	低 12 位 (为标志位)
物理页框号	第 0 位为有效位, 如果为 0, 表示 Invalid; 如果为 1 表示 Valid。 第 1 位为读写位, 如果为 0, 表示 Read Only ; 如果为 1 表示 Read/Write。

假设当前进程第一级页表的物理地址为 `0x0020 0000`, 利用后面物理内存的信息, 请在下表中写出以下指令的执行结果。

对于 Load 指令，如果成功执行，写出读入的数据（读取一个字节），否则写Error；
对于 Store 指令，如果成功执行，写OK，否则写 Error。并给出寻址过程。

指令	结果
Load [0x00001022]	20
Store [0x00C07222]	Error
Store[0x00C005BF]	OK
Load [0x00003013]	19
Load [0xFF80078F]	Error
Load [0xFFFFF005]	66

执行过程：

1. 执行结果: **20**

过程: PDX为0x000, PTX为0x001, 页内偏移为0x022, 页目录项为00100007, 物理页号0x00100, 页表项为00004067, 页表和页表项均有有效, 物理页号为0x00004, 读到20

2. 执行结果: **Error**

过程: PDX为0x003, PTX为0x007, offset为0x222, 页目录项为00103007, 有效, 且物理页号为0x00103, 页表项为EEFF0001, 有效, 但读写位为0, 不可写入, 返回Error

3. 执行结果: **OK**

过程: PDX为0x003, PTX为0x000, offset为0x5BF, 页目录项为00103007, 有效, 且物理页号为0x00103, 页表项为11220067, 有效, 且读写位为1, 可写入, 物理页号为0x11220, 可写入物理地址0x112205BF, 写入内容, 返回OK

4. 执行结果: **19**

过程: PDX为0x000, PTX为0x003, offset为0x013, 页表物理页号为0x00100, 数据物理页号为, 0x00004, 由offset为0x013, 可得数据为19

5. 执行结果: **Error**

过程: PDX为0x3FE, PTX为0x000, offset为0x78F, 页目录项为001FE007, 有效, 物理页号为0x001FE, 页表项为04150000, 无效, 返回Error

6. 执行结果: **66**

过程: PDX为0x3FF, PTX为0x3FF, offset为0x005, 页目录项为001FF007, 有效, 物理页号为0x001FF, 页表项为00103067, 有效, 物理页号为0x00103, 读到66

物理内存（大尾端）