

1. 动态内存分配需要对内存分区进行管理，一般使用位图和空闲链表两种方法。128MB 的内存以  $n$  字节为单元分配，对于链表，假设内存中数据段和空闲区交替排列，长度均为 64KB。并假设链表中的每个节点需要记录 32 位的内存地址信息、16 位长度信息和 16 位下一节点域信息。这两种方法分别需要多少字节的存储空间？哪种方法更好？

位图： $(16/n)$  MB，链表 8KB。

从内存角度来说，只有当  $n > 2048$  时，其占用空间大小会小于链表。但从访问效率来看，位图的访问效率较高，链表需要从头遍历访问。

2. 在一个交换系统中，按内存地址排列的空闲区大小是：10KB、4KB、20KB、18KB、7KB、9KB、12KB 和 15KB。对于连续的段请求：12KB、10KB、9KB。使用 FirstFit、BestFit、WorstFit 和 NextFit 将找出哪些空闲区？

FirstFit: 20KB, 10KB, 18KB

BestFit: 12KB, 10KB, 9KB

WorstFit: 20KB, 18KB, 15KB

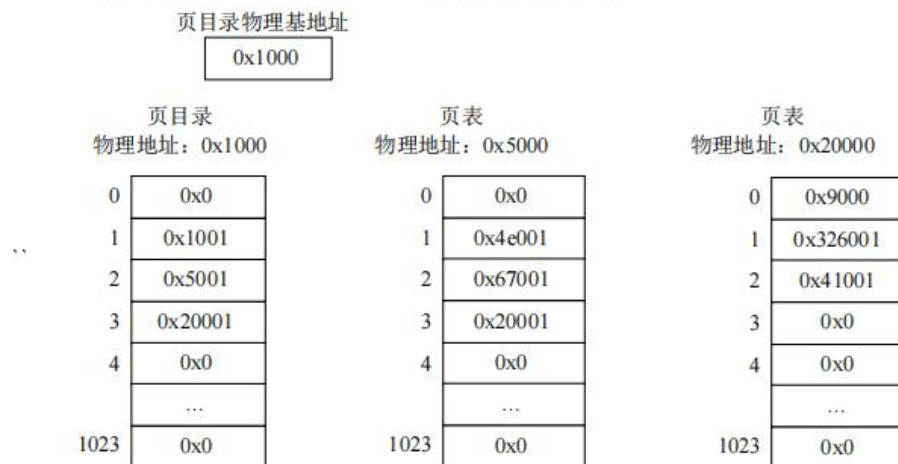
NextFit: 20KB, 18KB, 9KB

3. 解释逻辑地址、物理地址、地址映射，并举例说明。

逻辑地址是用户/程序运行时所用的地址，物理地址是物理内存中真正的地址，地址映射指的一般是指从逻辑地址到物理地址的转换。

例子如下：

- (2) 如果当前进程的页目录物理基地址、页目录和相应页表内容如图下所示，请描述访问以下虚拟地址时系统进行地址转换的过程，如可行给出最终访存获取到的数据。虚拟地址：0x0、0x00803004、0x00402001
- (3) 要想访问物理地址 0x326028，需要使用哪个虚拟地址？



③ VA: 0x00803004, PDX(VA): 0x2, PTX(VA): 0x3

对应页目录项有效，物理框号为 0x5000

对应页表项有效，物理框号为 0x20000

页内偏移为 0x004

故最终数据物理地址为 0x20000004

若为大端存储，取出 0x0，若为小端，取出 0x1

4. 解释页式（段式）存储管理中为什么要设置页（段）表和快表，简述页式（段式）地址转换过程。

以页式内存管理为例，页式存储管理的目的是高效利用内存，并实现大逻辑空间映射小物理空间。要使用页式内存管理，就需要存储一张页表，用于记录逻辑页向物理页的映射。由于查页表会增加一次访问内存，且访问内存的时间相较于 CPU 的工作时间过长，所以设计了快表，即页表的 cache，在查找页面映射时先查快表，若命中则无需查页表，若没有命中再访问页表，并做快表替换。

页式地址转换过程：

先从 CR3 寄存器中获取页目录的基地址，再以线性地址的 Directory 位段为下标，在页目录中取得相应页表的基地址，以线性地址中的 Table 位段为下标，在所得到的页表中获得相应的页表项，将页表项中给出的页面基地址与线性地址中的 offset 位段相加得到物理地址。

5. 叙述缺页中断的处理流程。

(1) 陷入内核态，保存 OS 及用户进程状态相关的必要信息。

(2) 查找出来发生页面中断的虚拟页面（进程地址空间中的页面）。这个虚拟页面的信息通常会保存在一个硬件寄存器中，如果没有的话，操作系统必须检索程序计数器，取出这条指令，用软件分析该指令，通过分析找出发生页面中断的虚拟页面。

(3) 检查虚拟地址的有效性及安全保护位。如果发生保护错误，则杀死该进程。

(4) 查找一个空闲的页框，如果没有空闲页框则需要通过页面置换算法找到一个需要换出的页框。

(5) 如果找的页框中的内容被修改了，则需要将修改的内容保存到磁盘上。

(6) 页框回写完毕后，操作系统将保存在磁盘上的页面内容复制到该页框中。

(7) 当磁盘中的页面内容全部装入页框后，向操作系统发送一个中断。操作系统更新内存中的页表项，将虚拟页面映射的页框号更新为写入的页框，并将页框标记为正常状态。

(8) 恢复缺页中断发生前的状态，将程序指针重新指向引起缺页中断的指令。

(9) 程序重新执行引发缺页中断的指令，进行存储访问。

6. 假设一个机器有 38 位的虚拟地址和 32 位的物理地址。

(1) 与一级页表相比，多级页表的主要优点是什么？

(2) 如果使用二级页表，页面大小为 16KB，每个页表项有 4 个字节。应该为虚拟地址中的第一级和第二级页表域各分配多少位？

(1) 若逻辑地址空间较大，划分的页比较多，若采用单级页表，页表自身占用的连续存储空间过大，实现较为困难，查询也较为耗时。多级页表能够实现页表的动态替换和调入，只需要将当前要用的页表项调入内存，节省了存储页表的空间。

(2) 12,12。

7. 假设页面的访问存在一定的周期性循环，但周期之间会随机出现一些页面的访问。例如：0,1,2,...,511,431,0,1,2,...,511,332,0,1,2,...,511 等。请思考：

(1) LRU、FIFO 和 Clock 算法的效果如何？

(2) 如果有 500 个页框，能否设计一个优于 LRU、FIFO 和 Clock 的算法？

(1) 如果页框号大于 512，三种算法效果相同，都不会产生缺页中断。若小于 512，由于循环访问的特点，LRU,FIFO,Clock 算法的效果几乎相同。最早装入的页面和最久不用的页面等价。

(2) 先进后出 FILO 算法，最后被装进的最先被调出。循环中前 400 个页面几乎每次命中，

夹在循环中的随机页同样大概率命中。

8. 一个交换系统通过紧缩技术来清理碎片。如果内存碎片和数据区域是随机分配的。而且假设读写 32 位内存字需要 10nsec. 那么如果紧缩 128MB 的内存需要多久？简单起见，假设第 0 个字是碎片的一部分而最高位的字包含了有效的数据。

$10 \cdot 2^{25}$  ns。