

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?

- 对于位图, 需要给每一个单元分配一位来记录该分配单元是否空闲:  
共需  $(128M / n)$  字节来存储空闲信息
- 对于空闲链表:  
内存地址信息: 4 字节  
长度信息: 2 字节  
下一节点域信息: 2 字节  
因此每个结点: 8 字节  
占用空间:  $(128M / 64K * 8B) = 16kB$
- 当  $n > 4096$  时位图占用空间更少, 反之链表占用空间更少, 但是位图没有容错能力: 如果一个分配单元对应的标志位为 1, 不能确定是否因错误变成 1。当  $n$  比较小的时候, 链表更好, 因为这时候不但占用空间小, 而且还有容错能力。

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?

下表展示请求所对应的空闲区

	10KB	4KB	20KB	18KB	7KB	9KB	12KB	15KB
FirstFit	10KB		12KB	9KB				
BestFit	10KB					9KB	12KB	
WorstFit			12KB	10KB				9KB
NextFit			12KB	10KB		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: 虚拟地址, 程序使用的地址, 需要经过转化
- 物理地址: 物理内存中数据使用的地址
- 地址映射: 地址映射是指将逻辑地址映射到物理地址的过程。这个过程通常由操作系统的内存管理单元 (MMU) 完成。
- 例子: 一个计算机的物理内存大小 1G, 逻辑地址空间大小 4G, 页大小 4kB  
物理地址: 实页号 (18 位) + offset (12 位)  
逻辑地址: 虚页号 (20 位) + offset (12 位)  
页表项有一个 0x00B00 -> 0x00ABC, 现有逻辑地址 0x00B00ABC  
则其物理地址为 0x00ABCABC

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。

- 设置快表是因为页 (段) 表机制使内存访问效率的严重下降, 增加了访存时间。
- 转换过程 (页表): CPU 产生逻辑地址的页号, 首先在快表中寻找, 若命中就找出其对应的物理块; 若未命中, 再到页表中找其对应的物理块, 并将相应的页表项复制到快表。若快表中内容满, 则按某种算法淘汰某些页
- 转换过程 (段表):  
1. 将逻辑地址中的段号  $S$  与段表长度  $TL$  进行比较。若  $S > TL$ , 表示访问越界, 产生越界中断。若未越界, 则根据段表的始址和该段的段号, 计算出该段对应段表项的位置, 从中读出该段在内存的始址。

2. 再检查段内地址  $d$ ，是否超过该段的段长  $SL$ 。若超过，即  $d > SL$ ，同样发出越界中断信号。若未越界，则将该段的基址与段内地址  $d$  相加，即可得到要访问的内存物理地址。
5. 叙述缺页中断的处理流程。  
当进程执行中需访问的页面不在物理内存中时，会引发发生缺页中断，进行所需页面换入：
  1. (现场保护) 陷入内核态，保存必要的信息 (OS 及用户进程状态相关的信息)。
  2. (页面定位) 查找出发生缺页中断的虚拟页面 (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. (权限检查) 检查虚拟地址的有效性 & 安全保护位。如果发生保护错误，则杀死该进程。
  4. (新页面调入) 查找一个空闲的页框 (物理内存中的页面)，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. (\*旧页面写回) 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. (\*新页面调入) 页框“干净”后，操作系统将保持在磁盘上的页面内容复制到该页框中。
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？  
节省内存空间
  - (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？  
页表项： $16KB/4B = 4K \Rightarrow 12$  位页号  
虚拟地址：页表页面号 (12 位) + 页号 (12 位) + offset (14 位)
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？  
这个循环以 513 为周期，最后一次随机访问一个  
LRU：若页框大小小于 512，则在第一个周期之后的每一次访问 (0~511)，都会未命中，最后一次随机访问有可能未命中 (如果页框大小为  $n$ ，并且最后访问的随机数  $x$  在  $(511 - n, 511]$  才会命中)  
FIFO：同上  
Clock：同上  
如果页框大小大于 512，则效果都一样，命中率都为 100%。
  - (2) 如果有 500 个页框，能否设计一个优于 LRU、FIFO 和 Clock 的算法？  
设计一种 FILO 算法，即 First in Last out，先进后出  
这样在周期的开始，0~499 会存入，之后每次只会修改最后一项，0~498 一直存在。  
这样每个周期会产生 12 次缺页，最后一次随机访问如果为 0~498 或 511 则会命中，命中率约为 97.7%。
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。

128MB 的内存共 $128MB/4B = 32M$ 字  
共需 $32M * 10ns = 335,544,320ns = 0.336s$