

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

空间容量 $128MB = 2^{27} B$ ，因为以 n 字节为单元进行分配，故数据总位数 $\frac{2^{27}}{n}$ ，共计 $\frac{2^{27}}{8 \times n} = \frac{2^{24}}{n}$ 个字节

- 链表

空闲块数 $\frac{128MB}{64KB \times 2} = \frac{2^{27}}{2^{16}} = 2^{11}$ ，每个块占用 64 位即 $8B$ ，链表占用空间为 $2^{14} B$

因此，当单元分配的 $n > 1KB$ 时，位图更好； $n < 1KB$ 时，链表更好

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？

	12KB	10KB	9KB
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：程序运行时看到的地址
- 物理地址：内存中真实存在的，标志一块物理内存空间的地址。信息在内存中按照字节存储，为了能够正确获取信息，对每个字节都有一个唯一的物理地址与之对应
- 地址映射：MMU机制，能够负责将逻辑地址转换为物理地址，使得物理地址对于用户不可见
- 举例：对于一个C语言程序，请求空间时，内存会分配一块空间，该程序可以完整访问所有用户可见的内存空间，对于使用的逻辑地址，由MMU转化为逻辑地址，再传入CPU进行实际地址的执行

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。

- 页表：为了便于再内存找到进程的每个页面对应块，在分页系统中对于每个进程配置了一张页表，进程逻辑地址空间中的每一页，都在页表中对应一个页表项
- 段表：保存段号以及相应段的基址，以便进行地址转换，查找进程段
- 快表：为了减少访问内存的次数，因为快表访问时间比访存速度更快，提升查找速度
- 页式地址转换过程：CPU从页表寄存器中读入页表起始地址以及页表长度，之后由访存逻辑地址确定页号以及页内偏移，由页号在页表中查找物理页号，与页内偏移结合得到逻辑地址
- 段式地址转换过程：根据段号找到段表项，找到对应的起始地址，与段内偏移结合得到物理地址

5. 叙述缺页中断的处理流程。

- 陷入内核态
- 保护现场，保存必要的信息供恢复使用
- 查找发生页面中断的虚拟页面。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，用软件分析指令，通过分析找出发生页面中断的虚拟页面
- 查找一个空闲的物理页框，如果没有空闲页框则需要通过页面置换算法好到一个需要换出的页框
- 如果找到的页框中内容被修改了，则需要将修改的内容保存在磁盘上
- 操作系统将保持在磁盘上的新的页面内容赋值到该页框中
- 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
- 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
- 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？

- 多级页表通过离散化存储页表，对页表也实现一个缓存，能够节约空间，且页表分配方式较灵活，可以实现按需分配

(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？

- 页内偏移 $16KB = 2^{14}B$ ，因此页内偏移需要 14 位。由于页表项大小为 2^2B ，则一个页面总共有 2^{12} 个页表项

因此给一级页表域以及二级页表域均分配 12 位，页内偏移 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？

- 由题目可以知道需要周期性装填 512 个页框
如果页框数大于 512 时，则不会出现缺页异常
如果小于 512，采用三种算法效果都比较低，满页后每次访问页面都护出现缺页

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

- 考虑设置某些可用于频繁交换的页框，其他页框固定不变。如固定前499页，只替换编号为499的页，则每一轮最多只有500 ~ 511的页面出现缺页，大部分情况不会出现缺页异常

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。

- 读写一个字节需要 $2.5ns$
紧缩 128MB 需要读写 $2^{27} \times 2 \times 2.5ns = 0.671s$