

作业2

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？

第一种方法需要 $(\frac{2^{24}}{n})$ 个字节，第二种方法需要 (2^{13}) 个字节。如果n较大，则位图法内存开销更小，位图法更好；如果n较小，则空闲链表更好。

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？

FirstFit: 段请求12KB，分配20KB分区；段请求10KB，分配10KB分区；段请求9KB，分配18KB分区。

BestFit: 段请求12KB，分配12KB分区；段请求10KB，分配10KB分区；段请求9KB，分配9KB分区。

WorstFit: 段请求12KB，分配20KB；段请求10KB，分配18KB；段请求9KB，分配15KB。

NextFit: 段请求12KB，分配20KB；段请求10KB，分配18KB；段请求9KB，分配9KB。

3. 解释逻辑地址、物理地址、地址映射，并举例说明。

逻辑地址是虚拟地址，是程序中使用的地址；物理地址指的是主存中储存信息的物理单元的编号；地址映射是逻辑地址转换为物理地址的过程。

举例：C语言中int x; &x即为变量x的逻辑地址，x在内存中实际所处的位置就是为物理地址，地址映射就是通过x的逻辑地址找到x的物理地址的过程。

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。

设置页（段）表记录逻辑地址和物理地址的映射关系，用于地址映射，设置快表可以加快地址转换的速度。

页式地址转换过程：程序生成的逻辑地址包括页号和页内地址，将页号与页表长度进行比较，如果页号大于或等于页表长度，则表示本次所访问的地址已超越进程的地址空间，产生地址越界中断。通过页表将页面号映射到物理地址的页框号，将框号和页内地址组合得到物理地址。若页表项不在TLB中（TLB未命中），则需要从主存中获取页表项并加载到TLB中。最终得到物理地址，程序访问对应的物理内存。

段式地址转换过程：程序生成的逻辑地址包括段号和段内偏移。系统将逻辑地址中的段号S与段表长度TL进行比较。若S>TL，表示段号太大，是访问越界，于是产生越界中断信号。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。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。

当进程执行过程中需访问的页面不在物理存储器中时，会引发发生缺页中断，进行所需页面换入，步骤如下：1. 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. 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(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. 页框处理完成后，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) 占用内存空间小。

(2) 各分配12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？(2) 如果有500个页框，能否设计一个优于LRU、FIFO和Clock的算法？

(1) LRU算法效果较好，因为LRU算法会选择最近最少被使用的页面进行替换，每个周期都会访问一遍循环的页面集合，较长时间没有使用的页面会被替换出去。

FIFO算法效果不好，因为即使经常访问的页面也会因为进入内存时间较早而被替换。

Clock算法是对FIFO算法的一种改进，它会考虑页面最近是否被访问过，所以Clock算法效果会好于FIFO算法。

(2) 选择访问频率最低的页面替换出去。

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。

紧缩128MB内存需要0.32sec,