

第二次作业

1.

如果使用空闲链表法：需要 $2^{(27-16)}=2^{11}$ 个链表项，每个链表项需要 $(32+16+16)=64$ 位=8字节，共需 $2^{14}=16$ KB的存储空间。

如果使用位图法：需要 $2^{27} / n / 8$ 个字节。

两种方法各有优劣。位图法空间成本固定，时间成本低，但是没有容错能力；链表表示法时间成本较高，但是具有一定的容错能力。

2.

	12KB	10KB	9KB
FirstFit	20KB	10KB	18KB
BestFit	12KB	10KB	9KB
WorstFit	20KB	18KB	15KB
NextFit	20KB	18KB	9KB

3.

逻辑地址：逻辑地址是程序或操作系统为内存中的数据或指令分配的地址。它是一个抽象的概念，用于简化程序员的工作，使他们无需关心物理内存的实际布局。程序在运行时，CPU 产生的地址通常是逻辑地址。

物理地址：物理地址是实际内存中数据或指令的实际位置。物理地址反映了内存的物理布局，直接与硬件相关。通常情况下，程序员不需要直接处理物理地址，因为操作系统和内存管理单元会负责将逻辑地址转换为物理地址。

地址映射：地址映射是将逻辑地址转换为物理地址的过程。这个过程由内存管理单元（MMU）负责完成。MMU 通常使用分页或分段等方法来实现地址映射。通过地址映射，操作系统可以实现虚拟内存、内存保护和内存共享等功能。

举例说明：在我们MOS使用的MIPS R3000中，0x8004 0000是一个处于 KSEG0 的虚拟地址，将其最高位置零，得到0x0004 0000的物理地址，这个过程就是一个简单的地址映射。

4.

因为需要维护从虚拟地址到物理地址（线性地址）的映射关系。而快表是为了加速页（段）表的查询过程

对于段式存储管理，首先使用相应段寄存器的内容；查找相应的地址段描述结构；例如在Linux中是LDT或GDT；然后从该描述结构中找到段基址；然后检查偏移量是否超过段限长；然后根据指令性质和段描述符中的访问权限来确定是否越权；最后将指令中发出的地址作为位移，与基地址相加而得出线性地址。

对于页式存储管理，首先将逻辑地址或是线性地址分成多段；先查询TLB，如果TLB命中则可直接得到物理页框号，与页内偏移合并可得到物理地址；如果TLB发生Miss，则引发缺页异常，交由内核填充 TLB。填充完毕后，原程序恢复执行，成功通过TLB完成地址转换。

5.

首先陷入内核态，保存必要的信息；接着获取到导致缺页虚拟页面；然后检查虚拟地址的有效性即权限，无效则杀死进程；再然后找到一个空闲的物理页框，或清出一个物理页面，如果页面被修改，则需要写回，同时清除相应TLB表项；然后在这个新获取的物理页中写入需要的数据；再然后更新页表；然后回复现场，最后让引发缺页中断的程序继续执行。

6.

(1) 只有一级页表需要占用连续的大量存储空间，而多级页表更为灵活，可以不要求页表占用连续的物理空间，并便于实现页表的按需调入。

(2) 由于页面大小为16KB，每个页表项有4个字节，故一页中有4K个页表项，二级页表域分配12位，剩余12位为以及页表域。

7.

(1) 仅考虑拥有的页框数小于512的情况。这三种算法表现都很差，在本题的案例中，由于周期性循环访问最近最久没访问几乎等同于最早装入，这三种算法在替换时总是会选择将不久之后要用到的页面换出，进而均有着接近100%的缺页率。

(2) 设计如下算法：500个页面中前n（例如480）个页面在装入后不再换出，剩余的页面采用先进先出方法。

8.

依题，每4字节需要10ns， $2^{27} / 4 * 10 * 2 = 671.08\text{ms}$