

30. PDP-1是最早的分时计算机之一，有4K个18位字的内存。在每个时刻它在内存中保持一个进程。当调度程序决定运行另一个进程时，将内存中的进程写到一个换页磁鼓上，磁鼓的表面有4K个18位字。磁鼓可以从任何字开始读写，而不仅仅是字0。请解释为什么要选这个磁鼓？
31. 一台计算机为每个进程提供65 536字节的地址空间，这个地址空间被划分为4096字节的页面。一个特定的程序有327 68字节的正文、16 386字节的数据和15 870字节的堆栈。这个程序能装入这个地址空间吗？如果页面大小是512字节，能放得下吗？记住一个页面不能同时包含两个不同段的成分。
32. 一个页面同一时刻可能在工作集中吗？请解释原因。
33. 人们已经观察到在两次缺页中断之间执行的指令数与分配给程序的页框数直接成比例。如果可用内存加倍，缺页中断间的平均间隔也加倍。假设一条普通指令需要1 μ s，但是如果发生了缺页中断就需要2001 μ s（即2ms处理缺页中断）。如果一个程序运行了60s，期间发生了15 000次缺页中断，如果可用内存是原来的两倍，那么这个程序运行需要多少时间？
34. Frugal计算机公司的一组操作系统设计人员正在考虑在他们的新操作系统中减少对后备存储数量的需求。老板建议根本不要把程序正文保存在交换区中，而是在需要的时候直接从二进制文件中调页进来。在什么条件下（如果有这样的条件话）这种想法适用于程序文本？在什么条件下（如果有这样的条件话）这种想法适用于数据？
35. 有一条机器语言指令将要被调入，该指令可把一个32位字装入含有32位字地址的寄存器。这个指令可能引起的最大缺页中断次数是多少？
36. 像在MULTICS中那样，当同时使用分段和分页时，首先必须查找段描述符，然后是页描述符。TLB也是这样按两级查找的方式工作的吗？
37. 一个程序中有两个段，段0中为指令，段1中为读/写数据。段0有读/执行保护，段1有读/写保护。内存是请求分页式虚拟内存系统，它的虚拟地址为4位页号，10位偏移量。页表和保护如下所示（表中的数字均为十进制）：

段0		段1	
读/执行		读/写	
虚拟页号	页框号	虚拟页号	页框号
0	2	0	在磁盘
1	在磁盘	1	14
2	11	2	9
3	5	3	6
4	在磁盘	4	在磁盘
5	在磁盘	5	13
6	4	6	8
7	3	7	12

对于下面的每种情形，或者给出动态地址所对应的实（实际）内存地址，或者指出发生了哪种失效（缺页中断，或保护错误）。

- 读取页：段1，页1，偏移3；
 - 存储页：段0，页0，偏移16；
 - 读取页：段1，页4，偏移28；
 - 跳转到：段1，页3，偏移32。
38. 你能想象在哪些情况下支持虚拟内存是个坏想法吗？不支持虚拟内存能得到什么好处呢？请解释。
39. 构造一个柱状图，计算你的计算机中可执行二进制文件大小的平均值和中间值。在Windows系统中，观察所有的.exe和.dll文件；在UNIX系统中，观察/bin、/usr/bin、/local/bin目录下的所有非脚本文件的可执行文件（或者使用file工具来查找所有的可执行文件）。确定这台机器的最优页面大小，只考虑代码（不包括数据）。考虑内部碎片和页表大小，对页表项的大小做出合理的假设。假设所有的程序被执行的可能性相同，所以可以同等对待。
40. MS-DOS中的小程序可以编译成.COM文件。这些文件总是装载到0x100地址的一个内存段，这个内存段用作代码、数据和堆栈。转移执行的指令（如JMP、CALL）和访问静态数据的指令把地址编译进目标代码中。写一个程序重定向这个程序文件，使之可以在任意开始地址处运行。读者的程序必须扫描代码，寻找指向固定内存地址的目标代码，然后在重定向范围内修改那些指向内存单元的地址。可以在汇编语言程序正文中找到这些目标地址。注意，要想不借助于额外的信息就出色完成这项工作通常是不可能的，因为有些数据字的值和指令目标代码相仿。
41. 编写一个程序，它使用老化算法模拟一个分页系统。页框的数量是参数。页面访问序列从文件中读取。对于一个给定的输入文件，列出每