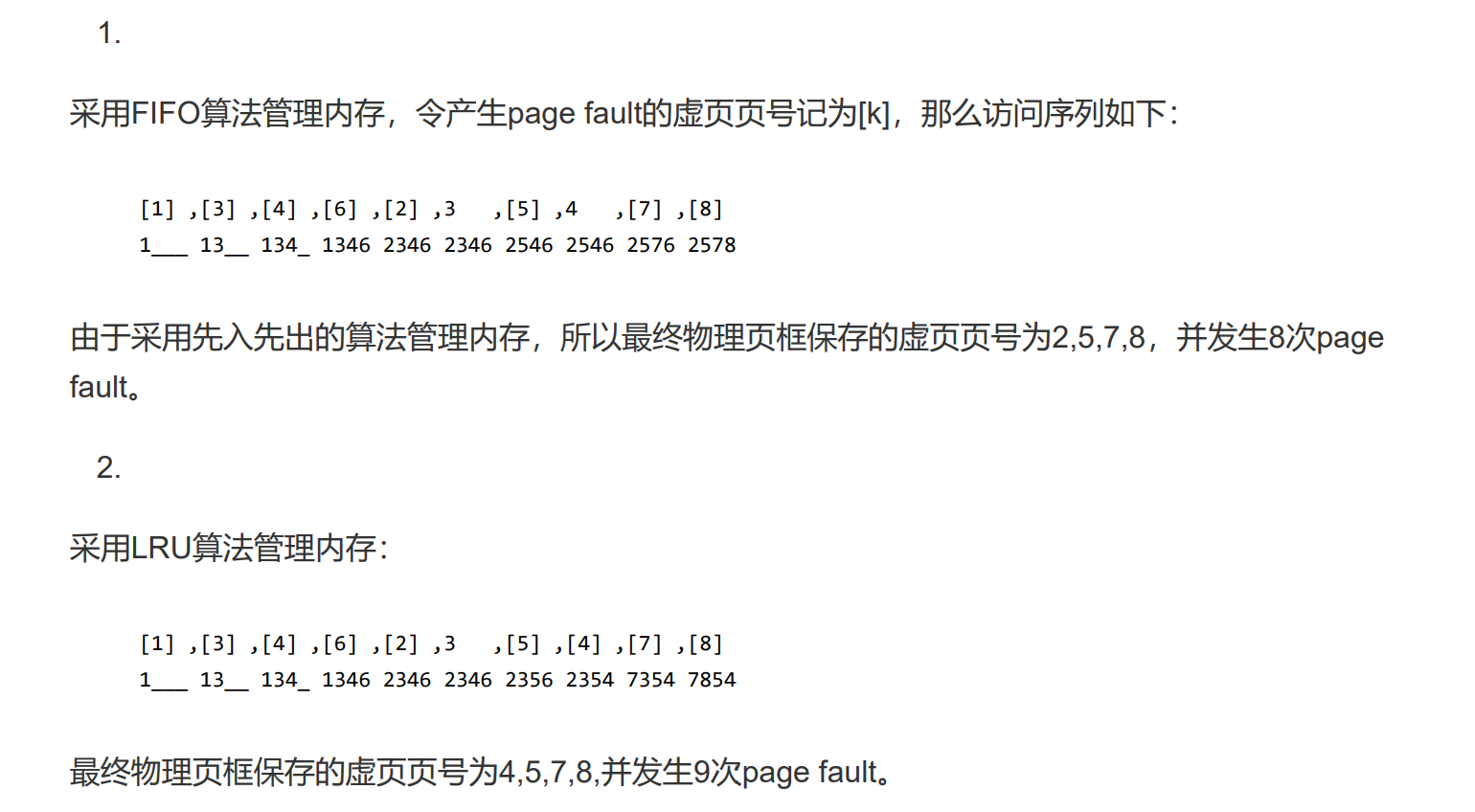
**作业9**

**刘子扬2020K8009929043**

9.1 假设一台计算机上运行的一个进程其地址空间有8个虚页（每个虚页大小为4KB，页号为1至8），操作系统给该进程分配了4个物理页框（每个页框大小为4KB），该进程对地址空间中虚页的访问顺序为 1 3 4 6 2 3 5 4 7 8。假设分配给进程的4个物理页框初始为空，请计算：

（1）如果操作系统采用FIFO算法管理内存，那么该进程访存时会发生多少次page fault？当进程访问完上述虚页后，物理页框中保存的是哪些虚页？

（2）如果操作系统采用LRU算法管理内存，请再次回答（1）中的两个问题



9.2 假设一台计算机给每个进程都分配4个物理页框，每个页框大小为512B。现有一个程序对一个二维整数数组（uint32 X[32][32]）进行赋值操作，该程序的代码段占用一个固定的页框，并一直存储在内存中。程序使用剩余3个物理页框存储数据。该程序操作的数组X以列存储形式保存在磁盘上，即X[0][0]后保存的是X[1][0]、X[2][0]…X[31][0]，然后再保存X[0][1]，以此类推。当程序要赋值时，如果所赋值的数组元素不在内存中，则会触发page fault，操作系统将相应元素以页框粒度交换至内存。如果该进程的物理页框已经用满，则会进行页换出。该程序有如下两种写法。

写法1：

for(int i=0;i<32;i++)

for(int j=0;j<32;j++)

X[i][j] = 0

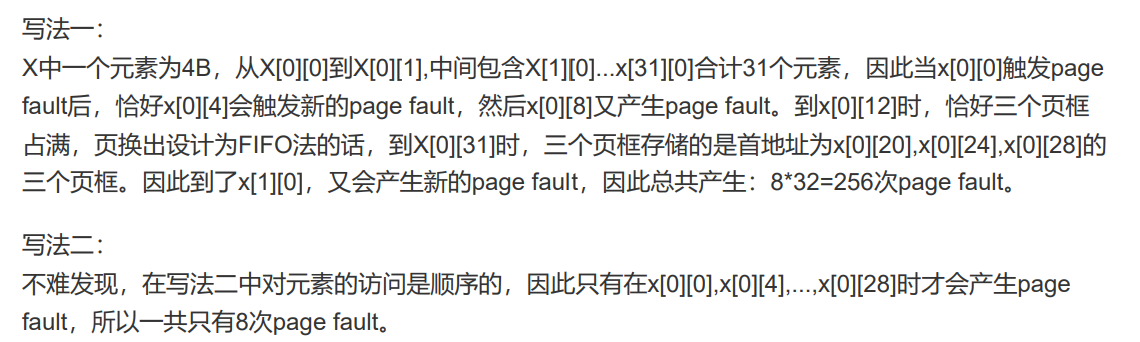
写法2：

for(int j=0;j<32;i++)

for(int i=0;i<32;j++)

X[i][j] = 0

请分析使用这两种写法时，各自会产生多少次page fault？（注：请写出分析或计算过程）



9.3 假设一个程序有两个段，其中段0保存代码指令，段1保存读写的数据。段0的权限是可读可执行，段1的权限是可读可写，如下所示。该程序运行的内存系统提供的虚址空间为14-bit空间，其中低10-bit为页内偏移，高4-bit为页号。

表格

描述已自动生成

当有如下的访存操作时，请给出每个操作的实际访存物理地址或是产生的异常类型（例如缺页异常、权限异常等）

1. 读取段1中page 1的offset为3的地址
2. 向段0中page 0的offset为16的地址写入
3. 读取段1中page 4的offset为28的地址
4. 跳转至段1中page 3的offset为32的地址

