操作系统第九章习题

9.1

当一个进程试图访问一个在页表条目中被标记为无效的页，即缺页时，就会发生页故障。

操作系统会进行调页操作，页面故障会产生一个中断，在特权模式下调用操作系统的代码。然后操作系统检查一些内部表（通常与该进程的进程控制块一起保存）以确定该页是否在磁盘上。如果该页在磁盘上（即它确实是一个有效的内存引用），操作系统就会分配一个空闲的帧，启动磁盘I/O，将所需的页读入新分配的帧，并启动下一个进程。当磁盘I/O完成后，与进程和页表一起保存的内部表被更新，以表明该页现在在内存中。被非法地址陷阱打断的指令被重新启动。该进程现在可以访问该页，就像它一直在内存中一样。

9.3

9EF -> 0EF

111 -> 211

700 -> D00

0FF -> EFF

9.4

LRU:4星，不受贝拉迪异常影响

FIFO：1星，受贝拉迪异常影响

OPT：5星，不受贝拉迪异常影响

Second-chance replacement：3星，不受贝拉迪异常影响

9.5

对于每个内存访问操作，都需要查询页表，以检查相应的页是否驻留在内存中，并确定是否触发了页故障。这些检查必须在硬件（MMU）中进行。一个TLB可以作为一个缓存，并提高查询操作的性能。

9.7

在这个系统中，每页可以容纳50个整数（整数的大小为4字节），因此A的一行需要2页，整个A需要2\*100=200页。

当数组A被逐行访问时，每行产生2个页面故障，因为对一个页面的第一次引用总是产生一个页面故障。使用LRU，它将产生200个页面故障。

当数组A被逐列访问时，进程在每个外部循环（I）中引用了100个页面，这就是程序的工作集。但是我们只有2个框架，因此每个数组引用将产生一个页面故障。使用LRU，它将产生100\*100=10,000个页面故障。

9.8

|  |  |  |  |
| --- | --- | --- | --- |
| frames number | LRU | FIFO | Optional |
| 1 | 20 | 20 | 20 |
| 2 | 18 | 18 | 15 |
| 3 | 15 | 16 | 11 |
| 4 | 10 | 14 | 8 |
| 5 | 8 | 10 | 7 |
| 6 | 7 | 10 | 7 |
| 7 | 7 | 7 | 7 |

9.15

a. 会改变，当一个页面故障发生时，一个tread从运行状态变为阻塞状态。当一个页面故障发生时，进程开始等待I/O操作的完成。

操作系统检查该页是否真的无效或只是在磁盘上，找到一个空闲的内存帧，安排一次磁盘访问将该页加载到该帧中，当磁盘I/O完成后用新的逻辑-物理映射更新页表，更新该条目的有效位，并最终重新启动进程，将其状态从阻塞状态变为就绪状态。

b. 不一定。如果在TLB中没有找到一个页表项（TLB缺失），页号被用来索引和处理页表。如果该页已经在主内存中，那么TLB被更新以包括新的页条目，同时由于不需要I/O操作，进程继续执行。如果该页不在主存中，就会产生一个页故障。在这种情况下，进程需要改变为阻塞状态，等待I/O访问磁盘。这与第一个问题中的程序相同。

c. 不会改变，因为不需要I/O操作，因为地址引用在页表中被解决了，这表明需要的页已经加载在主存中。

9.18

由于页面大小为2^12，所以页表大小为2^20。因此，低阶的12位0100 0101 0110被用作进入页面的位移，而剩下的20位0001 0001 0001 0010 0011被用作页表的位移。然后将偏移位与产生的物理页号（来自页表）相连接，形成最终地址。

9.21

LRU: 18

FIFO: 17

Optimal: 13

9.22

a.

0xE12C -> 0x312C

0x3A9D -> 0xAA9D

0xA9D9 -> 0x59D9

0x7001 -> 0xF001

0xACA1 -> 0x5CA1

b.0x4444

c.{9,1,14,13,8,0,4,2}

9.32

抖动是由于进程所需的最小页数分配不足而造成的，迫使它不断出现页面故障。系统可以通过评估CPU的利用率与多程序化水平的比较来检测阻塞。它可以通过减少多程序的水平来消除。

9.34

当设置为一个较小的值时，有可能低估一个进程的驻留页集，允许一个进程被安排，即使它所需要的所有页面没有被托管。这可能导致大量的页面错误。

当设置为一个较大的值时，高估一个进程的驻留集可能会阻止许多进程被安排，即使它们需要的页面驻留了。然而，一旦一个进程被调度，当高估驻留集时，就不可能产生一个页面错误