Chapter 9

9.1 Under what circumstances do page faults occur? Describe the actions taken by the operating system when a page fault occurs.

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时，就会发生页故障。
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。

9.3 Consider the page table shown in Figure 9.30 for a system with 12-bit virtual and physical addresses and with 256-byte pages. The list of free page frames is *D*, *E*, *F* (that is, *D* is at the head of the list, *E* is second, and *F* is last).

Page	Page Frame
0	–
1	2
2	C
3	A
4	–
5	4
6	3
7	–
8	B
9	0

Convert the following virtual addresses to their equivalent physical addresses in hexadecimal. All numbers are given in hexadecimal. (A dash for a page frame indicates that the page is not in memory.)

• 9EF

• 111

• 700

• **0FF**

1. 9EF -> 0EF
2. 111 -> 211
3. 700 -> D00
4. 0FF -> EFF

9.4 Consider the following page-replacement algorithms. Rank these algo- rithms on a five-point scale from “bad” to “perfect” according to their page-fault rate. Separate those algorithms that suffer from Belady’s anomaly from those that do not.

1. **LRU replacement**
2. **FIFO replacement**
3. **Optimal replacement**
4. **Second-chance replacement**

排序：OPT>LRU>Second-chance replacement>FIFO

其中 FIFO replacement 受到 Belady’s anomaly 影响

9.5 Discuss the hardware support required to support demand paging.

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 Consider the two-dimensional array A:

```
int A[][] = new int[100][100];
```

where `A[0][0]` is at location 200 in a paged memory system with pages of size 200. A small process that manipulates the matrix resides in page 0 (locations 0 to 199). Thus, every instruction fetch will be from page 0. For three page frames, how many page faults are generated by the following array-initialization loops? Use LRU replacement, and assume that page frame 1 contains the process and the other two are initially empty.

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。

a. 在这种情况下，数组A被逐行访问，因此每行产生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。

b. 在这种情况下，数组A被逐列访问，因此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 Consider the following page reference string:

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

How many page faults would occur for the following replacement algorithms, assuming one, two, three, four, five, six, and seven frames? Remember that all frames are initially empty, so your first unique pages will cost one fault each.

• LRU replacement

• FIFO replacement

• Optimal replacement

Number of frames	LRU	FIFO	OPT
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 simplified view of thread states is Ready, Running, and Blocked, where a thread is either ready and waiting to be scheduled, is running on the processor, or is blocked (for example, waiting for I/O). This is illustrated in Figure 9.31. Assuming a thread is in the Running state, answer the following questions, and explain your answer:

1. **Will the thread change state if it incurs a page fault? If so, to what state will it change?**
2. **Will the thread change state if it generates a TLB miss that is resolved in the page table? If so, to what state will it change?**
3. **Will the thread change state if an address reference is resolved in the page table? If so, to what state will it change?**

1. 会，当一个页面故障发生时，一个线程从运行状态变为阻塞状态。当一个页面故障发生时，线程开始等待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，并最终重新启动线程，将其状态从阻塞状态变为就绪状态。
2. 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。
3. 不会，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 A certain computer provides its users with a virtual memory space of 2^32 bytes. The computer has 2^22 bytes of physical memory. The virtual memory is implemented by paging, and the page size is 4,096 bytes. A user process generates the virtual address 11123456. Explain how the system establishes the corresponding physical location. Distinguish between software and hardware operations.

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 Consider the following page reference string:

7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 7, 1, 0, 5, 4, 6, 2, 3, 0 , 1.

Assuming demand paging with three frames, how many page faults would occur for the following replacement algorithms?

• LRU replacement

• FIFO replacement

• Optimalreplacement

LRU：18

7; 7 2; 7 2 3; 1 2 3; 1 2 5; 3 2 5; 3 4 5; 3 4 6; 7 4 6; 7 1 6; 7 1 0; 5 1 0; 5 4 0; 5 4 6; 2 4 6; 2 3 6; 2 3 0; 1 3 0

FIFO：17

7; 7 2; 7 2 3; 1 2 3; 1 5 3; 1 5 4; 6 5 4; 6 7 4; 6 7 1; 0 7 1; 0 5 1; 0 5 4; 6 5 4; 6 2 4; 6 2 3; 0 2 3; 0 1 3

OPT： 13
7; 7 2; 7 2 3; 1 2 3; 1 5 3; 1 5 4; 1 5 6; 1 5 7; 1 5 0; 1 4 0; 1 6 0; 1 2 0; 1 3 0

9.22 The page table shown in Figure 9.32 is for a system with 16-bit virtual and physical addresses and with 4,096-byte pages. The reference bit is set to 1 when the page has been referenced. Periodically, a thread zeroes out all values of the reference bit. A dash for a page frame indicates the page is not in memory. The page-replacement algorithm is localized LRU, and all numbers are provided in decimal.

Convert the following virtual addresses (in hexadecimal) to the equivalent physical addresses. You may provide answers in either

- `0xE12C` -> `0x312C`
- `0x3A9D` -> `0xAA9D`
- `0xA9D9` -> `0x59D9`
- `0x7001` -> `0xF001`
- `0xACA1` -> `0x5CA1`

From what set of page frames will the LRU page-replacement algorithm choose in resolving a page fault?

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

9.32 What is the cause of thrashing? How does the system detect thrashing? Once it detects thrashing, what can the system do to eliminate this problem?

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的利用率与多道程序程度的比较来检测阻塞。它可以通过减少多道程序程度来消除。

9.34 Consider the parameter used to define the working-set window in the working-set model. When is set to a small value, what is the effect on the page-fault frequency and the number of active (nonsuspended) processes currently executing in the system? What is the effect when is set to a very high value?

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