

1. Explain the following terms:

Fragmentation fault：好像没看到这个概念。只有内部碎片和外部碎片的概念。

TLB：转换表缓冲区，可以理解为查找内存的专用高速硬件缓冲（其思想类似于CPU的cache），它由两部分条目：键和值组成（通常为页码和帧码）。在典型应用中，TLB只包含少数的页表条目，当CPU产生一个逻辑地址后，页码发送到TLB，如果找到这个页码，它的帧码也就立即可用，用于访问内存。如果未命中，则需要访问页表，并将访问到的页码和帧码加入到TLB中。

Page fault：Page fault是一种当MMU发现所要访问的virtual page不可用（或者未被分配实际的物理空间）时CPU产生的中断，它将调用 Page fault服务子程序来为触发它的请求分配一段实际的物理空间，并更新page table。

Demand paging：Demand paging是一种内存分配策略，例如当我们调用malloc时，系统只是声明这段内存空间被分配了，即分配了虚拟地址空间，但实际的物理内存空间未被分配，而是当真正需要访问的时候，例如memset时，系统才真正为其分配物理内存空间，并更新页表。这样做的好处是避免了读入不需要的页，减少了交换时间和所需要的物理内存空间。

2. Introduce the concept of thrashing, and explain under what circumstance thrashing will happen.

如果进程没有需要支持活动使用页面的帧数，那么它将产生缺页错误并置换某个页面，但是，由于它的所有页面都在使用中，所以必须立即置换需要再次使用的页面。这样很快又会产生缺页错误，这种高度的页面调度活动称为“抖动thrashing”，表示一个进程的调页时间多于它的执行时间。

书中P271页对抖动的产生原因是这样阐述的：CPU调度程序再CPU利用率降低时会增加多道程度。新进程试图从其他运行进程中获取帧来启动，而其他运行进程也需要这些帧，从而导致了更多的缺页错误和更长的调页设备队列，因此CPU利用率下降，而CPU调度程序将再次试图增加多道程度，（有点类似于死锁），这样就出现了抖动，系统吞吐量骤降，进程都在忙于调页。

- ### 3. Consider a paging system with the page table stored in memory.
- If a memory reference takes 50 nanoseconds, how long does a paged memory reference take?
 - If we add TLBs, and 75 percent of all page-table references are found in the TLBs, what is the effective memory reference time? (Assume that finding a page-table entry in the TLBs takes 2 nanoseconds, if the entry is present.)

a. 需要100ns, 50ns用于访问页表, 50ns用于从页表获得地址后再次访问内存。

b. $75\% (50+2) + 25\% (100) = 64\text{ns}$

4. Assume a program has just referenced an address in virtual memory. Describe a scenario how each of the following can occur: (If a scenario cannot occur, explain why.)

- TLB miss with no page fault : page已在内存中, 地址映射不在TLB中。
- TLB miss and page fault: page不在内存中, 需要分配实际内存, 地址映射也不在TLB中。
- TLB hit and no page fault: page在内存中, 地址映射也在TLB中, 因此是最近访问过的内存块。
- TLB hit and page fault: 不会发生, 因为发生page fault说明不在内存中, 但又在TLB中, 而TLB是一段内存页表的缓存, 这种情况不可能发生。

5. Assume we have a demand-paged memory. The page table is held in registers. It takes 8 milliseconds to service a page fault if an empty page is available or the replaced page is not modified, and 20 milliseconds if the replaced page is modified. Memory access time is 100 nanoseconds. Assume that the page to be replaced is modified 70 percent of the time. What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds?

$$200\text{ns} = (1-p)100\text{ns} + 0.3p \cdot 8\text{ms} + 0.7p \cdot 20\text{ms} \Rightarrow p = 6 \cdot 10^{-6}$$

6. 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: 18
- FIFO replacement: 17
- Optimal replacement : 13