

操作系统作业 4

1.

段错误:

当你访问一块不允许访问的内存时，操作系统就会返回一个“段错误”。例如：向未使用或未申请的段读取内容，向只读段写内容，向未使用或未申请段写内容。

TLB:

转换表缓冲区。虚拟内存机制需要将虚拟地址转换为物理地址，就需要页表，但当内存较大时，使用页表查找延时太高，解决方法之一就是使用caching技术，地址转换中的cache技术实际就是TLB。TLB条目包括标签和值两部分，根据虚拟地址的部分高位再TLB中进行快速查找标签，如果找到，该标签对应的值就可以用于计算物理地址；如果没有找到，则使用页表进行查找。

Page fault:

在请求调页过程中，如果进程尝试访问那些尚未调入内存中的页面时会长生缺页错误(page fault)

Deman paging:

将可执行程序从磁盘加载到内存有两种选择，一种是将整个程序加载到物理内存，这会导致所有的执行代码都加载到内存而不管是否被最终使用，而另一种仅在需要时才载入页面，这种技术就是请求调页。请求调页在需要某个页面时会从页表中读取该页面是否有效，如果有效说明页面已经载入到内存可以执行，如果无效，就会缺页错误，并执行处理缺页错误的程序。

2.

thrashing抖动:

如果一个进程没有足够的帧来满足它所需要的帧数量，这就会导致频繁的缺页错误，因为一个页面刚刚被换出不久又会重新被换入，这种情况称为thrashing (抖动)

3.

a.内存引用50ns，则分页内存引用100ns=页表引用50ns+物理内存引用50ns

b. $0.75 \times (50 + 2) + 0.25 \times 100 = 64\text{ns}$

4.

- 页码没有在TLB中（TLB中没有标签与虚拟地址高位匹配），但其对应页面在内存中（页码在页表中对应的帧为有效）
- 页码没有在TLB中，其对应页面不在内存中，需要从外存中换入。
- 页码在TLB中，其对应页面在内存中
- 页码在TLB中，但其对应页面不在内存中

5.

$(1-p) \times 100 + p \times (0.3 \times 8000 + 0.7 \times 20000) < 200$

$p < 0.6\%$

最大缺页错误率为0.6%

6.

LRU:page fault=18

FIFO:page fault=17

Optimal:page fault=13

7.

在页面转换算法中，可用帧数量更大反而具有更大的缺页错误数量，这种与主观感觉相悖的现象称为Belady反常。

堆栈算法不会发生Belady异常，这种算法的特点是：帧数为 n 的内存页面集合是帧数为 $n+1$ 的内存页面集合的子集。