

OS hw4

1. T1

Explain the following terms:

Segmentation fault

TLB

Page fault

Demand paging

- **Segmentation fault:** 当程序访问了无法访问或未分配给它的内存段时，会发生段错误，可能是由于指针错误、数组越界或者试图执行不允许的操作等情况
- **TLB:** TLB是一种硬件高速缓存，保存了部分页号和帧号的对应关系，在查询页表前，会先查找TLB，若对应的项存在TLB中，可直接得到帧号，否则再去查找页表，用于加速虚拟内存地址到物理内存地址的转换过程
- **Page fault:** 页错误指的是当程序访问虚拟内存中的某个页面时，该页面当前不在物理内存中，需要从磁盘或其他存储介质中加载到内存。发生页错误时，操作系统会将程序暂停，并将所需的页面从磁盘加载到内存中，然后更新页表以反映新的物理内存位置
- **Demand paging:** 需求分页是一种虚拟内存管理技术，只在需要将页面加载到内存中，可以节省内存空间，并且只有当程序访问某个页面时，才需要将其加载到内存中，从而减少了不必要的磁盘访问

2. T2

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

- **concept:** 系统频繁地进行页面调度，刚刚换出到外存的页面马上要调入内存，刚刚调入内存的页面马上要调出到外存
- 发生情形：
 - 当使用高度的进程并发，系统物理内存不够

- 全局页面替换，过度的页面调度，消耗大量的处理器时间，从而导致系统陷入抖动状态

3. T3

Consider a paging system with the page table stored in memory.

- a. If a memory reference takes 50 nanoseconds, how long does a paged memory reference take?
- b. 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.)

- 访问页表需要50ns，访问内存中的目标数据需要50ns，所以一共100ns
- $0.75 \times (50ns + 2ns) + 0.25 \times (2ns + 50ns + 50ns) = 64.5ns$

4. T4

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

- 设页面错误率为 p :
一个页面错误: $t = (0.7 \times 20ms) + (0.3 \times 8ms) = 14ms + 2.4ms = 16.4ms$
所以平均时间: $p \times t + (1 - p) \times 100ns \leq 200ns$
得到 $p \leq 6.098 \times 10^{-6}$

5. T5

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
- Optimal replacement

- LRU : 18 个
- FIFO : 17 个
- Optimal : 13 个

6. T6

Explain what Belady's anomaly is, and what is the feature of stack algorithms which never exhibit Belady's anomaly

- Belady's anomaly : 分配的页面数增多但缺页率反而提高的异常现象
- feature: 在堆栈算法中, 当分配帧为 n 时在内存中的页面, 当 n 变为 $n+1$ 时这些页面也会在内存中, 保证了不受页面框数量的影响

7. T7

Suppose that a disk drive has 6,000 cylinders, numbered 0 to 5999. The drive is currently serving a request at cylinder 2150, and the previous request was at cylinder 1805. The queue of pending requests, in FIFO order, is: 2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681. Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms?

- FCFS
- SSTF
- SCAN

- d. LOOK
- e. C-SCAN
- f. C-LOOK

算法	访问顺序	距离
FCFS	2150, 2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681	13011
SSTF	2150, 2069, 2296, 2800, 3681, 4965, 1618, 1523, 1212, 544, 356	7586
SCAN	2150, 2296, 2800, 3681, 4965, 5999, 2069, 1618, 1523, 1212, 544, 356	9492
LOOK	2150, 2296, 2800, 3681, 4965, 2069, 1618, 1523, 1212, 544, 356	7424
C-SCAN	2150, 2296, 2800, 3681, 4965, 5999, 0, 356, 544, 1212, 1523, 1618, 2069	11917
C-LOOK	2150, 2296, 2800, 3681, 4965, 356, 544, 1212, 1523, 1618, 2069	9137

8. T8

Consider a file system similar to the one used by UNIX with indexed allocation, and assume that every file uses only one block. How many disk I/O operations might be required to read the contents of a small local file at /a/b/c in the following two cases? Should provide the detailed workflow.

- a. Assume that none of the disk blocks and inodes is currently being cached.
- b. Assume that none of the disk blocks is currently being cached but all inodes are in memory.

- 读根目录

/a的inodes

/a的disk blocks

/a/b的inodes

/a/b的disk blocks

/a/b/c的inodes

/a/b/c的disk blocks

七次

- 过程同上，但inodes都已缓存，所以7-3为4次

9. T9

What is the difference between hard link and symbolic link

- 硬链接指向与原始文件相同的物理数据块，与原始文件共享相同的索引节点，因此它们都具有相同的文件内容和元数据，创建硬连接会增加文件的链接计数，删除硬链接文件的链接计数减1，直到减为0时文件被删除
- 符号链接是一个特殊的文件，包含指向另一个文件或目录的路径。符号链接是一个独立的文件实体，它有自己的索引结点和元数据。创建符号链接不改变文件的链接计数，删除指向的文件，符号链接仍存在，但无法访问指向的文件

10. T10

What are the differences between data journaling and metadata journaling? Explain the operation sequence for each of the two journaling methods

数据日志：

1. 写入事务的内容（包括 TxB、元数据和数据）
2. 写TxE到日志
3. 写元数据和数据到磁盘对应位置

元数据日志：

1. 写入事务的内容（包括 TxB、元数据），将数据并行写入其磁盘位置
2. 写TxE到日志
3. 写元数据到磁盘对应位置

区别在数据处理不同：

- 数据日志先将数据写入日志，之后再写入磁盘；
- 元数据日志将元数据写入日志时会并行将数据写入磁盘