# Fall 2023 COMP 3511 Homework Assignment 4 (HW4)

## Handout Date: Nov 20, 2023, Due Date: Dec 4, 2023

| | |
|---|---|
| Name | Chloe Hu |
| Student ID | 21044009 |
| ITSC email | chuap@connect.ust.hk |

**Please read the following instructions carefully before answering the questions:**

- You should finish the homework assignment **individually**.
- This homework assignment contains **four** parts.
- When you write your answers, please try to be precise and concise.
- **Homework Submission:** submitted to **Homework #4** on **Canvas**.
- TA responsible for HW4: Yuheng ZHAO (yzhaoep@cse.ust.hk)

### [20 points] Problem 1 Multiple Choices

Please write down your answers in the boxes below:

| Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 |
|---|---|---|---|---|---|---|---|---|---|
| C | D | D | D | B | C | B | D | B | A |

1) Consider that a system uses 2-level paging scheme and has a TLB hit ratio of 90%. It requires 10 nanoseconds to access the TLB, and 100 nanoseconds to access main memory. What is the effective memory access time for this system? (A TLB hit directly results in page translation, i.e., frame number)

A) 99ns

B) 110ns

C) 120ns

D) 130ns

2) Assume a process is allocated with $m$ frames (initially all empty). The page-reference string has length $p$, in which there are $n$ distinct page numbers. Which of the following statement is true?

A) The maximum number of page faults is $m$.

B.) The maximum number of page faults is $p$-$n$.

C) The maximum number of page faults is $n$.

D) The maximum number of page faults is $p$.

3) Consider a paging scheme with 32-bit logical address and 2KB page size. Suppose that a single-level page table is used with a page table entry (PTE) size of 4 bytes, and each PTE contains 9 control bits. What are the sizes of logical address space and physical address space, respectively?

A) 8GB and 8GB

B) 4GB and 16GB

C) 4GB and 4GB

D) 4GB and 8GB

4) Consider a two-level page table with the following address structure.

| page number | | page offset |
|---|---|---|
| $p_1$ (10 bits) | $p_2$ (10 bits) | $d$ (12 bits) |

What is the page numbers of the logical address 0x 5504 3316 (in hexadecimal) in the outer page table ($p_1$) and the inner page table ($p_2$), respectively?

A) 0x316; 0x043

B) 0x154; 0x043

C) 0x316; 0x109

D) 0x550; 0x043

5) Consider a process with the following page reference strings from t0 to t9:

t: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

P1: 7, 2, 3, 1, 2, 5, 3, 7, 6, 7

Suppose that working set window is 4, what is the working set at time 4 for the process?

A) [1, 2, 3]

B) [1, 2, 3, 7]

C) [1, 2, 5]

D) [1, 2, 3, 5]

6) Consider a disk with average seek time is 5 ms, RPM is 10,000, transfer rate is 100MB/s, and a 1MB read occurs at a random location. The effective bandwidth or transfer rate is:

A) 60MB/s

B) 100MB/s

C) 55MB/s

D) 66MB/s

7) Consider a file system stored on a disk with block size of 1024 bytes. Suppose the disk address (block number) uses 4 byte. Please compute the total number of blocks required to allocate a file of size 110,550 bytes under (a) contiguous and (b) indexed allocation.

A) 100 and 109

B) 108 and 110

C) 108 and 109

D) 100 and 110

8) Which of the following data structures are commonly used in-memory structures that are used to implement a file system.

A) A directory-structure cache holds the directory information of recently accessed directories.

B) A system-wide open-file table contains a copy of the FCB of each open file.

C) A per-process open-file table contains a pointer to the appropriate entry in the system-wide open-file table.

D) All of the above

9) Suppose process A has 10 files open, process B has 8 files open and process C has 8 files. Four files are shared among the three processes. How many entries are in the system-wide open-file tables?

A) 20

B) 18

C) 26

D) 16

10) What is meaning of rights-set?

A) It is a subset of all valid operations that can be performed on the object

B) It is a subset consist of read and write

C) It is a subset consist of read, write and execute

D) None of the mentioned

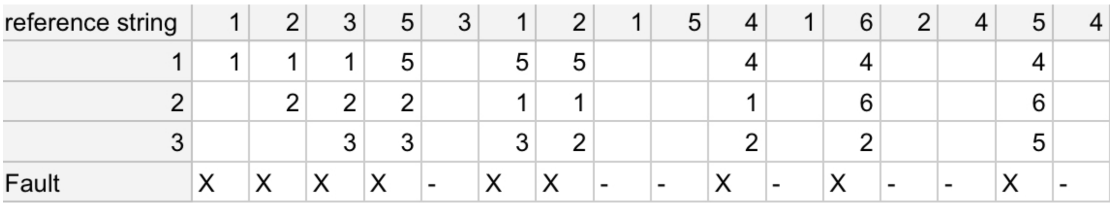
## [30 points] Problem 2 Page Replacement Algorithms

Consider the following page reference string:

1, 2, 3, 5, 3, 1, 2, 1, 5, 4, 1, 6, 2, 4, 5, 4

Assuming demand paging with 3 frames. Please illustrate each step that the following replacement algorithms work for this reference string and compute the page faults in each algorithm.

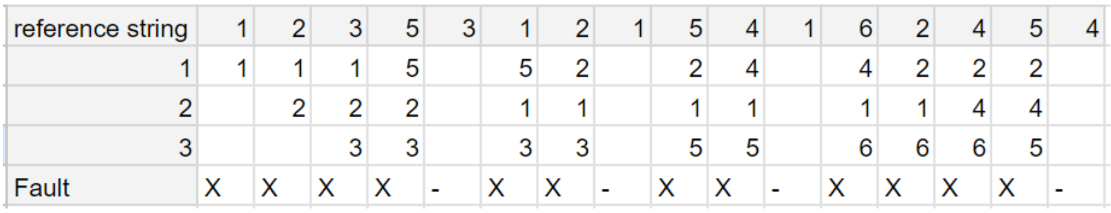
1) FIFO replacement



| reference string | 1 | 2 | 3 | 5 | 3 | 1 | 2 | 1 | 5 | 4 | 1 | 6 | 2 | 4 | 5 | 4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 5 |  | 5 | 5 |  |  | 4 |  | 4 |  |  | 4 |  |
| 2 |  | 2 | 2 | 2 |  | 1 | 1 |  |  | 1 |  | 6 |  |  | 6 |  |
| 3 |  |  | 3 | 3 |  | 3 | 2 |  |  | 2 |  | 2 |  |  | 5 |  |
| Fault | X | X | X | X | - | X | X | - | - | X | - | X | - | - | X | - |

9 page faults

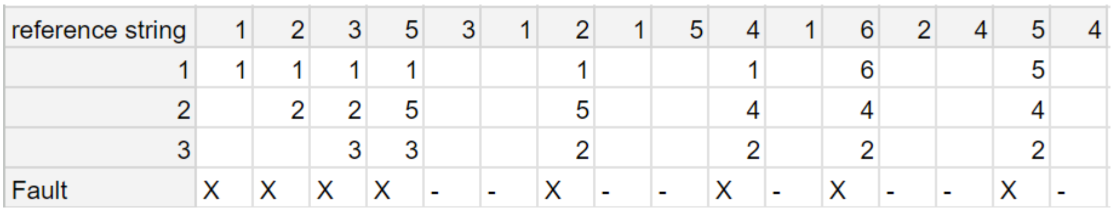
2) LRU replacement



| reference string | 1 | 2 | 3 | 5 | 3 | 1 | 2 | 1 | 5 | 4 | 1 | 6 | 2 | 4 | 5 | 4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 5 |  | 5 | 2 |  | 2 | 4 |  | 4 | 2 | 2 | 2 |  |
| 2 |  | 2 | 2 | 2 |  | 1 | 1 |  | 1 | 1 |  | 1 | 1 | 4 | 4 |  |
| 3 |  |  | 3 | 3 |  | 3 | 3 |  | 5 | 5 |  | 6 | 6 | 6 | 5 |  |
| Fault | X | X | X | X | - | X | X | - | X | X | - | X | X | X | X | - |

12 page faults

3) Optimal replacement



| reference string | 1 | 2 | 3 | 5 | 3 | 1 | 2 | 1 | 5 | 4 | 1 | 6 | 2 | 4 | 5 | 4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |  |  | 1 |  |  | 1 |  | 6 |  |  | 5 |  |
| 2 |  | 2 | 2 | 5 |  |  | 5 |  |  | 4 |  | 4 |  |  | 4 |  |
| 3 |  |  | 3 | 3 |  |  | 2 |  |  | 2 |  | 2 |  |  | 2 |  |
| Fault | X | X | X | X | - | - | X | - | - | X | - | X | - | - | X | - |

8 page faults

## [30 points] Problem 3 Disk Scheduling

Suppose that a disk drive has 1000 cylinders, numbered 0 to 999. The drive is currently serving a request at cylinder 153. The queue of pending requests, in FIFO order, is:

87, 110, 173, 126, 241, 313, 18, 554, 756, 818

Starting from the current head position (153), 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? The disk arm is moving from right to left (999 to 0). Note that for C-SCAN and C-LOOK, we assume the serving direction is "**From right to left**".

a) FCFS

Queue: 153, 87, 110, 173, 126, 241, 313, 18, 554, 756, 818

Distance = (153-87) + (110 – 87)+(173-110) + (176-126) +
(241-126)+(313-241)+(313-18)+(554-18)+(756-554)+(818-756) = 1484

b) SSTF

Queue: 153, 173, 126, 110, 87, 18, 241, 313, 554, 756, 818

Distance = (173-153) + (173 – 126)+(126-110) + (110-87) +
(87-18)+(241-18)+(313-241)+(554-313)+(756-554)+(818-756) = 975

c) SCAN

Queue: 153, 126, 110, 87, 18, 0, 173, 241, 313, 554, 756, 818

Distance = (153-126) + (126-110)+(110-87) + (87-18) +
(18-0)+(173-0)+(241-173)+(313-241)+(554-313)+(756-554)+(818-756) = 971

d) LOOK

Queue: 153, 126, 110, 87, 18, 173, 241, 313, 554, 756, 818

Distance = (153-126) + (126-110) +(110-87) + (87-18) + (173-18) +(241-173)
+(313-241) +(554-313) +(756-554) +(818-756) = 935

e) C-SCAN

Queue: 153, 173, 241, 313, 554, 756, 818, 999, 0, 18, 87, 110, 126

Distance = (173-153) + (241-173) +(313-241) +(554-313) +(756-554) +(818-756) +
(999-818) + (999-0) + (18-0) +(87-18) + (110-87) +(126-110) = 1971

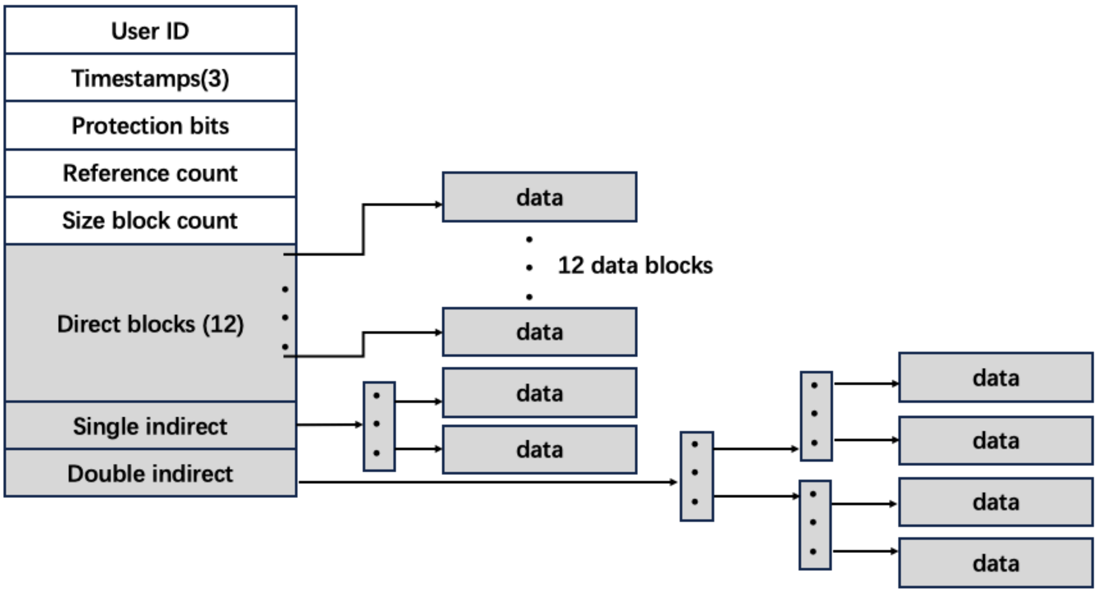
f) C-LOOK

Queue: 153, 173, 241, 313, 554, 756, 818, 18, 87, 110, 126

Distance = (173-153) + (241-173) +(313-241) +(554-313) +(756-554) +(818-756) +
(818-18) +(87-18) + (110-87) +(126-110) = 1573

## [20 points] Problem 4 File System

Consider a file system that has the following description:



- The hard disk is divided into **2048-byte** blocks.
- As shown in the figure, a file is stored in hard disk using indexed allocation. A File Control Block (FCB) contains:
    - **12** data block pointers, each of which is 4 bytes and each of which points to a disk block.
    - ONE single indirect pointer, which points to a disk block that contains data block pointers.
    - ONE double indirect pointer, which points to a block of pointers that point to other blocks of pointers that then point to data.
- The FCB also contains a user id (2 bytes), 3 timestamps (4 bytes each), 1 protection bits (2 bytes), a reference count (3 bytes), and a size block count (4 bytes).

a) (10 points) What is the maximum file size supported by this file system? Briefly explain.

Block size * (direct + indirect + double indirect) = 2048 * (12*(2048^0) + 1*(2048^1)+1*(2048^2)) = 8594153472 bytes = 8.59GB

b) (10 points) Consider file A with size 8KB, file B with size 1MB, and file C with size 200MB. Does the file system take the same amount of time to obtain the index of the last block of A,B,C and why?

The file system does not take the same amount of time for each file. File A can fit in four direct blocks of the FCB. To obtain the index of the last block, the file system only needs to read the FCB and access the fourth data block pointer.

File B, needs 1048576 / 2048 = 512 blocks to store its data. The first 12 blocks of file B can be stored in the direct blocks of the FCB. Since there are no ore free blocks, we are required to use the single indirect pointer which stores up to 2048 / 4 = 512 pointers and points to the remaining 500 blocks of file B. To obtain the index of the last block of B, the file system needs to read the FCB, access the single indirect pointer, read the single indirect block, and access the last data block pointer.

File C needs 209715200 / 2048 = 102400 blocks to store its data. The first 12 blocks are stored in the FCB and the next 512 are stored in the blocks from the single indirect pointer. This leaves 101876 blocks left to be stored. The double indirect pointer can store up to 512 * 512 data block pointers, which can point to the remaining 101876 blocks of file C.

To obtain the index of the last block of C, the file system needs to read the FCB, access the double indirect pointer, read the double indirect block, access the pointer, read the single indirect block, and access the last data block pointer.

Thus, the file system needs more disk accesses to obtain the index of the last block of larger files. The amount of time is summarised as follow C>B>A.