

Fall 2022 COMP 3511 Homework Assignment 4 (HW4)

Handout Date: Nov 18, 2022, Due Date: Dec 2, 2022

Name	Casper Kristiansson
Student ID	20938643
ITSC email	cok@connect.ust.hk

Please read the following instructions carefully before answering the questions:

- You should finish the homework assignment **individually**.
- This homework assignment contains **five** 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: Hao, WANG (hwangdv@connect.ust.hk)

Problem 1. [20 points] Multiple Choices.

Please write down your answers in the boxes below:

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

1) Assume a page-reference string for a process with m frames (initially all empty). The page-reference string has length p , and n distinct page numbers occur in it. Which of the following statement is true?

- A. The minimum number of page faults is m .
- B. The maximum number of page faults is $p-n$.
- C. The minimum number of page faults is n .
- D. None of the above.

2) Which of the following statement about the page replacement algorithms is true?

- A. FIFO is more efficient (fewer page faults) than second-chance algorithm.
- B. Second-chance algorithm does not suffer from the Belady's anomaly.
- C. The exact LRU algorithm is expensive to implement.
- D. LFU algorithm is not a stack algorithm and may have the Belady's anomaly.

3) Segment replacement algorithms are often more complex than page replacement algorithms because _____.

- A. Page replacement algorithm decides which frames to replace
- B. Segment replacement algorithms decides which frames to replace

C. Segments have variable sizes

D. Pages have variable sizes

4) A process is thrashing if _____, which results in serious performance problems.

A. it spends a lot of time executing, rather than paging

B. it spends a lot of time paging, rather than executing

C. it has a high page-fault rate

D. it needs to increase the degree of multiprogramming

5) The heads of the magnetic disk are attached to a _____ that moves all the heads as a unit.

A. spindle

B. track

C. cylinder

D. disk arm

6) RAID is short for Redundant Arrays of Independent Disks. For Both striped mirrors (RAID 1+0) or mirrored stripes (RAID 0+1): RAID level 0 provides _____ whereas RAID level 1 provides _____

A. performance, reliability

B. performance, redundancy

C. reliability, redundancy

D. redundancy, performance

7) The operating system keeps a small table containing information about all open files called _____.

A. system table

B. open-file table

C. info-file table

D. file-open count

8) Access Lists and Groups are used for file protection. In UNIX, supposing a file `game` has access privilege of 751 (`chmod 751 game`), which of the following statement is true?

TA responsible for HW4: Hao, WANG (hwangdv@connect.ust.hk)

- A. The owner can read, write, but not execute the file
- B. Any user can read the file, but not write or execute
- C. The group can write the file
- D. None of the above

9) An allocation method refers to how disk blocks are allocated for files. In which type of allocation method each file occupies a set of sequential blocks on the disk?

- A. dynamic-storage allocation
- B. static-storage allocation
- C. indexed allocation
- D. contiguous allocation

10) Protection domain specifies resources process may access. In Protection domain structure what means Access-right?

- A. The pair of object-name, rights-set.
- B. The pair of read-name, write-set.
- C. The pair of read-name, execute-set.
- D. The pair of object-name, execute-set.

Problem 2 [20 points] Two-level Paging

Consider a multi-level memory management scheme with the following format for virtual addresses:

Virtual Page #1	Virtual Page #2	Offset
6 bits	12 bits	14 bits

Virtual addresses are translated into physical addresses of the following form:

Physical Page #	Offset
18 bits	14 bits

Suppose the page table entries (PTE) are 32 bits in both outer and inner page tables.

1. What is the number of PTEs in each level (outer and inner page table), respectively? (5 points)

·TA responsible for HW4: Hao, WANG (hwangdv@connect.ust.hk)

Number of PTE in outer= 2^6 , and inner= 2^{12}

2. We have the following virtual addresses, please translate them into the physical addresses according to the page table. (To avoid losing all points for wrong answers, please describe in detail.) (15 points)

Address 1: 0x 5e8c1c93

Address 2: 0x 04e26e54

Address 3: 0x b5b55a55

Outer page table

Logical page number	Physical block number
1	2
2	3
⋮	⋮
23	unallocated
⋮	⋮
45	1
46	unallocated

Inner page table in physical block 1

Logical page number	Physical address
905	0x2db49000 0x2db48000
⋮	⋮
1749	0x3d45a000 0x3d458000

Inner page table in physical block 2

Logical page number	Physical address
905	0x13eb7000

·TA responsible for HW4: Hao, WANG (hwangdv@connect.ust.hk)

	0x13cb4000
:	:
1749	0x28ba0000

Address 1: 0x 5e8c1c93

Because this could be the first virtual page, we look at the first 6 bits of the address:

$$0x5e8c1c93 = 01011110100011000001110010010011$$

$$010111 = 23$$

Via the page table we can see that the physical address is Unallocated.

Address 2: 0x 04e26e54

The first virtual page, so we look at the first 6 bits of the address:

$$0x04e26e54 = 00000100111000100110111001010100$$

$$000001 = 1$$

Physical address: 2 (block 2)

We then calculate for virtual page 2 which will be the bits between 6-18.

$$001110001001 = 905$$

Physical address: 0x13cb4000

The total physical address = 0x13cb4000 + offset

The offset is the 14 last bits

$$0x13cb4000 = 00010011110010110100000000000000$$

$$\text{Offset} = 10111001010100$$

$$\text{Physical address} = 010011110010110110111001010100 = \underline{\underline{0x13CB6E54}}$$

Address 3: 0xb5b55a55

We look at the first 6 bits:

$$0xb5b55a55 = 10110101101101010101101001010101$$

$$101101 = 45$$

·TA responsible for HW4: Hao, WANG (hwangdv@connect.ust.hk)

Physical address: 1 (block 1)

We then calculate for virtual page 2 which will be the bits between 6-18

011011010101 = 1749

Physical address: 0x3d458000

0x3d458000 = 00111101010001011000000000000000

Finale Physical address = Physical + offset =

00111101010001011000000000000000+ 01101001010101 =

0111101010001011001101001010101

Physical address: 0x3D459A55

Problem 3 [24 points] Page Replacement

Consider the following page reference string for one process: 1, 4, 5, 2, 1, 3, 5, 2, 1, 3, 1, 2. Please illustrate each step that the following replacement algorithms work for the reference string and compute the page fault in each algorithm when demanding page is 3 and 4.

1) Optimal replacement

Optimal Replacement Demanding 3 6 Page Faults

Reference String	1	4	5	2	1	3	5	2	1	3	1	2
1	1	1	1	1	1	3	3	3	3	3	3	3
2		4	4	2	2	2	2	2	2	2	2	2
3			5	5	5	5	5	5	1	1	1	1
Fault	X	X	X	X	-	X	-	-	X	-	-	-

Optimal Replacement Demanding 4 5 Page Faults

Reference String	1	4	5	2	1	3	5	2	1	3	1	2
1	1	1	1	1	1	1	1	1	1	1	1	1
2		4	4	4	4	3	3	3	3	3	3	3
3			5	5	5	5	5	5	5	5	5	5
4				2	2	2	2	2	2	2	2	2
Fault	X	X	X	X	-	X	-	-	-	-	-	-

2) FIFO replacement

FIFO Demanding 3 10 Page Faults

Reference String	1	4	5	2	1	3	5	2	1	3	1	2
1	1	1	1	2	2	2	5	5	5	3	3	3
2		4	4	4	1	1	1	2	2	2	2	2
3			5	5	5	3	3	3	1	1	1	1
Fault	X	X	X	X	X	X	X	X	X	X	-	-

FIFO Demanding 4 6 Page Faults

Reference String	1	4	5	2	1	3	5	2	1	3	1	2
1	1	1	1	1	1	3	3	3	3	3	3	3
2		4	4	4	4	4	4	4	1	1	1	1
3			5	5	5	5	5	5	5	5	5	5
4				2	2	2	2	2	2	2	2	2
Fault	X	X	X	X	-	X	-	-	X	-	-	-

3) LRU replacement

LRU Demanding 3 10 Page Faults

Reference String	1	4	5	2	1	3	5	2	1	3	1	2
1	1	1	1	2	2	2	5	5	5	3	3	3
2		4	4	4	1	1	1	2	2	2	2	2
3			5	5	5	3	3	3	1	1	1	1
Fault	X	X	X	X	X	X	X	X	X	X	-	-

LRU Demanding 4 5 Page Faults

Reference String	1	4	5	2	1	3	5	2	1	3	1	2
1	1	1	1	1	1	1	1	1	1	1	1	1
2		4	4	4	4	3	3	3	3	3	3	3
3			5	5	5	5	5	5	5	5	5	5
4				2	2	2	2	2	2	2	2	2
Fault	X	X	X	X	-	X	-	-	-	-	-	-

Problem 4 [24 points] Disk Scheduling

Suppose a disk drive has 1000 cylinders numbered 0 to 999. The current head position is 345. The disk arm is moving from right to left (from 999 to 0).

Currently, the queue of the pending requests (in FIFO order) is:

523, 774, 292, 675, 105, 476

Write out the sequence of disk head positions in terms of cylinder numbers in order to satisfy all the pending requests starting from the current head position and the total distance (in cylinders) that the disk arm moves to satisfy

·TA responsible for HW4: Hao, WANG (hwangdv@connect.ust.hk)

all pending requests, for each of the following disk scheduling algorithms.
(FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK)

FCFS

Queue = 345, 523, 774, 292, 675, 105, 476

Distance = $(523 - 345) + (774 - 523) + (774 - 292) + (675 - 292) + (675 - 105) + (476 - 105) = 2235$

SSTF

Queue = 345, 292, 476, 523, 675, 774, 105

Distance = $(345 - 292) + (476 - 292) + (523 - 476) + (675 - 523) + (774 - 675) + (774 - 105) = 1204$

SCAN

Queue = 345, 292, 105, 0, 476, 523, 675, 774

Distance = $(345 - 292) + (292 - 105) + (105 - 0) + (476 - 0) + (523 - 476) + (675 - 523) + (774 - 675) = 1119$

C-SCAN

Queue = 345, 476, 523, 675, 774, 999, 0, 105, 292

Distance = $(476 - 345) + (523 - 476) + (675 - 523) + (774 - 675) + (999 - 774) + (999 - 0) + (105 - 0) + (292 - 105) = 1945$

LOOK

Queue = 345, 292, 105, 476, 523, 675, 774

Distance = $(345 - 292) + (292 - 105) + (476 - 105) + (523 - 476) + (675 - 523) + (774 - 675) = 909$

C-LOOK

Queue = 345, 476, 523, 675, 774, 105, 292

TA responsible for HW4: Hao, WANG (hwangdv@connect.ust.hk)

$$\begin{aligned} \text{Distance} &= (476 - 345) + (523 - 476) + (675 - 523) + (774 - 675) \\ &\quad + (774 - 105) + (292 - 105) = 1285 \end{aligned}$$

Problem 5 [12 points] File System

1. A variation of an inode is called a tinode (short for tiny inode). Assume the logical address is 8-bit, the size of a disk block is 2KB. A tinode has the following structure:

1) entries 0-3 are direct disk block pointers. 2) entry 4 is a single indirect. 3) entry 5 is a double indirect. 4) entry 6 is a triple indirect.

What is the maximum file size that can be supported in this file system?

Please answer with the following format: _ KB + _ MB + _ GB + _ TB. (4 points)

Because the logical address is 8 bit (1 byte) means that there is 1 byte pointer for the 2048-byte block.

$$\begin{aligned} \text{Block size} * (\text{direct} + \text{indirect} + \text{double indirect} + \text{triple indirect}) \\ &= 2048 * (4 * 2048^0 + 1 * 2048^1 + 1 * 2048^2 + 1 * 2048^3) \\ &= 17600780181504 \text{ bytes} \end{aligned}$$

$$17600780181504 = 8KB + 4MB + 8GB + 16TB$$

2. Consider a FAT-based (File allocation Table) file system. Entries in the table are 32-bits wide. A user wants to install a disk with 8,000,000,000 sectors (blocks).

a. Please briefly describe FAT-based file system. (2 points)

Fat (File Allocation Table) is a variation of the link allocation file system. It is a simple but efficient method for disk allocation where a section of the disk is used for a table called FAT. The table contains one entry for each block and its index. The system is based on linked list where each block contains the block number and the next block.

b. What is the potential problem? (Hint: check the size of the disk) (3 points)

A potential problem with FAT-based file systems is regarding the addresses stored in the table. Each entry is a 32-bit address because the operating system can at most handle 2^{32} sectors. This means that if a user wanted to use a disk with more sectors it would not be supported.

·TA responsible for HW4: Hao, WANG (hwangdv@connect.ust.hk)

c. Describe a solution to this problem and explain the trade-offs involved. (3 points)

A possible solution to this problem would be to increase the FAT table by dividing each entry into 16 bits or even lower. Doing this will allow a much bigger table size. But doing this will increase the complexity of the system and decrease its performance of it.