COMP2432 Operating Systems

Written Assignment 2

ZHOU Siyu April 13th

1. Page Replacement

3 memory frames

(a) FIFO: total page fault: 19

	0	1	2	3	4	5	2	4	3	2	5	4	2	5	6	1	3	2	1	0	2	4	3	2	1
F1	0	0	0	3	3	3	2	2	2	2	2	2	2	5	5	5	3	3	3	3	3	4	4	4	1
F2		1	1	1	4	4	4	4	3	3	3	3	3	3	6	6	6	2	2	2	2	2	3	3	3
F3			2	2	2	5	5	5	5	5	5	4	4	4	4	1	1	1	1	0	0	0	0	2	2

(b) optimal: total page fault: 14

	0	1	2	3	4	5	2	4	3	2	5	4	2	5	6	1	3	2	1	0	2	4	3	2	1
F1	0	0	0	3	3	5	5	5	5	5	5	5	5	5	6	1	1	1	1	0	0	4	4	4	1
F2		1	1	1	4	4	4	4	3	3	3	4	4	4	4	4	3	3	3	3	3	3	3	3	3
F3			2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

(c) LRU: total page fault: 18

	0	1	2	3	4	5	2	4	3	2	5	4	2	5	6	1	3	2	1	0	2	4	3	2	1
F1	0	0	0	3	3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	4	4	4	1
F2		1	1	1	4	4	4	4	4	4	5	5	5	5	5	5	3	3	3	0	0	0	3	3	3
F3			2	2	2	5	5	5	3	3	3	4	4	4	6	6	6	2	2	2	2	2	2	2	2

4 memory frames

(a) FIFO: total page fault: 13

	0	1	2	3	4	5	2	4	3	2	5	4	2	5	6	1	3	2	1	0	2	4	3	2	1
F1	0	0	0	0	4	4	4	4	4	4	4	4	4	4	4	4	3	3	3	3	3	3	3	3	1
F2		1	1	1	1	5	5	5	5	5	5	5	5	5	5	5	5	2	2	2	2	2	2	2	2
F3			2	2	2	2	2	2	2	2	2	2	2	2	6	6	6	6	6	0	0	0	0	0	0
F4				3	3	3	3	3	3	3	3	3	3	3	3	1	1	1	1	1	1	4	4	4	4

(b) optimal: total page fault: 10

	0	1	2	3	4	5	2	4	3	2	5	4	2	5	6	1	3	2	1	0	2	4	3	2	1
F1	0	0	0	0	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	1
F2		1	1	1	1	5	5	5	5	5	5	5	5	5	6	1	1	1	1	0	0	0	0	0	0
F3			2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
F4				3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

(c) LRU: total page fault: 14

	0	1	2	3	4	5	2	4	3	2	5	4	2	5	6	1	3	2	1	0	2	4	3	2	1
F1	0	0	0	0	4	4	4	4	4	4	4	4	4	4	4	1	1	1	1	1	1	1	3	3	3
F2		1	1	1	1	5	5	5	5	5	5	5	5	5	5	5	5	2	2	2	2	2	2	2	2
F3			2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	4	4	4	4
F4				3	3	3	3	3	3	3	3	3	3	3	6	6	6	6	6	0	0	0	0	0	1

(d)

0 1 2 3 4 5 2 4 3 2 5 4 2 5 6 1 3 2 1 0 2 4 3 2 1

FIFO: 0 1 **3** 2 3 4 5 2 4 3 2 5 4 2 5 6 1 3 2 1 0 2 4 3 2 1

LRU: 0 1 2 3 **2** 4 5 2 4 3 2 5 4 2 5 6 1 3 2 1 0 2 4 3 2 1

(e)

FIFO and LRU: 0 1 2 3 4 5 2 4 3 2 5 4 2 5 6 1 3 2 ~~1~~ 0 2 4 3 2 1

2. Alternative Rage Replacement

KBS (3 memory frame)

P=10: page fault = 18

	2	3	4	5	2	4	3	2	5	4	2	5	6	1	3	2	1	0	2	4	3	2	1
F1	0	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
F2	1	1	4	4	4	4	3	3	3	3	3	5	5	5	5	5	5	5	5	5	5	5	1
F3	2	2	2	5	5	5	5	5	5	4	4	4	6	1	3	3	1	0	0	4	3	3	3

P=5: page fault = 16

	2	3	4	5	2	4	3	2	5	4	2	5	6	1	3	2	1	0	2	4	3	2	1
F1	0	3	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	0	0	4	3	3	3
F2	1	1	1	1	1	4	4	4	4	4	4	4	4	4	3	2	2	2	2	2	2	2	2
F3	2	2	2	2	2	2	3	2	3	2	2	2	6	1	1	1	1	1	1	1	1	1	1

KBS (4 memory frame)

P=10: page fault =13

	3	4	5	2	4	3	2	5	4	2	5	6	1	3	2	1	0	2	4	3	2	1
F1	0	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	3	3
F2	1	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1
F3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
F4	3	3	3	3	3	3	3	3	3	3	3	6	1	3	3	1	0	0	0	0	0	0

P=5: page fault = 13

	3	4	5	2	4	3	2	5	4	2	5	6	1	3	2	1	0	2	4	3	2	1
F1	0	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	3	3
F2	1	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1
F3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
F4	3	3	3	3	3	3	3	3	3	3	3	6	1	3	3	1	0	0	0	0	0	0

Compared to the LRU algorithm, KBS algorithm will has less page fault rate in this case.

3. Deadlock Avoidance

(a)

	Allocation				Need			
	A	B	C	D	A	B	C	D
P_0	1	0	1	2	1	2	1	0
P_1	2	1	0	1	1	0	1	0
P_2	1	2	0	1	1	2	3	1
P_3	2	1	1	0	1	1	0	2
P_4	1	1	1	2	1	2	1	1
P_5	2	0	2	2	1	1	0	0

A possible safe sequence:

$$1\ 1\ 1\ 0 \xrightarrow{P1:\ 2\ 1\ 0\ 1} 3\ 2\ 1\ 1 \xrightarrow{P5:\ 2\ 0\ 2\ 2} 5\ 2\ 3\ 3 \xrightarrow{P0:\ 1\ 0\ 1\ 2} 6\ 2\ 4\ 5 \xrightarrow{P2:\ 1\ 2\ 0\ 1} 7\ 4\ 4\ 6 \xrightarrow{P3:\ 2\ 1\ 1\ 0} 9\ 5\ 5\ 6 \xrightarrow{P4:1\ 1\ 1\ 2} 10\ 6\ 6\ 8$$

Available: 1 1 1 0

	Work	Allocation	Need	W + A						
P_1	1 1 1 0	2 1 0 1	1 0 1 0	3 2 1 1						
P_0	3 2 1 1	1 0 1 2	1 2 1 0	4 2 2 3						
					Choice 1	P_3	4 2 2 3	2 1 1 0	1 1 0 2	**6 3 3 3**
					Choice 2	P_4	4 2 2 3	1 1 1 2	1 2 1 1	**5 3 3 5**
					Choice 3	P_5	4 2 2 3	2 0 2 2	1 1 0 0	**6 2 4 5**

6 3 3 3 is bigger than request of left 3 processes.

P_1-P_0-P_3-(P_2, P_4, P_5) 3P3

5 3 3 5 is bigger than request of left 3 processes.

P_1-P_0-P_4-(P_2, P_3, P_5) 3P3

6 2 4 5 is bigger than request of left 3 processes.

P_1-P_0-P_5-(P_2, P_3, P_4) 3P3

	Work	Allocation	Need	W + A					
P_1	1 1 1 0	2 1 0 1	1 0 1 0	3 2 1 1					
P_4	3 2 1 1	1 1 1 2	1 2 1 1	4 3 2 3					
				Choice 1	P_0	4 3 2 3	1 0 1 2	1 2 1 0	**5 3 3 5**
				Choice 2	P_3	4 3 2 3	2 1 1 0	1 1 0 2	**6 4 3 3**
				Choice 3	P_5	4 3 2 3	2 0 2 2	1 1 0 0	**6 3 4 5**

5 3 3 5 is bigger than request of left 3 processes.

P_1-P_4-P_0-(P_2, P_3, P_5) 3P3

6 4 3 3 is bigger than request of left 3 processes.

P_1-P_4-P_3-(P_0, P_2, P_5) 3P3

6 3 4 5 is bigger than request of left 3 processes.

P_1-P_4-P_5-(P_0, P_2, P_3) 3P3

	Work	Allocation	Need	W + A
P_1	1 1 1 0	2 1 0 1	1 0 1 0	3 2 1 1
P_5	3 2 1 1	2 0 2 2	1 1 0 0	**5 2 3 3**

5 2 3 3 is bigger than request of left 4 processes.

P_1-P_5-(P_0, P_2, P_3, P_4) 4P4

	Work	Allocation	Need	W + A
P_5	1 1 1 0	2 0 2 2	1 1 0 0	3 1 3 2
P_1	3 1 3 2	2 1 0 1	1 0 1 0	**5 2 3 3**

5 2 3 3 is bigger than request of left 4 processes.

P_5-P_1-(P_0, P_2, P_3, P_4) 4P4

	Work	Allocation	Need	W + A
P_5	1 1 1 0	2 0 2 2	1 1 0 0	3 1 3 2
P_3	3 1 3 2	2 1 1 0	1 1 0 2	**5 2 4 2**

5 2 4 2 is bigger than request of left 4 processes.

P_5-P_3-(P_0, P_1, P_2, P_4) 4P4

Possible safe sequences: $3P3 \times 6 + 4P4 \times 3 = 108$

(b)

P_x is under reporting but also safe, so which will not affect the process sequence. According to the description, P_x is not P_1, so must be P_5 (because of the first process must be P_1 or P_5), when change A, still be safe. Then P_x is P_5

(c)

Y_1 will be affected the safe sequence of these processes without Y_2, P_5 is safe for (b), if B is changed, it will be unsafe.

Then P_y is P_5, and Y_1 is B.

4. Deadlock Detection

(a)

	Allocation				Request			
	A	B	C	D	A	B	C	D
P_0	1	0	1	1	0	1	1	2
P_1	2	0	0	1	1	1	1	1
P_2	1	1	0	0	1	0	1	0
P_3	2	0	1	1	1	2	1	2
P_4	1	1	1	1	0	1	1	0
P_5	2	2	2	1	1	0	1	2

Available: 1 0 1 0

	Work	Allocation	Need	Work + Allocation
P_2	1 0 1 0	1 1 0 0	1 0 1 0	2 1 1 0
P_4	2 1 1 0	1 1 1 1	0 1 1 0	3 3 3 1
P_1	3 3 3 1	2 0 0 1	1 1 1 1	**5 3 3 2**

5 3 3 2 is bigger than request of all the left processes.

P_2-P_4-P_1-(P_0, P_3, P_5) 3P3

All possible safe sequence:

P_2-P_4-P_1-P_0-P_3-P_5

P_2-P_4-P_1-P_0-P_5-P_3

P_2-P_4-P_1-P_3-P_0-P_5

P_2-P_4-P_1-P_3-P_5-P_0

P_2-P_4-P_1-P_5-P_0-P_3

P_2-P_4-P_1-P_5-P_3-P_0

Possible safe sequences: $3P3 = 6$

(b)

If P_2 increase 2, then P_2 will be unsafe (because of the first process must be P_2), so $P_x = P_2$.

(c)

If Y_2 is D, then it will be deadlock. For P_2, P_4, P_1, if D increases, first three process can be changed, so that the result would not be safe. Then P_y is P_2, P_4 or P_1.