Name: Zhu Jin Shun

ID:22101071d

COMP2432 A5

Question 1

a)

FIFO Page Replacement Algorithm with 3 Memory Frames:

Reference String: 0 1 2 3 4 5 2 4 3 2 0 4 8 1 0 2 4 5 1 2 4 3 2 0

Page Faults: 21

Content of Frames:

1st access: |0|-|-| Page fault

2nd access: |0|1|-| Page fault

3rd access: |0|1|2| Page fault

4th access: |3|1|2| Page fault

5th access: |3|4|2| Page fault

6th access: |3|4|5| Page fault

7th access: |2|4|5| Page fault

8th access: |2|4|5|

9th access: |2|3|5| Page fault

10th access: |2|3|5|

11th access: |2|3|0| Page fault

12th access: |4|3|0| Page fault

13th access: |4|8|3| Page fault

14th access: |4|8|1| Page fault

15th access: |0|8|1| Page fault

16th access: |0|2|1| Page fault

17th access: |0|2|4| Page fault

18th access: |5|2|4| Page fault

19th access: |5|1|4| Page fault

20th access: |5|1|2| Page fault

21st access: |4|1|2| Page fault

22nd access: |4|3|2| Page fault

23rd access: |4|3|2|

24th access: |4|3|0| Page fault

FIFO Page Replacement Algorithm with 4 Memory Frames:

Page Faults: 14

Content of Frames:

1st access: |0|-|-|-| Page fault

2nd access: |0|1|-|-| Page fault

3rd access: |0|1|2|-| Page fault

4th access: |0|1|2|3| Page fault

5th access: |4|1|2|3| Page fault

6th access: |4|5|2|3| Page fault

7th access: |4|5|2|3|

8th access: |4|5|2|3|

9th access: |4|5|2|3|

10th access: |4|5|2|3|

11th access: |4|5|0|3| Page fault

12th access: |4|5|0|3|

13th access: |4|5|0|8| Page fault

14th access: |1|5|0|8| Page fault

15th access: |1|5|0|8|

16th access: |1|2|0|8| Page fault

17th access: |1|2|4|8| Page fault

18th access: |1|2|4|5| Page fault

19th access: |1|2|4|5|

20th access: |1|2|4|5|

21st access: |1|2|4|5|

22nd access: |3|2|4|5| Page fault

23rd access: |3|2|4|5|

24th access: |3|0|4|5| Page fault

b)

Optimal Page Replacement Algorithm with 3 Memory Frames:

Reference String: 0 1 2 3 4 5 2 4 3 2 0 4 8 1 0 2 4 5 1 2 4 3 2 0

Page Faults: 15

Content of Frames:

1st access: |0|-|-| Page fault

2nd access: |0|1|-| Page fault

3rd access: |0|1|2| Page fault

4th access: |3|1|2| Page fault

5th access: |3|4|2| Page fault

6th access: |5|4|2| Page fault

7th access: |5|4|2|

8th access: |5|4|2|

9th access: |3|4|2| Page fault

10th access: |3|4|2|

11th access: |0|4|2| Page fault

12th access: |0|4|2| Page fault

13th access: |0|8|2| Page fault

14th access: |0|8|1| Page fault

15th access: |0|2|1| Page fault

16th access: |0|2|1|

17th access: |4|2|1| Page fault

18th access: |5|2|1| Page fault

19th access: |5|2|1| Page fault

20th access: |5|2|1| Page fault

21th access: |4|2|1| Page fault

22th access: |4|2|3| Page fault

23th access: |4|2|3| Page fault

24th access: |4|2|0| Page fault

Optimal Page Replacement Algorithm with 4 Memory Frames:

Reference String: 0 1 2 3 4 5 2 4 3 2 0 4 8 1 0 2 4 5 1 2 4 3 2 0

Page Faults: 12

Content of Frames:

1st access: |0|-|-|-| Page fault

2nd access: |0|1|-|-| Page fault

3rd access: |0|1|2|-| Page fault

4th access: |0|1|2|3| Page fault

5th access: |4|1|2|3| Page fault

6th access: |4|5|2|3| Page fault

7th access: |4|5|2|3|

8th access: |4|5|2|3|

9th access: |4|5|2|3|

10th access: |4|5|2|3|

11th access: |4|0|2|3| Page fault

12th access: |4|0|2|3|

13th access: |4|0|2|8| Page fault

14th access: |4|0|2|1| Page fault

15th access: |4|0|2|1|

16th access: |4|0|2|1|

17th access: |4|0|2|1|

18th access: |4|5|2|1| Page fault

19th access: ||4|5|2|1|

20th access: |4|5|2|1|

21st access: |4|5|2|1|

22nd access: |4|3|2|1| Page fault

23rd access: |4|3|2|1|

24th access: |4|0|2|1| Page fault

c)

LRU Page Replacement Algorithm with 3 Memory Frames:

Reference String: 0 1 2 3 4 5 2 4 3 2 0 4 8 1 0 2 4 5 1 2 4 3 2 0

Page Faults: 21

Content of Frames:

1st access: |0|-|-| Page fault

2nd access: |0|1|-| Page fault

3rd access: |0|1|2| Page fault

4th access: |3|1|2| Page fault

5th access: |3|4|2| Page fault

6th access: |3|4|5| Page fault

7th access: |2|4|5| Page fault

8th access: |2|4|5|

9th access: |2|4|3| Page fault

10th access: |2|4|3|

11th access: |2|0|3| Page fault

12th access: |2|0|4| Page fault

13th access: |8|0|4| Page fault

14th access: |8|1|4| Page fault

15th access: |8|1|0| Page fault

16th access: |2|1|0| Page fault

17th access: |2|4|0| Page fault

18th access: |2|4|5| Page fault

19th access: |1|4|5| Page fault

20th access: |1|2|5| Page fault

21th access: |1|2|4| Page fault

22th access: |3|2|4| Page fault

23th access: |3|2|4|

24th access: |3|2|0| Page fault

LRU Page Replacement Algorithm with 4 Memory Frames:

Reference String: 0 1 2 3 4 5 2 4 3 2 0 4 8 1 0 2 4 5 1 2 4 3 2 0

Page Faults: 15

Content of Frames:

1st access: |0|-|-|-| Page fault

2nd access: |0|1|-|-| Page fault

3rd access: |0|1|2||-| Page fault

4th access: |0|1|2|3| Page fault

5th access: |4|1|2|3| Page fault

6th access: |4|5|2|3| Page fault

7th access: |4|5|2|3|

8th access: |4|5|2|3|

9th access: |4|5|2|3|

10th access: |4|5|2|3|

11th access: |4|0|2|3| Page fault

12th access: |4|0|2|3|

13th access: |4|0|2|8| Page fault

14th access: |4|0|1|8| Page fault

15th access: |4|0|1|8|

16th access: |2|0|1|8| Page fault

17th access: |2|0|1|4| Page fault

18th access: |2|0|5|4| Page fault

19th access: |2|1|5|4| Page fault

20th access: |2|1|5|4|

21st access: |2|1|5|4|

22nd access: |2|1|3|4| Page fault

23rd access: |2|1|3|4|

24th access: |2|0|3|4| Page fault

d)

For FIFO, it is possible to insert one item in reference string that will instead reduce number of page faults. The string will be 0 1 2 5 3 4 5 2 4 3 2 0 4 8 1 0 2 4 5 1 2 4 3 2 0. This will change the page faults into 18.

For LRU, it is also possible. The string will be 0 1 2 3 4 5 2 4 3 2 0 4 2 8 1 0 2 4 5 1 2 4 3 2 0 . This will change the page faults into 20.

For both LRU and FIFO, it is also possible. The string will be 0 1 2 3 4 5 2 4 3 2 4 0 4 8 1 0 2 4 5 1 2 4 3 2 0.

This will change the page faults into 20.

Question 2

Applying KBS with 3 memory frames and P = 8:

Reference String: 0 1 2 3 4 5 2 4 3 2 0 4 8 1 0 2 4 5 1 2 4 3 2 0

Page Faults: 23

Content of Frames:

1st access: |0|-|-| Page fault

2nd access: |0|1|-| Page fault

3rd access: |0|1|2| Page fault

4th access: |0|1|3| Page fault

5th access: |0|1|4| Page fault

6th access: |0|1|5| Page fault

7th access: |0|1|2| Page fault

8th access: |0|1|4| Page fault

9th access: |3|1|4| Page fault

10th access: |3|2|4| Page fault

11th access: |3|0|4| Page fault

12th access: |3|0|4|

13th access: |3|8|4| Page fault

14th access: |3|1|4| Page fault

15th access: |3|0|4| Page fault

16th access: |3|0|2| Page fault

17th access: |4|0|2| Page fault

18th access: |4|0|5| Page fault

19th access: |4|0|1| Page fault

20th access: |2|0|1| Page fault

21th access: |4|0|1| Page fault

22th access: |4|0|3| Page fault

23th access: |4|2|3| Page fault

24th access: |4|2|0| Page fault

Applying KBS with 4 memory frames and P = 8:

Content of Frames:

Reference String: 0 1 2 3 4 5 2 4 3 2 0 4 8 1 0 2 4 5 1 2 4 3 2 0

Page Faults: 15

Content of Frames:

1st access: |0|-|-|-| Page fault

2nd access: |0|1|-|-| Page fault

3rd access: |0|1|2||-| Page fault

4th access: |0|1|2|3| Page fault

5th access: |0|1|2|4| Page fault

6th access: |0|1|2|5| Page fault

7th access: |0|1|2|5|

8th access: |0|1|2|4| Page fault

9th access: |3|1|2|4| Page fault

10th access: |3|1|2|4|

11th access: |3|0|2|4| Page fault

12th access: |3|0|2|4|

13th access: |3|8|2|4| Page fault

14th access: |3|1|2|4| Page fault

15th access: |3|0|2|4| Page fault

16th access: |3|0|2|4|

17th access: |3|0|2|4|

18th access: |5|0|2|4| Page fault

19th access: |1|0|2|4| Page fault

20th access: |1|0|2|4|

21st access: |1|0|2|4|

22nd access: |3|0|2|4| Page fault

23rd access: |3|0|2|4|

24th access: |3|0|2|4|

Applying KBS with 3 memory frames and P = 5:

Reference String: 0 1 2 3 4 5 2 4 3 2 0 4 8 1 0 2 4 5 1 2 4 3 2 0

Page Faults: 12

Content of Frames:

1st access: |0|-|-| Page fault

2nd access: |0|1|-| Page fault

3rd access: |0|1|2| Page fault

4th access: |0|1|3| Page fault

5th access: |0|1|4| Page fault

6th access: |5|1|4| Page fault

7th access: |5|2|4| Page fault

8th access: |5|2|4|

9th access: |5|3|4| Page fault

10th access: |5|2|4| Page fault

11th access: |0|2|4| Page fault

12th access: |0|2|4|

13th access: |0|2|8| Page fault

14th access: |0|2|1| Page fault

15th access: |0|2|1|

16th access: |0|2|1|

17th access: |0|4|1| Page fault

18th access: |0|5|1| Page fault

19th access: |0|5|1|

20th access: |2|5|1| Page fault

21th access: |4|5|1| Page fault

22th access: |3|5|1| Page fault

23th access: |3|2|1| Page fault

24th access: |3|2|0| Page fault

Applying KBS with 4 memory frames and P = 5:

Reference String: 0 1 2 3 4 5 2 4 3 2 0 4 8 1 0 2 4 5 1 2 4 3 2 0

Page Faults: 15

Content of Frames:

1st access: |0|-|-|-| Page fault

2nd access: |0|1|-|-| Page fault

3rd access: |0|1|2||-| Page fault

4th access: |0|1|2|3| Page fault

5th access: |0|1|2|4| Page fault

6th access: |5|1|2|4| Page fault

7th access: |5|1|2|4|

8th access: |5|1|2|4|

9th access: |5|3|2|4| Page fault

10th access: |5|3|2|4|

11th access: |0|3|2|4| Page fault

12th access: |0|3|2|4|

13th access: |0|3|2|8| Page fault

14th access: |0|1|2|8| Page fault

15th access: |0|1|2|8|

16th access: |0|1|2|8|

17th access: |0|1|4|8| Page fault

18th access: |0|1|4|5| Page fault

19th access: |0|1|4|5|

20th access: |2|1|4|5| Page fault

21st access: |2|1|4|5|

22nd access: |2|1|3|5| Page fault

23rd access: |2|1|3|5|

24th access: |2|0|3|5| Page fault

With 3 memory frames:

LRU: 21 page faults

KBS with P = 5: 19 page faults

KBS with P = 8: 23 page faults

With 4 memory frames:

LRU: 15 page faults

KBS with P = 5: 15 page faults

KBS with P = 8: 15 page faults

Based on these results, we can see that KBS outperforms LRU in terms of page fault performance in 3 memory frames when P=5 and LRU outperforms KBS in 3 memory frames when P=8 .

For 4 memory frames KBS and LRU performs the same.

Question 3

a)

Allocation				
	A	B	C	D

P0	2	0	1	1
P1	1	1	2	1
P2	1	0	0	2
P3	2	4	3	2
P4	0	0	1	0
P5	1	0	1	1

Need				
	A	B	C	D
P0	2	3	1	3
P1	2	3	0	2
P2	1	0	1	0
P3	2	0	0	2
P4	1	0	0	1
P5	1	3	1	1

Avail= 1 1 1 1 Req0=0 0 1 0

New Available= Avail-Req0 =1 1 0 1

Safe sequence:

First process 4:

1 1 0 1 >1 0 0 1

1 1 0 1 + 0 0 1 0=1 1 1 1

Next process 2:

1 1 1 1 >1 0 1 0

1 1 1 1 + 1 0 0 2= 3 1 1 3

Next process 3:

3 1 1 3>2 0 0 2

3 1 1 3 + 2 4 3 2=6 5 4 5

Next process 5:

6 5 4 5>1 3 1 1

6 5 4 5 + 1 0 1 1=7 5 5 6

Next process 1:

7 5 5 6 >2 3 0 2

7 5 5 6 + 1 1 2 1= 8 6 7 7

Finally process 0:

8 6 7 7> 2 3 1 0

8 6 7 7 + 2 0 1 1= 10 6 8 8

Safe sequence P4, P2 ,P3,P5,P1,P0

The number of possible safe sequences is 6

b)

X can be 0, 1, 5

X=0

There are correct needs 3 3 1 3, 2 4 1 3, 2 3 2 3, 2 3 1 4

X=1:

There are correct needs 3 3 0 2, 2 4 0 2, 2 3 1 2, 2 3 0 3

X=2:

No safe sequence because 2 0 1 0 or 1 0 2 0 can’t be satisfied with 1 1 0 1.

X=3:

No safe sequence because 3 0 0 2 can’t be satisfied with 1 1 0 1.

X=4:

No safe sequence because 1 0 1 1 can’t be satisfied with 1 1 0 1.

X=5:

There are correct needs 2 3 1 1, 1 4 1 1, 1 3 2 1, 1 3 1 2

c)

Y can only be B.

Because when Y=A, as P2 is 2 0 1 0 there will be no safe sequence for this question because P2 can’t be satisfied with 1 at A because 1 1 0 1+0 0 1 0=1 1 1 1.

When Y=C, as P4 is 1 0 1 1, there will be no safe sequence for this problem, because P4 can’t be satisfied with 1 at C with 1 1 0 1.

When Y=D, as P4 is 1 0 0 2, there will be no safe sequence for this problem, because P4 can’t be satisfied with 2 at D with 1 1 0 1.

Question 4

a)

Allocation				
	A	B	C	D
P0	2	0	1	1
P1	1	2	1	0
P2	1	0	0	2
P3	0	1	0	0
P4	1	0	1	1
P5	2	0	1	1

Avail= 1 1 0 1

First process 1 :

1 1 0 1>1 0 0 1

1 1 0 1+1 2 1 0=2 3 1 1

Second process 0:

2 3 1 1> 1 0 1 1

2 3 1 1+2 0 1 1=4 3 2 2

Third process 3:

4 3 2 2+ 0 1 0 0=4 4 2 2

4 4 2 2<1 2 3 4

4 4 2 2<0 1 2 3

4 4 2 2<2 4 3 2

So, process 2,4,5 are involved in deadlock.

b)

x is 5 and X is C.

Because the requirement now for P5 will be 2 4 2 2 which is lower than 4 4 2 2. So process 5 can be continue executed and then available will become 6 4 3 3, then process 2 and finally process 4 resulting in a safe sequence and no deadlock.

c)

For the new process y where it makes a new request for an instance of A, to let the system enters a deadlocked state, the request of new A of y will be 1. Because when y=1 then 1 1 0 1 can't satisfy 2 0 0 1 leading to a deadlock.