COMP2432 Operating Systems
Assignment 5

Wang Ruijie 22103808D

April 21, 2024

Question 1

(a) The whole process of page replacement can be described as follows: 0, 01, 012, 123, 234, 345, 452, 452, 523, 523, 230, 304, 048, 481, 810, 102, 024, 245, 451, 512, 124, 243, 243, 430. There are 21 page faults in total for FIFO.

(b) The content of each frame after each page reference is as follows: 0, 01, 012, 032, 432, 452, 452, 452, 432, 432, 402, 402, 802, 102, 102, 102, 142, 152, 152, 152, 452, 352, 352, 052. There are 15 page faults in total for Optimal.

(c) The content of each frame after each page reference is as follows: 0, 01, 012, 312, 342, 345, 245, 245, 243, 243, 203, 204, 804, 814, 810, 210, 240, 245, 145, 125, 124, 324, 324, 320. The content of the stack after each page reference is as follows: 0, 10, 210, 321, 432, 543, 254, 425, 342, 234, 023, 402, 840, 184, 018, 201, 420, 542, 154, 215, 421, 342, 234, 023. There are 21 page faults in total for LRU with stack.

(a), (b), (c) with 4 Frames If the page replacement is with 4 memory frames, the content of each frame after each page reference under FIFO is as follows: 0, 01, 012, 0123, 1234, 2345, 2345, 2345, 2345, 2345, 3450, 3450, 4508, 5081, 5081, 0812, 8124, 1245, 1245, 1245, 1245, 2453, 2453, 4530. There are 14 page faults.

The contents under Optimal are as follows: 0, 01, 012, 0123, 0423, 5423, 5423, 5423, 5423, 5423, 5420, 5420, 8420, 1420, 1420, 1420, 1420, 1425, 1425, 1425, 1425, 3425, 3425, 0425. There are 12 page faults.

The contents under LRU are as follows: 0, 01, 012, 0123, 4123, 4523, 4523, 4523, 4523, 4523, 4023, 4023, 4028, 4018, 4018, 2018, 2014, 2054, 2154, 2154, 2154, 2154, 2134, 2034. The contents of the stack are as follows: 0, 10, 210, 3210, 4321, 5432, 2543, 4253, 3425, 2345, 0234, 4023, 8402, 1840, 0184, 2018, 4201, 5420, 1542, 2154, 4215, 3421, 2341, 0234. There are 15 page faults.

(d) For FIFO, by inserting an additional reference 5 at the 16th position (assume that the reference is 0-indexed), the page faults can be reduced to 20. The reference string is: 0 1 2 3 4 5 2 4 3 2 0 4 8 1 0 2 5 4 5 1 2 4 3 2 0.

For LRU, by inserting an additional reference 2 at the 18th position (assume that the reference is 0-indexed), the page faults can be reduced to 20. The reference string is: 0 1 2 3 4 5 2 4 3 2 0 4 8 1 0 2 4 5 2 1 2 4 3 2 0.

For both FIFO and LRU, by inserting an additional reference 4 at the 10th position (assume that the reference is 0-indexed), the page faults can be reduced to 20. The reference string is: 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.

Question 2 The future conditions for p=8 and p=5 are as follows:

Reference	Future of p=8	Future of p=5
0		
1	0	0
2	01	01
3	012	012
4	0123	0123
5	01234	01234
2	012345	12345, 0
4	0123452	23452, 01
3	01234524	34524, 012
2	12345243, 0	45243, 0123
0	23452432, 01	52432, 01234
4	34524320, 012	24320, 12345, 0
8	45243204, 0123	43204, 23452, 01
1	52432048, 01234	32048, 34524, 012
0	24320481, 012345	20481, 45243, 0123
2	43204810, 0123452	04810, 52432, 01234
4	32048102, 01234524	48102, 24320, 12345, 0
5	20481024, 12345243, 0	81024, 43204, 23452, 01
1	04810245, 23452432, 01	10245, 32048, 34524, 012
2	48102451, 34524320, 012	02451, 20481, 45243, 0123
4	81024512, 45243204, 0123	24512, 04810, 52432, 01234
3	10245124, 52432048, 01234	45124, 48102, 24320, 12345, 0
2	02451243, 24320481, 012345	51243, 81024, 43204, 23452, 01
0	24512432, 43204810, 0123452	12432, 10245, 32048, 34524, 012

If the frame size is 3 and p = 8, the content of each frame is as follows: 0, 01, 012, 013, 014, 015, 012, 014, 013, 213, 203, 243, 248, 241, 240, 240, 240, 250, 210, 210, 410, 310, 210, 210. The number of page faults is 20.

If the frame size is 3 and p = 5, the content of each frame is as follows: 0, 01, 012, 013, 014, 015, 215, 245, 345, 245, 045, 045, 048, 041, 041, 042, 042, 052, 012, 012, 412, 413, 412, 012. The number of page faults is 20.

If the frame size is 4 and p = 8, the content of each frame is as follows: 0, 01, 012, 0123, 0124, 0125, 0125, 0124, 0123, 0123, 0123, 0423, 8423, 1423, 0423, 0423, 0423, 0425, 0421, 0421, 0421, 0321, 0321, 0321. The number of page faults is 15.

If the frame size is 4 and p = 5, the content of each frame is as follows: 0, 01, 012, 0123, 0124, 0125, 0125, 0425, 3425, 3425, 0425, 0425, 0428, 0421, 0421, 0421, 0421, 0521, 0521, 0521, 4521, 4531, 4521, 4021. The number of page faults is 16.

Performance Comparison There is no obvious difference in the number of page faults between the two algorithms. If the frame size is 3, the LRU algorithm is slightly better, while if the frame size is 4, the KBS is slightly better. Meanwhile, the performance, or the number of page faults is also dependent on the value of p. However, since LRU is more easy to implement in the real-world scenarios, it is more likely to be used in practice. The advantage of KBS is not notable enough in this case, and more experiments are needed to determine the better algorithm.

Question 3

(a) The amount of each resource requested by Req_0 is less than or equal to the amount of each resource currently held. Therefore, Req_0 can be granted. After Req_0 is granted, the resources owned by the system are

1, 1, 0, 1, and P$_4$, P$_2$, and P$_3$ can be granted in order. At this moment, the resource held by the operating system are 4, 5, 4, 5. Thus, all other requests are able to be handled without causing a deadlock.

Since the requests of P$_0$, P$_1$ and P$_5$ can be granted in an arbitrary order, a possible safe sequence can be P$_4$, P$_2$, P$_3$, P$_0$, P$_1$, P$_5$, and the number of all possible safe sequences is $A_3^2 = 6$.

(b) There are the following possible conditions regarding the identities of P$_x$:

1. If P_x is P$_0$, P$_1$, or P$_5$, there is no deadlock, since after the requests of P$_4$, P$_2$, and P$_3$ are granted, the resources held by the operating system is 4, 5, 4, 5, which is enough for all other requests even with any under-reporting of any instance. This is because that the maximum number of any kind of instance initially requested by P$_0$, P$_1$, and P$_5$ is 3. If the under-reporting is considered, the maximum number will be 4, which is less that or equal to any number of instances held by the operating system.
2. If P_x is P$_2$ and the under-reported instance is A, the system will be in a deadlock. In this case, the number of instances of A requested is 2, but the operating system is unable to grant any other request to obtain more instances of A, i.e., the number of instance A held by the operating system is always 1. Therefore, the request can not be granted, the system is in a deadlock.
3. If P_x is P$_3$ and the under-reported instance is A, the system will be in a deadlock. In this case, after the requests of P$_4$ and P$_2$ are granted, the number of instances A held by the operating system is 2, which is not enough for the requests of P$_3$. No more requests can be granted, and the system is in a deadlock.
4. If P_x is P$_4$ and the under-reported instance is A, the system will be in a deadlock. Prior to P$_4$, the operating system can not grant any other request. If P$_4$ requests for 2 instances of A, the system will be in a deadlock.

In conclusion, P_x can only be P$_0$, P$_1$, or P$_5$ to avoid a deadlock.

(c) There are the following possible conditions regarding the identities of Y:

1. If Y is A and P$_y$ is P$_2$, P$_3$, or P$_4$, the system will be in a deadlock as discussed in (b).
2. If Y is B, the system will not be in a deadlock. In this case, if P$_y$ is P$_2$, P$_3$, or P$_4$, since the number of instance B requested is all 0, their requests can be granted even if they under-reports the number of instance B by 1. If P$_y$ is P$_0$, P$_1$ or P$_5$, after the requests of P$_4$, P$_2$ and P$_3$ are granted, the resources owned by the system are 4, 5, 4, 5, which is enough for all other requests even if considering any under-reporting instance B by 1.
3. If Y is C and P$_y$ is P$_4$, the system will be in a deadlock. In this case, the number of instance C requested is 1, but the operating system can not grant any other request to obtain more instances of C, and it holds no instance of C. Therefore, the request can not be granted, and the system is in a deadlock.
4. If Y is D, the system will be in a deadlock. If P$_y$ is P$_4$, the number of instance D requested is 2. Similarly, the operating system can not grant any other request to obtain more instances of D, and it holds only 1 instance of D. Therefore, the request can not be granted, and the system is in a deadlock.

Therefore, the only reason for Y is B.

Question 4

(a) At first, only P_1's request can be granted, and afterwards, resources owned by the system are 2,3,1,1. Then, only P_0's request can be granted, and then only that of P_3 can be granted. At that time, resources owned by the system are 4, 4, 2, 2, and no more request of the processes can be granted, since the number of instance D owned by the operating system is less than the requests of instance D by P_2 and P_4, and the number of instance C owned by the operating system is less than the requests of instance C by P_5. Therefore, the system is in a deadlock, and the processes in the deadlock are P_2, P_4, and P_5.

(b) X should be C, and P_x should be P_5. If P_x is P_2, X must be D, and after the system satisfies P_2's request, the resources owned by the system are 4, 4, 2, 4, and no more request can be granted by the system. If P_x is P_4, there is no X such that it can make P_4's request able to be granted. If P_x is P_5, X must be C, and after the system satisfies P_5's request, the resources owned by the system are 5, 4, 3, 3, and request of P_2, and then that of P_4 can be granted.

(c) If without the new request for an instance of A, the sequence of request handling must begin with P_1, and then P_0, as mentioned in (a). After the requests of P_1 and P_0 are granted, the resources owned by the system are 4, 3, 2, 2. If P_y is one of P_2, P_3, P_4 and P_5, a new request for one more instance of A can always be granted without causing a deadlock, since the number of instances of A held by the operating system is 4, which is 2 more than the maximum number of A in each request. If P_y is P_0, the new request for one more instance of A can also be granted. Therefore, P_y can only be P_1.