Worksheet 12

2/12/2025

9/9 Points

Attempt 1

Review Feedback
2/12/2025

Attempt 1 Score:
9/9

Add Comment

Anonymous Grading: **no**

Unlimited Attempts Allowed

Details

Q1: The next page access will cause a page fault as page 5 is not in memory. Using the optimal replacement algorithm, which frame will be removed from memory to allow page 5 to take its place?

Q2: Considering the following page access history and frame content. Which frame would be removed when FIFO replacement is used? Which frame is removed when using Least Recently Used (LRU)?

Q3: What would be the complications of using a timestamp approach to find the least recently used frame?

Answer1:
We will remove **page 4** (1, 2, 4) with the new entry page 5; making the frame contents 1, 2, 5 (**Optimal Replacement Algorithm**)

Given page access history,

✗	✗	✗	✗	✓	✓						
1	2	3	4	1	2	5	1	2	3	4	5

↑

Given the frame contents for the 3 available frames (1, 2, 4), we want to replace one from these three and add 5.
According to the "**Optimal Replacement Algorithm**", we will look into the future access right after 5, which are: 1 , 2 , 3 , 4 , 5.
We will be accessing 1 and 2 both right after 5 so replacing 1 or 2 will cause us extra page fault which we would like to avoid. So, we would remove page 4 (before checking with 5, frame contents are : 1, 2, 4) and add 5 instead. (after checking 5, new frame contents : 1, 2, 5)

Answer2:
Given page access history and frame content:

✗	✗	✗	✗	✗	✗	✗	✓	✓	✗		
1	2	3	4	1	2	5	1	2	3	4	5

In the **FIFO** approach: **page 1** will be replaced.
As FIFO means "first in first out", page 1 was in first, among 1, 2, 5 ; so if we want to add 3, we will remove 1 first and then add 3.

In the **Least Recently Used (LRU)** approach: **page 5** will be replaced.
Among 1, 2, 5 ; pages 1 and 2 have been recently accessed before 3, making 5 the least recently used among them.

Answer3:
Complications of using a timestamp approach to find the least recently used frame:

- Large storage is required for storing the timestamp values
- Inefficient to update the timestamp values every time a frame is accessed, lots of overhead
- Expensive lookup of the timestamp values to check when the page was accessed

New Attempt