

Assignment 2

Operating Systems

Question 1:

Suppose a process P is currently executed on a CPU and process references five pages 1, 2, 3, 4, and 5 in the following order:

1, 3, 2, 4, 1, 3, 5, 1, 3, 2, 4, 5, 4, 2

The main memory has three frames and initially following pages have been loaded into the frames:

Frame 1	1
Frame 2	3
Frame 3	2

- If most frequently used page replacement algorithm is used, show how pages will be replaced upon each new page reference.
- Calculate hit-ratio for the above scenario.

Note: Once all frames are full, the next page reference is to be made by replacing one of the page currently residing in frames. A newly referenced page does not need to be loaded again if it is already in a frame.

Question 2:

Consider the following two processes P1 and P2 are being executed concurrently and are sharing binary semaphores 'S' and 'W'. Both of the semaphores are initialized to 1.

<pre>void P1 () { while (true) { wait(S); wait(W);</pre>	<pre>void P2 () { while (true) { wait(W); wait(S);</pre>
---	---

<pre>critical region; signal (S); signal (W); } }</pre>	<pre>critical region; signal (W); signal (S); } }</pre>
---	---

- Will the concurrent execution of both of the above process result in an infinite wait?
- Make changes in the above code so the two processes never enter into an infinite waiting state without violating the mutually exclusive entry to the critical region.
(**Hint:** Changing order of wait on semaphores will not work here. Think about turn variable and remember, turn alone creates strict alternation)