CSCE 5640: Operating System Design

Homework-4

Name: Kishan Kumar Zalavadia

EUID: 11685261

---

1. (10) Consider a paging system with the page table stored in memory.

a. If a memory reference takes 60 nanoseconds, how long does a paged memory reference take?

b. If we add TLBs, and if 80 percent of all page-table references are found in the TLBs, what is the effective memory reference time? (Assume that finding a page-table entry in the TLBs takes 3 nanoseconds if the entry is present.)

In answering the above two questions, briefly justify your answers.

Ans:

a. Given that memory reference takes 60 nanoseconds.
In the paging system, every memory reference needs two memory access, one to read the page table and another to read the actual memory location.
Total time = 60ns+60ns = 120 nanoseconds.

b. With TLB, we have two cases, hit and miss.
Hit: Given 80% cases
    Total time for TLB hit = 3ns + 60ns
Miss: 100%-80% = 20%
    Total time for TLB miss = 3ns + 60ns + 60ns = 123ns
Effective memory reference time = (Hit rate * Hit time) + (Miss rate + Miss time)
= (0.8 * 62ns) + (0.2 * 123ns)
= 50.4+24.6
= 75 nanoseconds

---

3. (20) Assume that memory contains three holes of 10 MB each. A sequence of 14 requests for 1MB each is to be processed. For each of the memory allocation methods listed below, determine the sizes of each of the remaining holes after all 14 requests have been satisfied (assume that the direction of searching the holes is from top to bottom or left to right):

a. first fit

b. best fit

c. worst fit

Ans:

Initially, we have three holes, each 10MB in size.

Holes: [10, 10, 10]

Total holes size = 30MD

There are 14 requests, each of size 1MB.

Total request size = 14MB

a. First fit means allocating memory to the top hole, which can take the request.
   Since we fill the holes from top to bottom, we fill the top hole with ten requests.
   The next hole will be filled by 4MB because we have 4MB requests left. This reduces the 2nd hole size to 10-4 = 6MB.
   Holes memory remaining: [0, 6, 10]

b. Best fit allocates the memory to the smallest hole that can take the request.
   Initially, all holes are the same in size, so the 1st request goes to hole 1. Now, the remaining size of hole 1 is 9MB.
   Hole 1 is smaller now, so all the upcoming requests will be taken by Hole 1 until it is filled. When hole one is filled, there will be ten completed requests.
   Now, the 11th request goes to the next smallest hole that can take it, which is hole 2. [Holes 2 and 3 are small, so they take from the top]. All remaining requests, which are four requests, go to hole 2.
   Holes memory remaining: [0, 6, 10]

c. The worst case tries to leave big holes, which means the request is allocated to the largest hole possible.
   Initial holes remaining size: [10, 10, 10]
   Request 1 given to hole 1; Holes remaining size: [9, 10, 10]
   Request 2 given to hole 2; Holes remaining size: [9, 9, 10]
   Request 3 given to hole 3; Holes remaining size: [9, 9, 9]
   Request 4 given to hole 1; Holes remaining size: [8, 9, 9]
   Request 5 given to hole 2; Holes remaining size: [8, 8, 9]
   Request 6 given to hole 3; Holes remaining size: [8, 8, 8]
   Request 7 given to hole 1; Holes remaining size: [7, 8, 8]
   Request 8 given to hole 2; Holes remaining size: [7, 7, 8]
   Request 9 given to hole 3; Holes remaining size: [7, 7, 7]
   Request 10 given to hole 1; Holes remaining size: [6, 7, 7]
   Request 11 given to hole 2; Holes remaining size: [6, 6, 7]
   Request 12 given to hole 3; Holes remaining size: [6, 6, 6]
   Request 13 given to hole 1; Holes remaining size: [5, 6, 6]
   Request 14 given to hole 2; Holes remaining size: [5, 5, 6]
   After all the requests are allocated, the remaining sizes of holes are [5, 5, 6].

4. **(10) Consider the working set model with Δ=3. Given the following reference string of a process p:**
   y x x x x x x y y u x x x y z y z w w z x x w w
   a. **What is the largest working set process p will ever have?**
   b. **What is the smallest working set process p will ever have (not counting the first Δ references)?**

Ans:

Given Δ=3

We calculate all possible working sets.

3 Recent References	Working Set
y, x, x	{y, x}
x, x, x	{x}
x, x, x	{x}
x, x, x	{x}
x, x, y	{x, y}
x, y, y	{x, y}
y, y, u	{y, u}
y, u, x	{y, u, x}
u, x, x	{u, x}
x, x, x	{x}
x, x, y	{x, y}
x, y, z	{x, y, z}
y, z, y	{y, z}
z, y, z	{z, y}
y, z, w	{y, z, w}
z, w, w	{z, w}
w, w, z	{w, z}
w, z, x	{w, z, x}
z, x, x	{z, x}
x, x, w	{x, w}
x, w, w	{x, w}

a. Largest Working set = ({y, u, x}, {x, y, z}, {y, z, w}, {w, z, x})
   The Target working set size is three references.
b. Smallest Working set excluding the first Δ references = {(x)- multiple times}
   The smallest working set size is one reference.

5. (30) Consider the following page reference string:

3, 1, 4, 2, 5, 4, 1, 3, 5, 2, 0, 1, 1, 0, 2, 3, 4, 5, 0, 1

Assuming demand paging with three frames, how many page faults would occur for the following replacement algorithms? (Show your results similar to Figure 10.12, 10.14, 10.15)

a. FIFO replacement

b. Optimal replacement

c. LRU replacement

Ans:

a. FIFO replacement:

Reference:	3	1	4	2	5	4	1	3	5	2	0	1	1	0	2	3	4	5	0	1
Frame 1:	3	3	3	2	2		2	3		3	3	1				1	1	5	5	5
Frame 2:	-	1	1	1	5		5	5		2	2	2				3	3	3	0	0
Frame 3:	-	-	4	4	4		1	1		1	0	0				0	4	4	4	1
Fault:	Y	Y	Y	Y	Y	N	Y	Y	N	Y	Y	Y	N	N	N	Y	Y	Y	Y	Y

Fault ‘Y’ means yes there is a page fault and ‘N’ means no page fault.

Number of page faults = 15

b. Optional replacement

Reference:	3	1	4	2	5	4	1	3	5	2	0	1	1	0	2	3	4	5	0	1
Frame 1:	3	3	3	2	5			5		2	2					3	4	5		
Frame 2:	-	1	1	1	1			1		1	1					1	1	1		
Frame 3:	-	-	4	4	4			3		3	0					0	0	0		
Fault:	Y	Y	Y	Y	Y	N	N	Y	N	Y	Y	N	N	N	N	Y	Y	Y	N	N

Number of page faults: 11

c. LRU replacement

Reference:	3	1	4	2	5	4	1	3	5	2	0	1	1	0	2	3	4	5	0	1
Frame 1:	3	3	3	2	2		1	1	1	2	2	2				2	2	5	5	5
Frame 2:	-	1	1	1	5		5	3	3	3	0	0				0	4	4	4	1
Frame 3:	-	-	4	4	4		4	4	5	5	5	1				3	3	3	0	0
Fault:	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y	Y

Number of page faults: 16