# **Luke Pepin - Homework 2**

# **Released: Feb 16, 2024**

# **Due: Feb 28, 2024, 11:59pm**

**Problem 1 (5 points):** System schedulable

RMS algorithm utilization bound, n = 4, U = n(2^(1/n)-1), U = 0.756

U = (10/40) + (10/30) + (15/80) + (5/160) = 0.802 > 0.756

The system is not schedulable under Rate Monotonic Scheduling (RMS).

**Problem 2 (2 +2 + 3 points):** Virtual Addressing

A. Entires in Page Table

Entries (Number of Pages) = (virtual address space)/(page size) = (2^38)/(2^13) = **2^25**

B. Bits needed to index Page Table

Number of Pages = 2^25 -> Bits needed to index Page Table = **25**

C. Translate Virtual Addresses:

13025 -> 0001 |Offset| 1 0010 1110 0001 -> VP = 1, causes miss

75816 -> 1001 |Offset| 0 1000 0010 1000 -> VP = 9, causes miss

6500 -> 0000 |Offset| 1 1001 0110 0100 -> VP = 0, Hit, Physical Page = 13

Bits needed for physical address: (Physical Memory Size)/ (Page Size) = (2^30)/ (2^13) = 2^17->17 Bits

Outgoing Physical Address: 1 1011 1001 0110 0100, w/ Offset outlined 1101 |Offset| 1 1001 0110 0100

**Problem 3 (2 + 2 points):** Assume the following...

A: Entries in the hash table

Entries in the hash table = (Total Physical memory)/ (Page Size) = 1GB/4MB = (2^30)/(2^22) = 2^8

B: For Scenario, draw inverted page table

Inverted Page Table:

Hash Value| Linked List

0 | None

1 | None

... | ...

4 | None

5 | A -> 5 -> 2, B -> 261 -> 3

... | ...

50 | A -> 50 -> 6, B ->306->9

... | ...

255 | None

**Problem 4 (2 + 2 points):** Keep Track of Free Memory

1. How many Bytes of storage are required for using a linked list?

Total Storage = Number of nodes \* Storage per node

Number of nodes = Memory Size/ Segment hole size = 64 MB / 64KB = 1024

Storage per node = memory address + length + next node filed = 4 + 2 + 2 = 8

Total Storage = 1024 \* 8 = 8192 bytes

1. How many Bytes of storage are required using bitmap?

Total Storage = Total bits required / 8

Total bits required = Number of 64 KB blocks

Number of 64 KB blocks = Total Memory / Size of each block

Total memory = 64MB = 64 \* 1024 \*1024 bytes

Size of each block = 64 \* 1024

Total Storage = 1024/8 = 128 bytes

**Problem 5 (5 points):** TLB hit rate

P = TLB hit rate, 1-p = TLB miss rate

Mean overhead = p \* TLB hit overhead + (1 – p) \* TLB miss overhead

3 = p \* 1 + (1-p) \* 5

p = 2/4 = 0.5

TLB hit rate needed to reduce the mean overhead to 3nsec is 50%

**Problem 6 (3 points):** Clock ticks

1. Which page will the second-chance algorithm replace if needed?

Start from the page that was loaded first, check the reference bit of each page in order, If the reference bit is set clear it and move the page to the end of the queue (second chance), If the reference bit is not set replace that page.

Start: Page 0 (R = 1) moves to end of queue, Page 1 is replaced (R = 0), Page 2 is replaced (R = 0) and Page 3 (R =1 moves to the end of the queue.

Tiebreaker between Page 1 and Page 2 is the clock ticks since loaded and Page 1 was loaded first.

Page 1 will be replaced if needed.

1. Which page will the NRU (Not Recently Used) algorithm replace if needed?

Categorizes pages based on the R and M bit in classes and selects the page from the lowest non-empty class.

Classes:

Class 0 R = 0 M = 0, Page 2

Class 1 R = 0 M = 1, Page 1

Class 2 R = 1 M = 0, Page 0

Class 3 R = 1 M = 1, Page 3

As a result, Page 2 will be replaced.

1. Which page will the FIFO algorithm replace if needed?

Replaces the page that has been in memory the longest

Order of loading: Page 0, Page 1, Page 2, Page 3

As a result, page 0 will be replaced as it was loaded first.

For each algorithm, mention why which page is selected. (6 points) (included in above answers)