

### CIS\*3110 Suggested Exercises #1b

These suggested exercises are not to be handed in for marking, but should be used as a study aid. Solutions will be posted **online**.

1) For the attached map of main memory and paging disk, create a page table for process P1. Then use this page table to show what happens when P1 attempts to access each of the following virtual addresses. (Do each case separately, *ie.*; do not assume changes caused by a) are there for b), c),...) The target architecture uses pure demand paging with a page size of 2048bytes, which implies a 11 bit page offset.

- a) Read virtual address 0x2804
- b) Modify virtual address 0x2244
- c) Modify virtual address 0x1911
- d) Read virtual address 0x1403
- e) Modify virtual address 0x4f00
- f) Read virtual address 0x5010

2) Given a TLB access time of 2 cycles and a memory (RAM) access time of 40 cycles, describe the fraction of time a TLB access takes, relative to the standard access if the TLB hit rate is:

- a) 0.8
- b) 0.95
- c) 0.2

3) How slow must the TLB be before no improvement is seen, if the TLB hit rate is 0.25 and a hardware memory access takes 25 cycles?

4) The following page reference strings were collected by a hardware monitor for a program running on an Acme computer system. (The Acme system is known to support a pure demand paging virtual memory subsystem.)

- i) 0x7 0x6 0x7 0x8 0x9 0x7 ^ 0x8 0x7 0x8 0xa 0x8 ^ 0x8 0xa 0xb 0xb 0xc ^ 0x8 0x8 0xc 0xa 0xc ^ 0xa 0x7
- ii) 0x1 0x1 0x1 0x1 0x1 ^ 0x1 0x1 0x2 0x3 0x4 ^ 0x5 0x2 0x3 0x4 0x5 ^ 0x2 0x3 0x4 0x5 0x2 ^ 0x3 0x4 0x5 0x2 0x3 ^ 0x4 0x5
- iii) 0x1 0x2 0x3 0x4 0x5 0x6 0x1 0x2 ^ 0x3 0x4 0x5 0x6 0x8 0x9 0xa ^ 0x8 0x9 0xa 0x8 0x9 0xa 0x1 ^ 0x2 0x3 0x4 0x5 0x6

The tildes (^) indicate where the operating system cleared the Use bit on all frames in memory.

For a fixed page frame allocation of 3, simulate each of the following page replacement algorithms and count the number of page faults that occurs:

- FIFO
- LRU
- LFU
- Belady's optimal replacement
- Use Bit (falling back to FIFO)

5) What are the tradeoffs between using a small vs large page size in paged virtual memory systems?

6) If a machine that supports paged virtual memory is observed to be doing a lot of swapping, what change(s) should be made to the machine, in order to improve performance and why?

# Page Table

	V	RO	D	U	Page Frame	Disk Addr
0x0						
0x1						
0x2						
0x3						
0x4						
0x5						
0x6						
0x7						
0x8						
0x9						
0xA						
0xB						
0xC						
0xD						
0xE						
0xF						

# Main Memory

0x00	<b>P0-1 (RO)</b>	<b>P1-6</b>
0x02	<b>P5-4</b>	<b>P5-0</b>
0x04	<b>P1-2 (RO)</b>	<b>P1-5</b>
0x06	<b>P0-4</b>	<b>P5-F</b>
0x08	<b>P0-C</b>	<b>P2-1</b>
0x0A	<b>P5-1</b>	<b>PA-A</b>
0x0C	<b>P0-A</b>	<b>P1-3 (RO)</b>
0x0E	<b>PA-0</b>	
0x10	<b>P1-4</b>	<b>P1-E</b>
0x12	<b>P2-2</b>	<b>P4-2</b>
0x14	<b>P0-6</b>	<b>P2-0</b>
0x16	<b>P2-3</b>	<b>P0-2 (RO)</b>

