University of Calgary

CPSC 457: Principles of Operating System, Winter 2018

Assignment 5

For
Coskun Sahin, Dr.Pavol Federl

By
BenKun Chen
30005337
benkun.chen@ucalgary.ca
Tutorial Section: T02

Q1 – Written question (5 marks)

Execute sequence: P1(212KB), P2(417KB), P3(112KB), P4(426KB)

First-fit:

Free 100KB	P10 30KB	P1 212KB	P3 112KB	Free 176KB	P11 30KB	Free 200KB	P12 30KB	Free 300KB	P13 30KB	P2 417KB	Free 183KB

After P1, P2 and P3 been executed, the process P4 can't be placed due to fragmentation.

Best-fit:

Free 100KB	P10 30KB	P2 417KB	Free 83KB	P11 30KB	P3 112KB	Free 88KB	P12 30KB	P1 212KB	Free 88KB	P13 30KB	P4 426KB	Free 174KB

P1, P2, P3, P4 have been executed successfully.

Worst-fit:

Free 100KB	P10 30KB	P2 417KB	Free 83KB	P11 30KB	Free 200KB	P12 30KB	Free 300KB	P13 30KB	P1 212KB	P3 112KB	Free 276KB

After P1, P2 and P3 been executed, the process P4 can't be placed due to fragmentation.

Next-fit:

Free 100KB	P10 30KB	P1 212KB	Free 288KB	P11 30KB	Free 200KB	P12 30KB	Free 300KB	P13 30KB	P2 417KB	P3 112KB	Free 71KB

After P1, P2 and P3 been executed, the process P4 can't be placed due to fragmentation.

Q2 – Written question (5 marks)

Consider a system with 1KB (1024 bytes) page size. What are the page numbers and offsets for the following addresses?

Since page size = 1024 bytes = 2^{10} bytes. So that, offset = 10 bit

Address	Page number	Offset
2375	10b	0101000111b
19366	10010b	1110100110b
30000	11101b	0100110000b
256	0b	100000000b
16385	10000b	0000000001b

Q3 – Written question (5 marks)

Consider a system with a 32-bit logical address space and 4KB page size. The system supports up to 512MB of physical memory. How many entries are there in each of the following?

a) A conventional single-level page table.

$$Entries = \frac{logical\ address\ space}{page\ size} = \frac{2^{32}}{2^{2} * 2^{10}} = 1048576\ entries$$

b) An inverted page table.

$$Entries = \frac{physical\ memory\ space}{page\ size} = \frac{512MB}{4KB} = \frac{2^{9} * 2^{20}}{2^{12}} = 131072\ entries$$

Q4 – Written question (5 marks)

Consider a system where a direct memory reference takes 200ns.

a) If we add a single-level page table stored in memory to this system, how much time would it take to locate and reference a page in memory?

Since we know the direct memory reference takes 200 ns, we can conclude that the memory access takes 200 ns. In this case, every instruction access requires at least 2 memory access, which are for page table lookup and fetching. So that, the time which takes to locate and reference a page in memory is $2 * 200ns = 400ns$.

b) If we also add a TLB, and 75% of all page-table references are found in the TLB, what is the effective access time ? Assume that searching TLB takes 10ns.

Since the equation to calculate effective memory-access time for TLB equals to $(1 - p) * (tlbs + 2 * ma) + p * (tlbs + ma)$. So that,

$$\begin{aligned} effective\ access\ time &= (1 - p) * (tlbs + 2 * ma) + p * (tlbs + ma) \\ &= (1 - 0.75) * (10ns + 2 * 200ns) + 0.75 * (10ns + 200ns) \\ &= 260ns \end{aligned}$$

Q5 – Written question (5 marks)

Consider the following page reference string:

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

Assume there are 3 frames in the physical memory and all frames are initially empty. Illustrate how pages are placed into the frames according to the LRU and the optimal replacement algorithms. How many page faults would occur for each algorithm?

LRU:

1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
1	1	1	4		4	5	5	5	1		1	7	7		2	2			2
	2	2	2		2	2	6	6	6		3	3	3		3	3			3
		3	3		1	1	1	2	2		2	2	6		6	1			6

15 page faults

Optimal replacement algorithm:

1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
1	1	1	1			1	1				3	3			3	3			6
	2	2	2			2	2				2	7			2	2			2
		3	4			5	6				6	6			6	1			1

11 page faults

Q6 - Programming question (20 marks)

Please see attached file “pagesim.cpp”.

Q7 - Programming question (30 marks)

Please see attached file “impl.cpp”.