Marc-Andre Fichtel - 30014709
CPSC 457 - Fall 2017

Assignment 5

Due Friday, Dec 8, 2017 at 11:59pm.
Individual assignment. No group work allowed.
Weight: 8% of the final grade.

Question 1:

The following are given.

100 KB free	30 KB P_{10}	500 KB free	30 KB P_{11}	200 KB free	30 KB P_{12}	300 KB free	30 KB P_{13}	600 KB free

- P1: 212 KB
- P2: 417 KB
- P3: 112 KB
- P4: 426 KB

These processes will be placed in memory using 4 different algorithms as follows.

First-Fit

100 KB free	30 KB P_{10}	212 KB P_1	112 KP P_3	176 KB free	30 KB P_{11}	200 KB free	30 KB P_{12}	300 KB free	30 KB P_{13}	417 KB P_2	183 KB free

- P4 cannot be placed anywhere.

Best-Fit

100 KB free	30 KB P_{10}	417 KB P_2	83 KB free	30 KB P_{11}	112 KB P_3	88 KB free	30 KB P_{12}	212 KB P_1	88 KB free	30 KB P_{13}	426 KB P_4	174 KB free

Worst-Fit

100 KB free	30 KB P_{10}	417 KB P_2	83 KB free	30 KB P_{11}	200 KB free	30 KB P_{12}	300 KB free	30 KB P_{13}	212 KB P_1	112 KB P_3	276 KB free

- P4 cannot be placed anywhere.

Next-Fit

100 KB free	30 KB P_{10}	212 KB P_1	288 KB free	30 KB P_{11}	200 KB free	30 KB P_{12}	300 KB free	30 KB P_{13}	417 KB P_2	112 KB P_3	71 KB free

- P4 cannot be placed anywhere.

Question 2:

Let the page size be 1 KB (1024 ~ 2^{10}).

Note: The table includes the numbers' binary representations with the first 10 bits underlined for increased clarity.

Address	**Page Number**	**Offset**
2375 100101000111	2 10	327 0101000111
19366 100101110100110	18 10010	934 1110100110
30000 111010100110000	29 11101	304 0100110000
256 (0)100000000	0 0	256 100000000
16385 100000000000001	16 10000	1 0000000001

Question 3:

The following are given.

- Logical address space: 1 GB (2^{32} = 4.294.967.296 bits = 1.073.741.824 bytes)
- Page size: 4 KB (2^{10} = 1024 bytes)
- Physical memory: Up to 512 MB

Then a conventional single-level page table would have 250.000 entries.

- # pages = logical address space / page size
- 1 GB / 4 KB = 1.000.000 KB / 4 KB = 250.000

Marc-Andre Fichtel - 30014709

Inverted page tables have one entry for each real (i.e. physical) page of memory (i.e. frames). And an inverted page table would have 128.000 entries.

- # frames = physical address space / page size
- 512 MB / 4 KB = 512.000 KB / 4 KB = 128.000

Question 4:

Given that a direct memory reference takes 200ns, with a single-level page table stored in memory, it would take 2 * 200ns = 400ns to locate and reference a page in memory.

With a translation lookaside buffer (TLB), if 75% of all page-table entries are found in the TLB, and searching the TLB takes 10ns, then the effective access time is:

- (1-0.75) * (10ns + 2 * 200ns) + 0.75 * (10ns + 200ns) =
- 0.25 * 410ns + 0.75 * 210ns =
- 102.5ns + 157.5ns =
- 260ns

Question 5:

The following are given:

- The page reference string: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6
- 3 frames in physical memory

Then the following are how pages are placed into frames for the LRU and OPT replacement algorithms. Note that the "oldest" page is in the top row. A star signifies that a page fault occured.

LRU (Least Recently Used):

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

There are 15 page faults.

Marc-Andre Fichtel - 30014709
CPSC 457 - Fall 2017

OPT (Optimal):

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

There are 11 page faults.