CPSC 457 – Assignment 5

Lamess Kharfan
Student ID: 10150607
Tutorial: T02

Q1 – Assume an OS has five free memory partitions of 100KB, 500KB, 200KB, 300KB and 600KB. The OS needs to place 4 new processes in memory in the following order: P1 of 212KB, P2 of 417KB, P3 of 112KB and P4 of 426KB. Draw the diagrams of the partitions after the OS has placed the processes using 4 different algorithms: first-fit, best-fit, worst-fit and next fit.

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

P4 cannot be placed because no hole that is big enough exists after placing P1, P2, and P3.

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

Worst fit:

Free 100KB	P10 30KB	P2 417KB	Free 83KB	P11 30KB	Free 200KB	P12 30KB	P3 112KB	Free 188KB	P13 30KB	P1 212KB	Free 388KB

P4 cannot be placed because no hole that is big enough exists after placing P1, P2, and P3.

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

P4 cannot be placed because no hole that is big enough exists after placing P1, P2, and P3.

Q2 – Consider a system with 1KB (1024 bytes) page size. What are the page numbers and offsets for the following addresses?

Address	Page number	Offset
2375	2	327
19366	18	934
30000	29	304
256	0	256
16385	16	1

Q3 – 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.

- 32-bit logical address
- 4KB page size = 2^{12} bytes
- $2^{32}/2^{12} = 2^{20}$ = 1048576 entries

b) An inverted page table

- 512MB physical memory = 2^{29} bytes
- 4KB page size = 2^{12}
- $2^{29} / 2^{12} = 2^{17}$ = 131072 entries

Q4 – 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?**
Every data/instruction access requires two memory accesses, one for page table look up + one for instruction fetch, so:
200ns for page table lookup + 200ns to reference page = 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.**
Effective Memory-Access Time = (1-p)*(tlbs + 2 *ma) + p*(tlbs + ma)
= (1 - 0.75)*(10ns + 2*200) + 0.75*(10ns + 200ns)
= 0.25*410ns + 0.75*210ns
= 102.5ns + 157.5ns
= 260ns

Q5 – 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:

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

11 page faults.