Operating Systems and Systems Programming (15B11CI412)

Tutorial - 8 Virtual Memory

1. Answer
   a) 32/8=4 is the size of the table

   b) size of ram / page size = inverted page table entries
   Inverted page tables have 1 entry / physical frame of memory
   65536

2. Since a page is 2^{10} bytes (1 KB) and each PTE is 2^4 bytes (16 bytes), a 1-page page table contains 64 or 2^6 PTEs ($2^{10}/2^4=2^6$). Each entry points to a page that is 2^{10} bytes (1 KB). With one level of page tables we can address a total of $2^6 \times 2^{10} = 2^{16}$ bytes). Adding another level yields another 2^6 pages of page tables, addressing a total of $2^6 \times 2^6 \times 2^{10} = 2^{22}$bytes. adding a third level 2^6 pages of page tables, addressing a total of $2^6 \times 2^6 \times 2^6 \times 2^{10} = 2^{28}$ bytes. Finally adding a fourth level 2^6 pages of page tables, addressing a total of $2^6 \times 2^6 \times 2^6 \times 2^6 \times 2^{10} = 2^{34}$ bytes. So, we need **4** levels.
3. Answer
   a) n
   b) p

```
1  P ---> page-reference string length.

   n ---> number of different frames in P.

   m ---> number of frames.
```

```
2  Answer (a):
   lower bound on the number of page faults is (n)
```

```
3  Answer (b):
   upper bound on the number of page faults is (p)
```

4. Let p be the page fault rate (the probability that a memory access results in a page fault). Then (1 − p) is the probability that a memory access costs 100 nsec. The probability that a page fault costs 20 msec is 0.7 ∗ p and the probability that a page fault costs 8 msec is 0.3 ∗ p. Since 1 nsec = 1000000 msec,

   EAT = (1-p)*(100) + (p) * (**100**+(1-0.7) * (8msec) + (0.7)*(20msec))

   = 100 -100p + 100p + (2.4e6) *p + (14e6)*p

   200 = 100 + (16.4e6)*p

p = 100/16.4e6

Many of you did not account for this '100' and marks were therefore debited.

5. LRU:

3 frames:

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

Frame 1: 1 4 5 1 7 2 -

Frame 2: 2 - 6 3 - -

Frame 3: 3 1 2 - 6 1 6

15 faults

4 frames:

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

Frame 1: 1 - - 6 -

Frame 2: 2 - - - - -

Frame 3: 3 5 3 - -

Frame 4: 4 6 7 1

10 faults

5 frames:

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

Frame 1: 1 - - -

Frame 2: 2 - - - - -

Frame 3: 3 6 - -

Frame 4: 4 3 - -

Frame 5: 5 7

8 faults

6 frames:

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

Frame 1: 1 - - -

Frame 2: 2 - - - - -

Frame 3: 3 - - -

Frame 4: 4

Frame 5: 5 7

Frame 6: 6 - -

7 faults, which is the fewest we can have with 7 pages

FIFO:

3 frames:

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

Frame 1: 1 4 6 3 - 2 - 6

Frame 2: 2 - 1 2 - 7 1

Frame 3: 3 5 1 6 3

16 faults

4 frames:
1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6
Frame 1: 1 - 5 3 - 1
Frame 2: 2 - 6 7 3
Frame 3: 3 2 - 6 -
Frame 4: 4 1 2 -
14 faults

5 frames:
1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6
Frame 1: 1 - 6 - -
Frame 2: 2 - - 1 -
Frame 3: 3 2 - -
Frame 4: 4 3 - -
Frame 5: 5 7
10 faults

6 frames:
1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6
Frame 1: 1 - - 7
Frame 2: 2 - - - - 1
Frame 3: 3 - - 2
Frame 4: 4 3
Frame 5: 5
Frame 6: 6 - -
10 faults
Optimal:
3 frames:
1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6
Frame 1: 1 - - 3 - -
Frame 2: 2 - - - 7 2 -
Frame 3: 3 4 5 6 - 1 6
11 faults

4 frames:
1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6
Frame 1: 1 - - 7 1
Frame 2: 2 - - - - - -
Frame 3: 3 - - -
Frame 4: 4 5 6 - -
8 faults

5 frames:
1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6
Frame 1: 1 - - - -
Frame 2: 2 - - - - - -
Frame 3: 3 - - -
Frame 4: 4 7

Frame 5: 5 6 - -
7 faults, which is the best we can do

6. Thrashing is caused by under allocation of the minimum number of pages required by a process, forcing it to continuously page fault. The system can detect thrashing by evaluating the level of CPU utilization as compared to the level of multiprogramming. It can be eliminated by reducing the level of multiprogramming.
7. Using optimal page replacement technique, following pagefaults will occur:

0	9	0	1	8	1	8	7	8	7	1	2	8	2	7	8	2	3	8	3
0	0	X	0	9	X	X	1	X	X	X	8	X	X	X	X	X	8	X	X
	9		9	1			8				7						2		
			1	8			7				2						3		

We can see that total 7 page faults occur.

8. (a) 10451 = page 5, offset 451
(b) 5421 = page 2, offset 1421
(c) 14123 = page 7, offset 123
(d) 9156 = page 4, offset 1156

So (a) is not in memory, and faults.

(b) is in memory, corresponds to frame 200, offset 1421. If memory is strictly linear, this corresponds to a physical address of 200*2000+1421.

(c) is in memory, corresponds to frame 101, offset 123, ie 101*2000+123.

(d) is not in memory, and faults.