Operating systems

Sheet 9
(EED)

Name: Samuel Ayman Shawky

ID: 20010750

Q1)

a- Split binary address into virtual page number and offset; use virtual page number as index into page table; extract page frame number; concatenate offset to get physical memory address.

b-

(i)1052 = 1024 + 28

So virtual page number=1 & page frame number=7

Physical address=7*1024+28

(ii) 2221 = 2 × 1024 + 173

So virtual page number=2

Page fault

(iii) 5499 = 5 × 1024 + 379

So virtual page number=5 & page frame number=0

Physical address=0*1024+379

Q2)

a-4MB

b-Number of rows: 2^6x2=128 entries. Each entry consists of: 20 (page number) + 20 (frame number) + 8 bits (chain index) = 48 bits = 6 bytes. Total: 128 × 6= 768 bytes

Q3)

a-

$2^{32} / 2^{10} = 2^{22}$ pages,

so 22 bits needed to specify a page in virtual memory.

Each page table contains (2^{10} bytes per page table)/(4 bytes/entry) = 2^8 entries. each page table can handle 8 of the required 22 bits. Therefore, 3 levels of page tables are needed.

b- (8 + 8 + 6 = 22).

c-Less space is consumed if the top level has 2^6 entries. In this case pages =1 + 2^6 + 2^{14} = 16,449 pages. If the middle level has 2^6 entries, then pages is 1 + 2^8 + 2^{14} = 16,641 pages. If the bottom level has 2^6 entries, then the number of tables is 1 + 2^8 + 2^{16} pages = 65,973 pages.

Q4)

$2^{64} / 2^{12} = 2^{52}$ entries

Entries per page=4kbyte/4byte=1kbyte=2^{10}

It will be 6 levels.

(2+10+10+10+10+10)

Q5)

a-

8x2=16kbyte

b-

4x16=64kbyte

c-

$\mathbf{2^{32}/2^{11}=2^{21}}$

0000 0000 0000 0010 00011 010 1011 1100

Page number(2)	Segment(3)	Offset(11)
21		11

2^{32} = 4 GBytes.

Q6)

a- page number take 5 bits(32 $=2^5$page)

offset take 11 bits(2k $=2^{11}$)

Page number(5)	Offset(11)

b-

length equal number of pages =32

width equal number of bits needed to represent physical memory.

1Mbyte/2Kbyte=512=2^9

Width =9 bits

c-

if the physical memory space is reduced by half

the length of page table stays the same =32.

the width will change to equal 8bits.

Q7)

a-

In paged system we need two memory access one to access page table and get frame number and the other to access data so access time =2*200=400ns

b-

EMAT = 0.85 * 220 ns + 0.15 * 420 ns = 187 ns + 63 ns = 250 ns.

Q8)

a-FIFO

7	0	1	2	0	3	0	4	2	3	0	3	2
7	7	7	2	2	2	2	4	4	4	0	0	0
	0	0	0	0	3	3	3	2	2	2	2	2
		1	1	1	1	0	0	0	3	3	3	2
			f		f	f	f	f	f	f		

b-LRU

7	0	1	2	0	3	0	4	2	3	0	3	2
7	7	7	2	2	2	2	4	4	4	0	0	0
	0	0	0	0	0	0	0	0	3	3	3	3
		1	1	1	3	3	3	2	2	2	2	2
			f		f		f	f	f	f		

c-CLOCK

7	0	1	2	0	3	0	4	2	3	0	3	2
7	7	7	2	2	2	2	4	4	4	0	0	0
	0	0	0	0	3	3	3	2	2	2	2	2
		1	1	1	1	0	0	0	3	3	3	3
			f		f	f	f	f	f	f		

d-optimal

7	0	1	2	0	3	0	4	2	3	0	3	2
7	7	7	2	2	2	2	2	2	2	2	2	2
	0	0	0	0	0	0	4	4	4	0	0	0
		1	1	1	3	3	3	3	3	3	3	3
			f		f		f			f		

e-

algorithm	Page fault	Miss rate
FIFO	7	7/13
LRU	6	6/13
CLOCK	7	7/13
OPTIMAL	4	4/13

Q9)

a-page frame number=3 as first entered at time 20.

b-page frame number=1 as first referenced at 160.

c-page frame number=0 as R=0.

d-page frame number=3 as furthest use in the future.

e- There are 6 faults.

Q10)

1-FIFO 3-page frames

A	B	C	D	A	B	E	A	B	C	D	E
A	A	A	D	D	D	E	E	E	E	E	E
	B	B	B	A	A	A	A	A	C	C	C
		C	C	C	B	B	B	B	B	D	D
			F	F	F	F			F	F	

2-FIFO 4-page frames

A	B	C	D	A	B	E	A	B	C	D	E
A	A	A	A	A	A	E	E	E	E	D	D
	B	B	B	B	B	B	A	A	A	A	E
		C	C	C	C	C	C	B	B	B	B
			D	D	D	D	D	D	C	C	C
						F	F	F	F	F	F

Replacement done every F

Q11) a- LRU

1	0	2	2	1	7	6	7	0	1	2	0	3	0	4
1	1	1	1	1	1	1	1	1	1	1	1	1	1	4
	0	0	0	0	0	6	6	6	6	2	2	2	2	2
		2	2	2	2	2	2	0	0	0	0	0	0	0
					7	7	7	7	7	7	7	3	3	3
F	F	F			F	F		F		F		F		F

5	1	5	2	4	5	6	7	6	7	2	4	2	7	3
4	4	4	4	4	4	4	4	4	4	2	2	2	2	2
5	5	5	5	5	5	5	5	5	5	5	4	4	4	4
0	0	0	2	2	2	2	7	7	7	7	7	7	7	7
3	1	1	1	1	1	6	6	6	6	6	6	6	6	3
F	F		F			F	F			F	F			F

3	2	3
2	2	2
4	4	4
7	7	7
3	3	3

Hit ratio=16/33

b- FIFO

1	0	2	2	1	7	6	7	0	1	2	0	3	0	4
1	1	1	1	1	1	6	6	6	6	6	6	6	6	4
	0	0	0	0	0	0	0	0	1	1	1	1	1	1
		2	2	2	2	2	2	2	2	2	0	0	0	0
					7	7	7	7	7	7	7	3	3	3
F	F	F			F	F			F		F	F		F

5	1	5	2	4	5	6	7	6	7	2	4	2	7	3
4	4	4	4	4	4	6	6	6	6	6	6	6	6	6
5	5	5	5	5	5	5	7	7	7	7	7	7	7	7
0	1	1	1	1	1	1	1	1	1	1	4	4	4	4
3	3	3	2	2	2	2	2	2	2	2	2	2	2	3
F	F		F			F	F				F			F

3	2	3
6	2	2
7	7	7
4	4	4
3	3	3
	F	

Hit ratio=16/33

C- in this case the two policies are equally effective.

Q12)

a-lower bound on the number of page faults=N

b-upper bound on the number of page faults=P

Q13)

a-

64*64=4096

b-

```
#define Size 64
 int A[Size; Size], B[Size; Size], C[Size; Size];
 int register i, j;
for (i = 0; i< Size; i++)
   for (j = 0; j< Size; j ++)
      C[i; j] = A[i; j] + B[i; j];
```

c-

64