a. 由于该进程是按列读数组的,然后有3个页可以放数据,正好给A,B,C三个数组各分配三个页。又因为每隔数组在每页中可以存4行,所以每执行4次语句就会出现一次缺页中断。所以一共出现了

$$64 \times 64 \div 4 \times 3 = 3072$$
次

b.将其改为按行访问数组

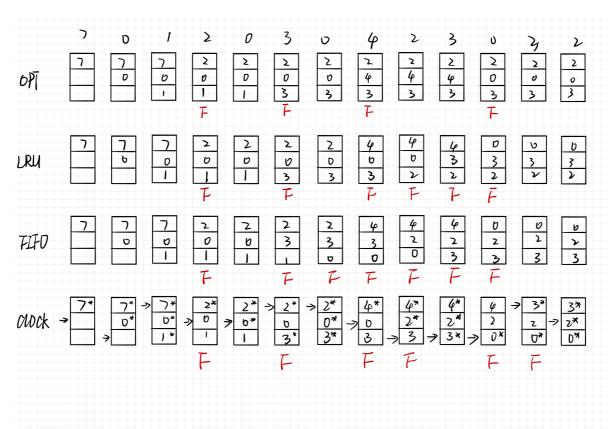
```
for(int i = 0; i < Size; i++){
   for(int j = 0; j < Size; j++){
        C[i, j] = A[i, j] + B[i, j];
   }
}</pre>
```

c.修改之后就是每进行4\*64次语句出现一次缺页中断,所以频率是

$$64 \times 64 \div (64 \times 4) \times 3 = 48\%$$

8.4

假设系统为进程分配了3个页框,一共进行了13次页面访问



置换算法	缺页中断次数	缺页率
OPT	4	30.77%
LRU	6	46.15%
FIFO	7	53.85%
CLOCK	6	46.15%

## 8.6

a.使用LRU置换策略时,在4个页框中驻留的页面情况如下表。

命中率为 $16 \div 33 = 48.48\%$ 

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

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

b.使用FIFO置换策略时,在4个页框中驻留的页面情况如下表。

命中率为  $16 \div 33 = 48.48\%$ 

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

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

c.从计算结果可以看出,这两种页面置换算法对于这个访问序列的命中率一样,说明两种算法在这个访问序列的情况中效果一致,不分优劣。

## 8.11

a.一共需要400ns,因为页表在内存中,所以先用200ns访问页表用来获取相应的页表项,之后200ns访问对应的物理内存,一共需要400ns。

b.

$$220 \times 85\% + 420 \times 15\% = 250 ns$$

c.如果TLB命中,则直接获取数据,不用再访问内存,可以节省200ns的时间,所以TLB命中率越高,EMAT越短。