

1

OS Test

Fernando
Vazquez
3/4/19

Direct Map

Reference	Binary Address	Hit/Miss	index	tag
6	0000 0110	Miss	0011	0
26	0001 1010	Miss	1101	0
7	0000 0111	Miss Hit	0011	0
24	0001 1000	Miss	1100	0
17	0001 0001	Miss	1000	0
66	0100 0010	Miss	0001	0
22	0001 0110	Miss	1011	0
6	0000 0110	Miss Miss	0011	0
8	0000 1000	Miss	0100	0
118	0111 0110	Miss	1011	0
68	0100 0100	Miss	0010	0

$$\# \text{ index} = \log_2 16 = 4$$

$$\# \text{ offset} = \log_2 8 = 3$$

$$\# \text{ of tag bits} = 8 - 4 - 3 = 1$$

cache address	TAG	Data
0 0000		
1 0001	0001	0001 0001
2 0010	0100	0100 0100
3 0011		
4 0100	0100	0100 0100
5 0101		
6 0110	0000 0111	0000 0110 0111 0110
7 0111	0000	0000 0111
8 1000	0000	0000 1000
9 1001		
10 1010	0001	0001 1010
11 1011		
12 1100		
13 1101		
14 1110		
15 1111		

2-way associative

cache
address

0 0000

1

2 0001

3 0010

4 0011

5 0100

6 0101

7 0110

8 0111

	Bin address	Hit/Miss	Index	Tag
6	0000 0110	Miss	011	00
26	0001 1010	Miss	101	00
7	0000 0111	Miss	011	00
24	0001 1000	Miss	100	00
17	0001 0001	Miss	000	00
66	0100 0010	Miss	001	01
22	0001 0110	Miss	011	00
6	0000 0110	Miss Hit	011	00
8	0000 1000	Miss	100	00
110	0110 0110	Miss	011	01
60	0100 0100	Miss	010	01

~~# of index~~ # of index = $\log_2 8 = 3$

of offset = $\log_2 8 = 3$

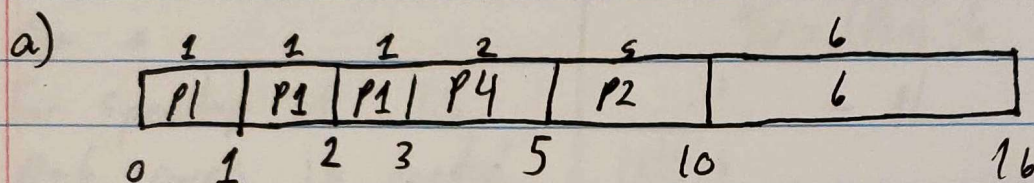
of Tag = $8 - 3 - 3 = 2$

2.

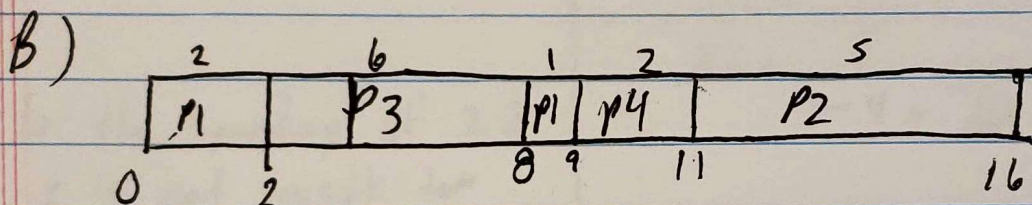
	arrival	Burst	Priority
P1	0	3	2
P2	1	5	4
P3	2	6	1
P4	3	2	3

Quantum = 2 units

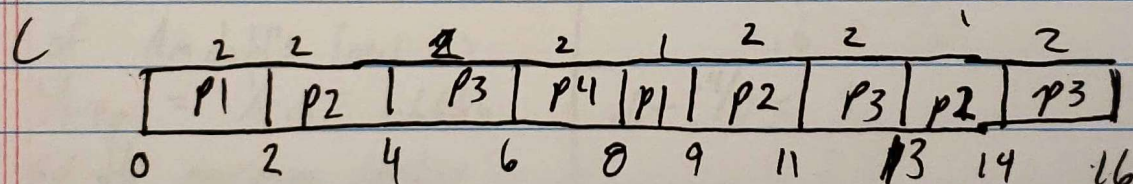
preemptive
SJF



preemptive
priority



RR



2a. P2 wait : 4
P3 wait : 8

average wait time for preemptive
SJF is 3 five units

2b. P1 6
P2 10
P3 0
P4 6

average wait time for preemptive
priority is $\frac{6+10+0+6}{4} = 5.5$ units of time

P4: 3+

$$\frac{3+6+8+6}{4} = \frac{23}{4} = 6.25$$

2c. P1: 6

P2: 5+2=8

P3: 2+5+1=8

average wait time for RR is
6.25 five units

3. Parallelizable = 40%

$$\text{speedup} = \frac{1}{(1-F_p) + F_p/N}$$

~~For~~ *

For speedup of 1.5
 $N=6$ cores is needed.

For the speedup of 2.2,
it is not possible to
reach because the limit
of Amdahl's law is
 $1/(1-F_p) = 1/(1-.4) = 1.6666$.
So the maximum speedup
possible with 40% of
code parallelizable is
1.6667

$$1.5 = \frac{1}{1-.4 + .4/N}$$

$$1.5((1-.4) + .4/N) = 1$$

$$.6 + .4/N = 1/1.5$$

$$.4/N = (1/1.5) - .6$$

$$\frac{.4}{(1/1.5) - .6} = N$$

$$N = 6$$

$$2.2 = \frac{1}{1-.4 + .4/N}$$

$$2.2 = \frac{1}{.6 + .4/N}$$

$$.6 + .4/N = \frac{1}{2.2}$$

$$.4/N = .4545 - .6$$

$$\frac{.4}{.4545 - .6} = N$$

$$.4545 - .6$$

4. step 1: assign 4 numbers from array A and 4 numbers to array B ~~to~~ to the 16 cores.

step 2: ~~multiply~~ multiply each A value to its corresponding B value. 4 steps

$D+D$ $D+D$

step 3: add ~~the~~ numbers together $a_1b_1 + a_2b_2$ and $a_3b_3 + a_4b_4$, ~~steps~~ for each of the processors 4 numbers. 2 steps

1

step 4: Each processor adds its last 2 numbers. 1 step

step 5: 8 processors add the remaining 16 numbers. 1 step

step 6: 4 processors add the remaining 8 numbers. 1 step

step 7: 2 processors add the ~~remaining~~ remaining 4 numbers. 1 step

step 8: 1 processor adds the last 2 numbers, thus resulting in the dot product of 2 ~~arrays~~ arrays of 64 integers each.

Total steps is 11 steps to calculate the dot product.