

الف 1)

	Need
P <sub>0</sub>	0 0 0 0
P <sub>1</sub>	0 7 5 0
P <sub>2</sub>	1 0 0 2
P <sub>3</sub>	0 0 2 0
P <sub>4</sub>	0 6 4 2

	work	need i
✓	1 5 2 0	0 0 0 0
X	1 5 3 2	0 7 5 0
✓	1 5 3 2	1 0 0 2
✓	2 8 8 6	0 0 2 0
✓	2 14 11 8	0 6 4 2

work = 1 5 2 0 1 5 3 2  
finish 

0	1	2	3	4
✓	✓	✓	✓	✓

2 8 8 6 2 14 11 8

safe!

~~2 8 8 6~~

ب) 

work	need i
>	0 4 2 0

 ⇒ نه!

موتدان « لحظه منابع را اقتباس داد و امن است »

2)

//lock//

for ( j=0 و j < number-of-processes; j++) {

flags[i]=j

turn[j]=i

{ for all k != i, flags[k] < j } || { turn[j] != i }

//unlock//

flags[i]=0

⇒ I'm Done!

✓ مع پردازش ای در CR برای معنی ماند (به خاطر شرط ها)

Mutual Exclusion

✓ مع پردازش ای می تواند در CR پردازش ای دیگر دخالت کند

Progress

✓ مع پردازش ای مع معنی کند

Bounded waiting

برای شرط 3 گانه :

3)

mutual exclusion نداریم، زیرا پردازش ها صرفاً به حسب interest وارد می شوند و نه interest (AND) (میت)

progress هم نداریم چون اگر interest نباشد با هم علاوه منتظر بمانیم یا ترتیب آن تمام شود.

Bounded waiting هم طبقاً نداریم به خاطر progress.

④

available: [6, 4, 4, 2]

Need

$$\text{Max-Allocated: } \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 2 \\ 2 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{bmatrix}$$

الف) بعد از انجام Banker's Algorithm پدانه  $P_1$  می تواند درخواست کند.

ب) خیر، بن بست نداریم و اگر پدانه ها به طور ترتیبی مجاز طبق

الگوریتم منابع را بگیرند در نهایت  $P_1$  هم درست می شود و به بن بست نمی خوریم.

ج) چن بن بست نداریم پس این است.

د) خیر زیرا مقدار باید از available کمتر باشد و همچنین از need که این طور نیست.

Finishes: [ ] [ ] [ ] [ ]

work: [6, 4, 4, 2]

شماره پدانه			
$P_0$ needs: [1, 2, 0, 0] work: [5, 2, 4, 2] ✓ safe	$P_1$ needs: [0, 1, 0, 2] work: [0, 1, 4, 0] ✓ safe	$P_2$ needs: [2, 2, 0, 0] X wait	$P_3$ : [1, 1, 0, 0] needs: [2, 2, 0, 0] <del>work: [ ]</del> work: [4, 0, 4, 0] ✓

⑤

الف)

موتال اکسکلوژن

Mutual Exclusion

Progress

Boundary waiting

MA

ب)

MA: داریم زیرا هیچ پدانه ای نمی تواند وارد CR شود!

Progress: نداریم زیرا هر پدانه که بخواهد را قفل کرده است.

BW: نداریم و پدانه ها به نقطه هم نمی رسند.