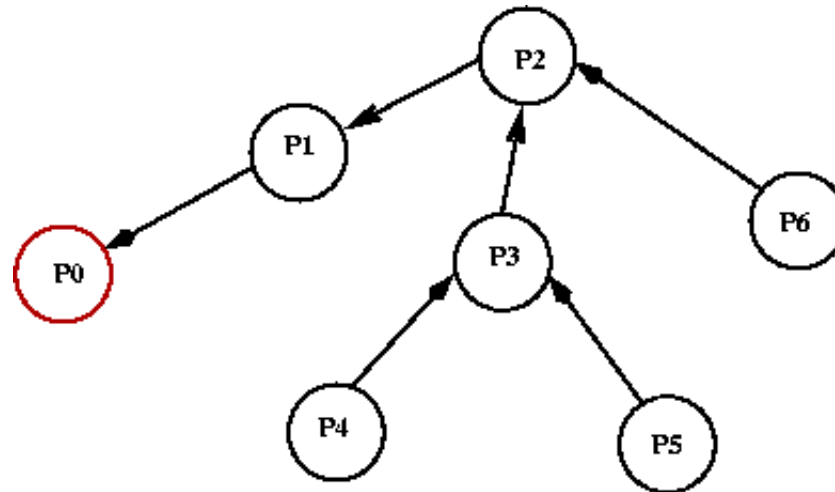


## A Tree-Based, Mutual Exclusion, Token Algorithm Example

by Raymond (1989)

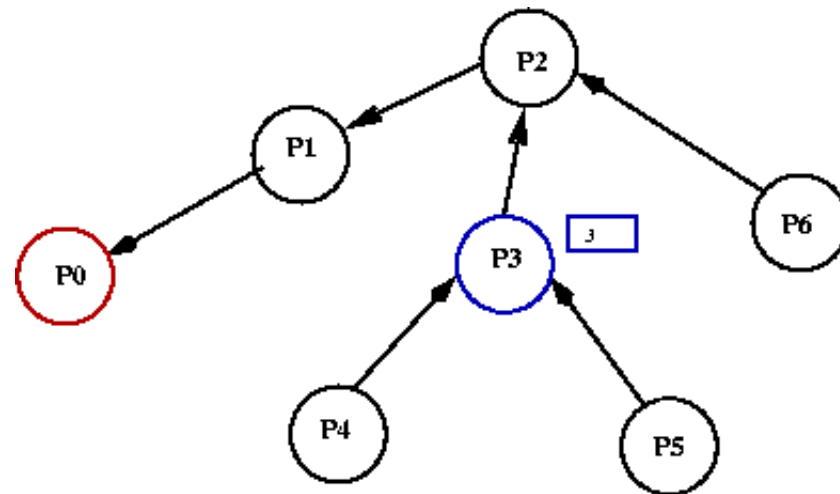
Initially,  $P_0$  holds the token.  
Also,  $P_0$  is the current root.



Raymond Tree Algorithm: Step 0

*(Adapted from Chojoh97)*

$P_3$  wants the token  
to get into its critical section.  
So,  $P_3$  adds itself to its own FIFO queue  
and sends a request message to its parent  $P_2$ .



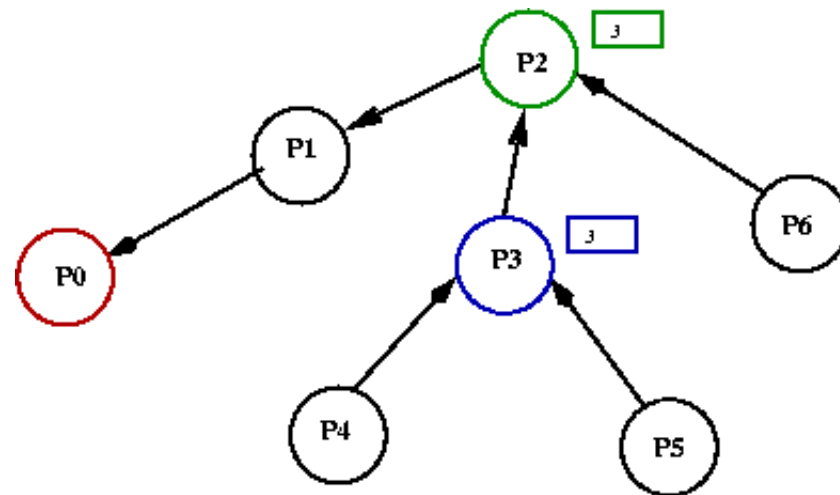
Raymond Tree Algorithm: Step 1

*(Adapted from Chojoh97)*

P2 receives the request from P3.

It adds P3 to its FIFO queue

and passes the request message to its parent P1.



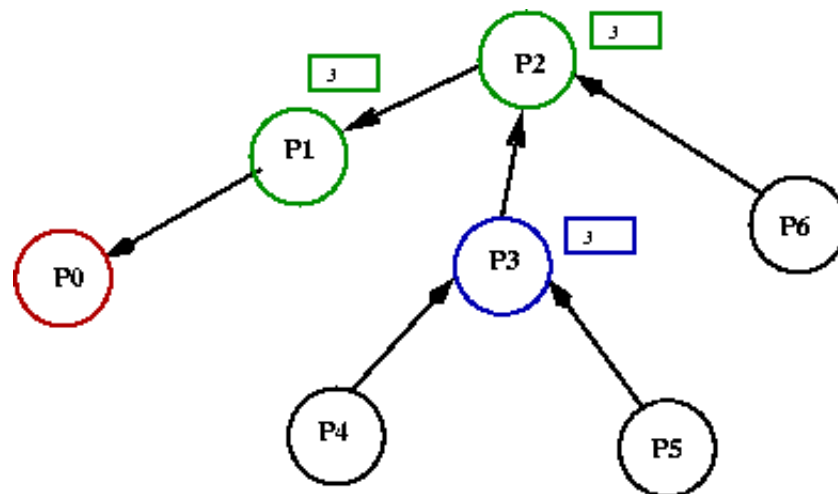
Raymond Tree Algorithm: Step 2

*(Adapted from Chojoh97)*

P1 receives the request from P2.

It adds P3 to its FIFO queue

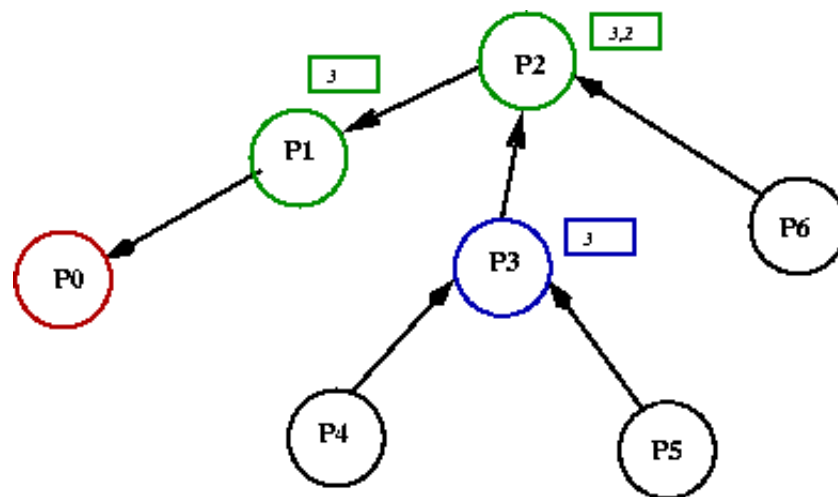
and passes the request message to its parent  $P_0$



Raymond Tree Algorithm: Step 3

*(Adapted from Chojoh97)*

At this point,  $P_2$  also wants the token.  
Since its FIFO queue is not empty,  
it adds itself to its own FIFO queue.



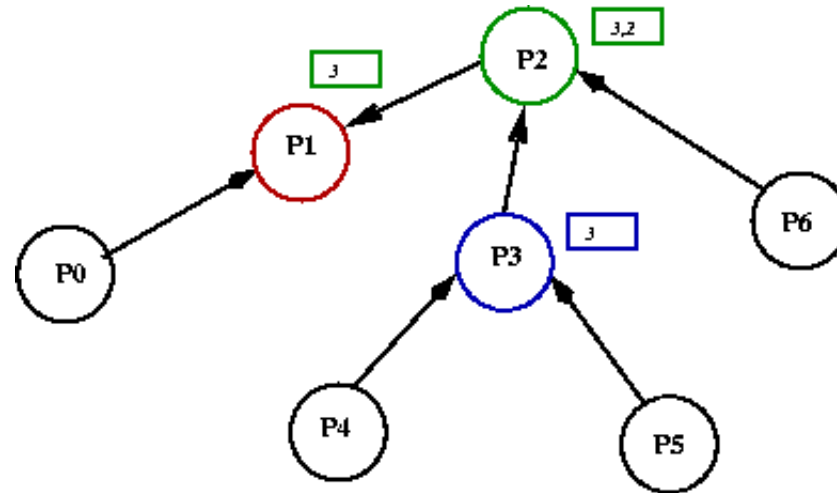
Raymond Tree Algorithm: Step 4

*(Adapted from Chojoh97)*

P0 receives the request message from P3 through P1.

It surrenders the token and passes it on to P1.

It also changes the direction of the arrow between them, making P1 the root, temporarily.



**Raymond Tree Algorithm: Step 5**

*(Adapted from Chojoh97)*

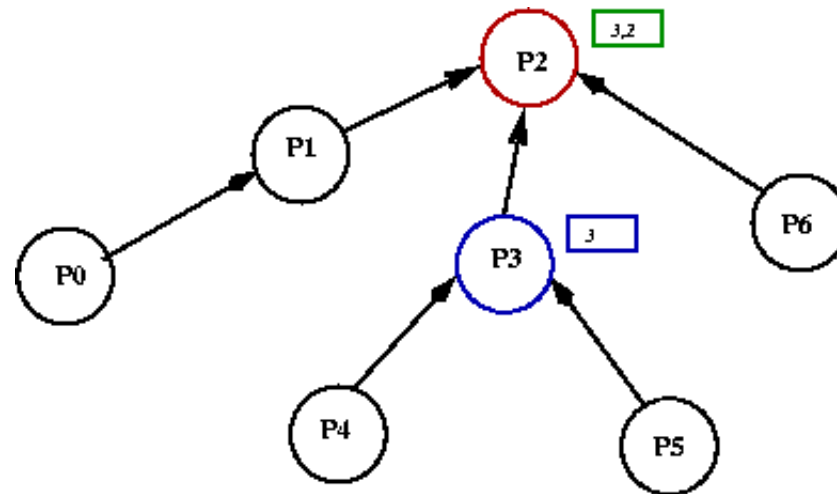
P1 removes the top element of its FIFO queue

to see which node requested the token.

Since the token needs to go to P3,

P1 surrenders the token and passes it on to P2.

It also changes the direction of the arrow between them, making P2 the root, temporarily.



**Raymond Tree Algorithm: Step 6**

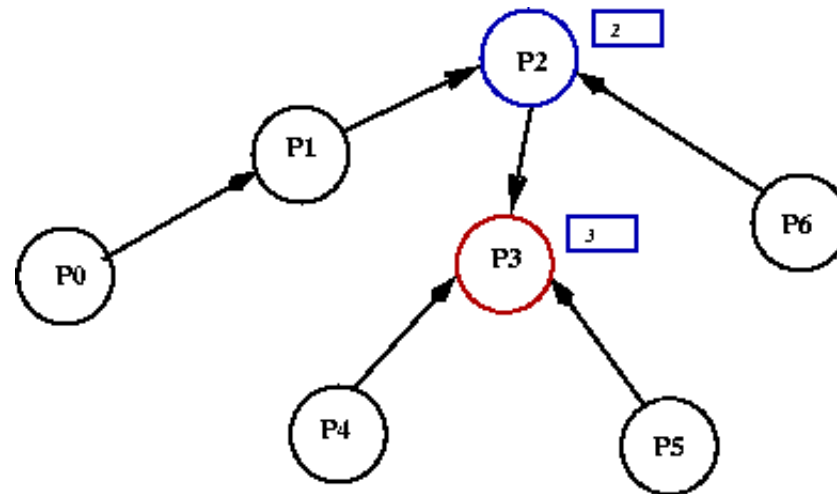
*(Adapted from Chojoh97)*

P2 removes the top element of its FIFO queue  
to see which node requested the token.

Since the token needs to go to P3,

P2 surrenders the token and passes it on to P3.

It also changes the direction of the arrow between them,  
making P3 the root.

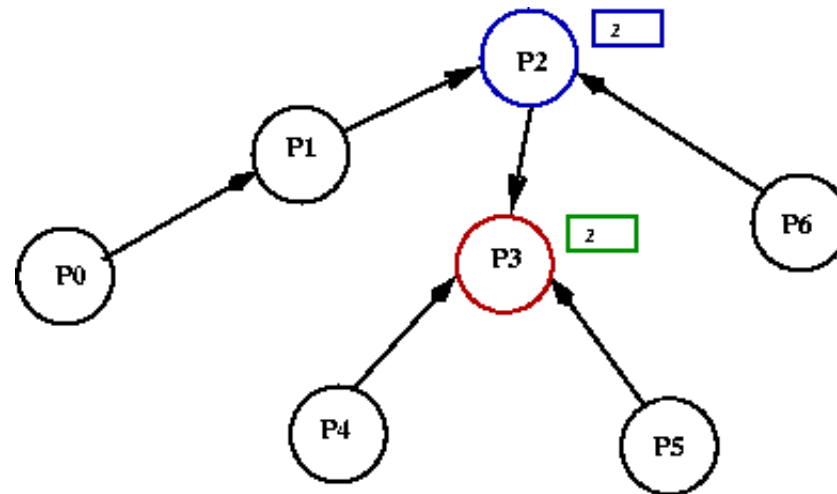


**Raymond Tree Algorithm: Step 7**

*(Adapted from Chojoh97)*

Now, **P3** holds the token.  
and can execute its critical section.  
It is able to clear the top (and only)  
element of its FIFO queue.  
Note that **P3** is the current root.

In the meantime, **P2** checks the top element of its FIFO queue  
and realizes that it also needs to request the token.  
So, **P2** sends a request message  
to its current parent, **P3**,  
who appends the request to its FIFO queue.



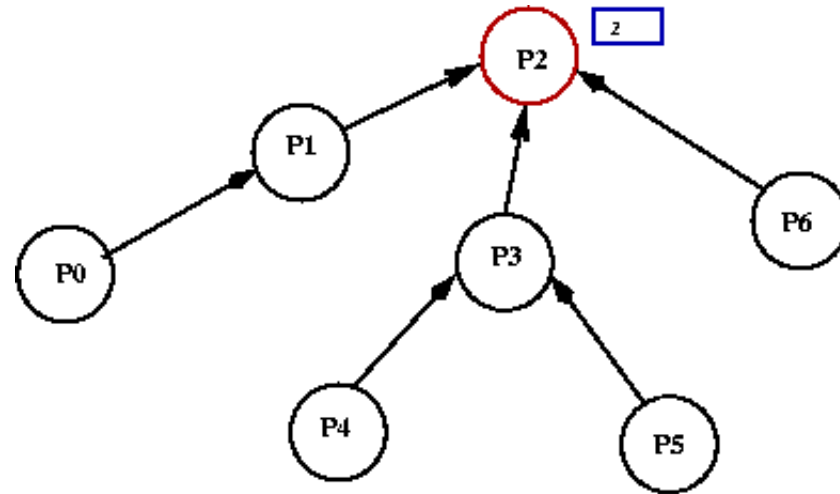
**Raymond Tree Algorithm: Step 8**

*(Adapted from Chojoh97)*

As soon as **P3** completes its critical section,  
it checks the top element of its FIFO queue  
to see if it is needed elsewhere.

In this case, **P2** has requested it,  
so **P3** sends it back to **P2**.

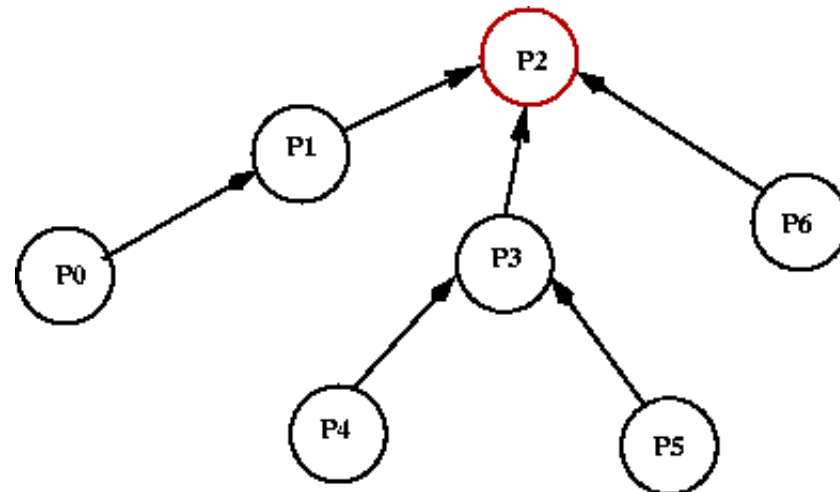
It also changes the direction of the arrow between them,  
making **P2** the new root.



**Raymond Tree Algorithm: Step 9**

*(Adapted from ChoJoh97)*

P2 holds the token  
 and is able to complete its critical section.  
 Then it checks its FIFO queue, which is empty.  
 So it waits until some other node requests the token.



**Raymond Tree Algorithm: Step 10**

*(Adapted from ChoJoh97)*

(Adapted from ChoJoh97)



---

---

Comments: [schauble@cs.colostate.edu](mailto:schauble@cs.colostate.edu)

---