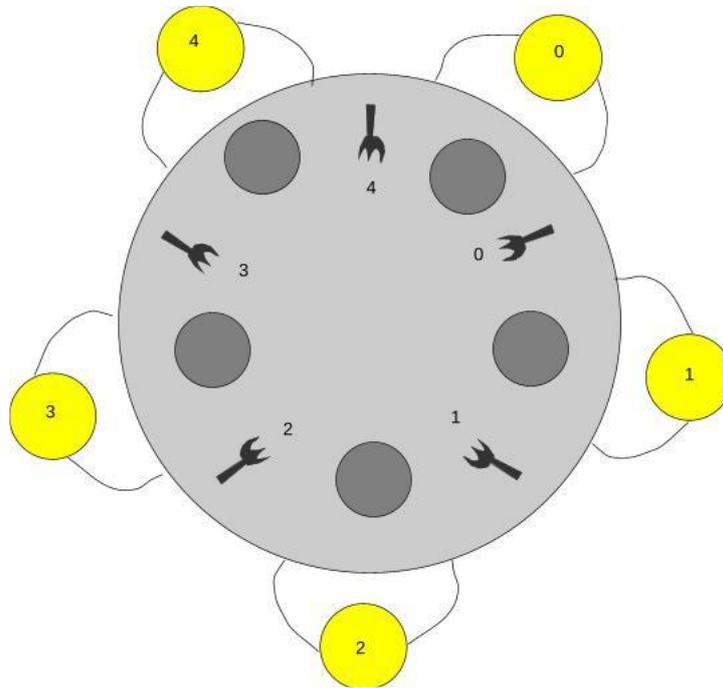


Le problème des Philosophes

Le problème des Philosophes



Philosophe i
<code>penser();</code> <code>manger();</code>

5 philosophes sont réunis autour d'une table pour manger des spaghetti. Chaque philosophe doit utiliser 2 fourchettes pour manger.

Problème : modéliser le comportement de chaque philosophe pour éviter les privations et les blocages.

Solution (fausse)

Philosophe i
<code>Penser();</code> <code>P(fourchette i);</code> <code>P(fourchette $(i+1) \bmod 5$);</code> <code>manger();</code> <code>V(fourchette i);</code> <code>V(fourchette $(i+1) \bmod 5$);</code>

Si tous les philosophes prennent en même temps leur fourchette i , il y a interblocage.

Si les philosophes ne commencent pas à manger en même temps.

La solution serait : Penser=0 ; Faim=1 ; Manger=2 ; tableau de sémaphores S initialisé à 0 (donc bloquants des le départ).

Philosophe <i>i</i>
<pre> Etat[i]:=penser; si etat[(i-1) mod 5]=Manger ou etat[(i+1)mod 5]=Manger alors Etat[i]:=Faim; P(s[i]); Finsi Etat[i]:=Manger; Si Etat[(i-1) mod 5]=Faim et Etat[(i-2) mod 5]#Manger alors V(s[i-1]); Si Etat[(i+1) mod 5]=Faim et Etat[(i+2) mod 5]#Manger alors V(s[i+1]); Etat[i]:= Penser; </pre>

Cette solution est erronée parce que le tableau Etat peut être testé par tous les philosophes en même temps.

La solution est que ce tableau soit manipulé en exclusion mutuelle.

Solution (juste)

Philosophe <i>i</i>	prendre_fourchette(<i>i</i>)	poser_fourchette(<i>i</i>)
<pre> Penser(); prendre_fourchette(<i>i</i>); manger(); poser_fourchette(<i>i</i>); </pre>	<pre> P(mutex); état[i] = FAIM; test(<i>i</i>); V(mutex); P(s[i]); </pre>	<pre> P(mutex); état[i] = PENSE; test(GAUCHE); test(DROITE); V(mutex); </pre>

test(<i>i</i>)
<pre> si (état[i] == FAIM et état[GAUCHE] # MANGE et état[DROITE] # MANGE) alors . état[i] = MANGE; . V(s[i]); </pre>