

Concurrence – Master 1

TD 4 : Sémaphores

Correction 1 : Comme le sémaphore s_2 est initialement à 0, le processus P1 est bloqué.

Si P2 s'exécute avant P3, il garde x à 0, donc P3 le mets à 3 et P1 ensuite affecte x à 6.

Si P3 s'exécute avant P2, il mets x à 3, et il débloquent P1 en signalant s_2 . Alors P1 et P2 se trouvent en concurrence pour s_1 , dès que P3 signale s_1 . Si P1 gagne, il va bloquer P2 (car P1 oublie de signaler s_1) et le résultat est x à 6. Si P2 gagne, il augmente x à 9 et signale s_1 , donc P1 augmente x à 18.

Attention, ceci est vrai pour toute implementation du sémaphore binaire en Promela car le comportement est acyclique...

Pour vérifier ce résultat, tester la formule $\langle \rangle (x6 \ || \ x18)$ avec $x6$ défini par $x==6$, etc. en crochant l'option No execution.

Correction 2 :

Voir les listings joints.

Correction 3 :

Voir les listings joints.