

Rapport TP1

- Organisation

On a un fichier partagé entre plusieurs processus verDeTerre qui vont accéder à différentes parties de ce fichier en même temps. Le choix le plus judicieux : pose de verrous internes car on ne veut pas bloquer tout le fichier. Un verrou externe n'est pas envisageable sinon le fichier serait édité de manière synchrone. La portée d'un verrou interne rend possible l'écriture/lecture de plusieurs processus dans un fichier, et ce, de manière asynchrone.

- Vision.c

Le programme vision.c s'occupe d'afficher la totalité du contenu (édité par les vers) à chaque modification du fichier. Vision.c accède au fichier sans que celui-ci soit édité par les vers : on fait la pose d'un verrou bloquant sur tout le fichier et on affiche les positions des vers. Si le verrou n'était pas de type bloquant, les données affichées ne seraient pas fidèles (puisque le fichier est en constante « édition »). On pourrait toutefois éviter les bugs d'affichage en faisant un sleep() plus long que celui utilisé par les vers et changer le verrou de lecture de Vision.c en un verrou « non-bloquant » (F_SETLK au lieu de F_SETLKW).

- Pour le fichier verDeTerre.c

La fonction qui place la première tête du ver la place n'importe où dans le fichier, donc on pose un verrou sur tout le fichier. Cela évite qu'il y ait deux processus ou plus en mode écriture sur le même morceau du fichier. On enlève le verrou et on met en attente le verDeTerre pour qu'il ne monopolise pas tout le terrain (faire en sorte que les autres vers puissent aussi placer leur tête).

Pendant leur durée de « vie », chaque processus verDeTerre pose un verrou bloquant sur les cases disponibles autour de sa tête. On prend d'abord la liste de ses voisins, on verrouille les cases voisines indiquées dans cette liste, on regarde à nouveau dans la liste des voisins les cases disponibles pour y écrire la marque du ver, on déplace la tête du ver et on enlève le verrou (de chacune des cases verrouillées, donc 3 x 3 cases).