

Cours "Informatique Embarquée"

François Armand

M2 SRI/Crypto

Exercice N° 8

Compte-rendu à rendre le 31/12/2013 23H59 :-)

armand@informatique.univ-paris-diderot.fr

Pour ou contre l'héritage de priorité

Cet exercice se présente sous forme d'une suite de questions indépendantes les unes des autres qui nécessitent d'avoir lu le papier (en anglais) de V. Yodaiken « Against priority inheritance ».

Ce papier est accessible par le lien suivant: <http://www.yodaiken.com/papers/inherit.pdf> et aussi depuis le répertoire Documents et Liens du site du cours.

Après avoir lu ce papier (6 pages), vous répondrez aux questions suivantes:

1. Quelles sont selon l'auteur les 2 contraintes qui sont à l'origine des situations d'inversion de priorité?
2. Le système RTLinux (développé par l'auteur) fournit-il un support pour l'héritage de priorité?
3. Quelle propriété doit avoir l'héritage de priorité pour répondre correctement aux cas où il y a des sections critiques imbriquées?
4. Dans le cas de sections critiques imbriquées, quel est le délai maximum qui s'impose alors à la tâche la plus prioritaire bloquée par un verrou déjà occupé par une tâche moins prioritaire?
5. Si on considère que les tâches peuvent de synchroniser avec différents mécanismes, à quelle condition l'héritage de priorité pourra-t-il fonctionner avec succès? Posix le permet-il?
6. Pourquoi le blocage d'une tâche sur des entrées/sorties (I/O) comme un tube, par exemple, peut-il perturber l'héritage de priorité?
7. Est-il possible, selon l'auteur, d'implémenter correctement l'héritage de priorité sur des verrous de type multiples lecteurs / producteur unique (read/write lock).
8. Quelles sont les « étapes » élémentaires induites par un héritage de priorité dans la situation la plus simple?
9. Quelle alternative simple est-elle proposée par l'auteur pour éviter l'héritage de priorité dans certains cas? Quel avantage cela procurerait-il dans l'exemple proposé?
10. Quelle hypothèse fausse a été faite dans l'article [3] cité en référence?