



**POLYTECHNIQUE
MONTREAL**

**LE GÉNIE
EN PREMIÈRE CLASSE**

INF3610 – Systèmes embarqués

**Rapport du laboratoire 1
Introduction à μ C-II/OS**

Groupe 02

**Simon Nadeau – 1791314
Mathieu Châteauvert – 1846732**

**Travail présenté à
Eva Terriault**

**Rapport remis le 20 septembre 2018
École Polytechnique de Montréal**

Réponses aux questions supplémentaires

Question a

Dans le contexte du problème d'inversion de priorité et de ses solutions, précisez ce qu'on entend par temps de blocage versus temps de préemption.

Le temps de préemption est le temps pendant lequel une tâche ne peut pas s'exécuter à cause d'une autre tâche plus prioritaire qui a besoin de s'exécuter, ce qui devrait se produire normalement. La tâche la moins prioritaire est donc préemptée pendant un certain temps, jusqu'à ce que la tâche plus prioritaire termine son exécution. D'un autre côté, le temps de blocage est le temps pendant lequel une tâche plus prioritaire ne peut pas s'exécuter à cause d'une ressource qu'elle partage avec une autre tâche moins prioritaire qui l'utilise déjà et la rend indisponible. La tâche la plus prioritaire est donc bloquée par la tâche la moins prioritaire, comme si les priorités étaient inversées, ce qui n'est pas désirable compte tenu du fait que les tâches plus prioritaires sont souvent plus critiques.

Question b :

Toujours dans le contexte du problème d'inversion de priorité, donnez 2 avantages d'utiliser le protocole ICPP par rapport à l'héritage de priorité.

Le protocole ICPP est plus efficace que l'héritage de priorités pour minimiser le temps de blocage. De plus, le protocole ICPP permet de limiter plusieurs changements de contexte par rapport à l'héritage de priorité.

Question c :

Donnez un exemple de situation où il serait préférable d'utiliser un sémaphore plutôt que des drapeaux d'événements.

Un exemple de situation où il serait préférable d'utiliser un sémaphore plutôt que des drapeaux d'événements serait un cas semblable à celui de l'exercice 3. En effet, utiliser des drapeaux ainsi qu'une variable partagée (qu'il faut alors protéger avec des mutex) à incrémenter ou décrémenter est moins efficace que d'utiliser seulement un sémaphore qui peut posséder différentes valeurs et

être utilisé comment une variable partagée entre les différentes tâches qui l'utilisent. Un système de vente d'articles en ligne pourrait représenter un bon exemple. Lorsque plus d'articles sont mis disponibles, le sémaphore est directement incrémenté, et celui-ci est décrémenté chaque fois qu'un client se procure un article. Lorsque le sémaphore est à 0, il n'y a plus d'article disponible pour la vente.

Question d :

Donnez une courte appréciation du laboratoire (temps consacré, explications, clarté de l'énoncé, difficultés, etc).

Ce laboratoire était bien structuré. Il se faisait bien en deux semaine et était suffisamment complexe pour nous faire assimiler les concepts de base et revoir la matière rapidement vue en classe et en INF2610. Les explications données en classe ont été utiles pour démarrer le laboratoire et les attentes pour chaque numéro étaient claires. La principale difficulté rencontrée a été de suspendre les tâches à la fin de l'exercice 5. Quelque chose semblait nous échapper.