

Rapport d'Implémentation pour le Devoir 2 : Multithreading dans NACHOS

2023/2024

Magaye NDIAYE

Table des matières

I. Bilan.....	3
II. Points délicats.....	3
III. Limitations.....	4
IV. Tests.....	4

I. Bilan

Ce projet portait sur l'intégration avancée du multithreading dans les programmes utilisateurs sous NACHOS, couvrant non seulement la base du multithreading, mais aussi des aspects plus complexes de la gestion des threads.

Réalisations :

- Actions I.1 à I.7 : Implémentation réussie des fonctionnalités de base pour le multithreading au niveau utilisateur, y compris la création, la gestion du cycle de vie des threads, et leur intégration avec les mécanismes du noyau NACHOS.
- Action II.1 : Réalisation de l'action avancée de multithreading, englobant des aspects complexes de la gestion des threads, tels que la synchronisation avancée et le partage efficace des ressources entre threads.

Non réalisé :

- Les actions restantes de la Partie II n'ont pas été implémentées, laissant certains aspects avancés du multithreading, tels que la gestion des conditions de course dans des scénarios de charge élevée, pour une exploration future.

Problèmes rencontrés :

- Complexité accrue dans la gestion des interactions et de la synchronisation entre les threads noyau et utilisateur.
- Défis dans l'optimisation des performances pour des scénarios de charge élevée.

II. Points délicats

La partie la plus délicate a été l'implémentation de l'Action II.1, qui nécessitait une compréhension approfondie et une manipulation précise des mécanismes de synchronisation et de partage des ressources.

Stratégie de résolution :

- Développement de mécanismes de verrouillage sophistiqués et de variables de condition pour une gestion efficace des ressources partagées.
- Tests approfondis pour assurer la cohérence et l'efficacité de ces mécanismes dans différents scénarios.

III. Limitations

Avantages :

- Amélioration significative de la capacité de NACHOS à gérer le multithreading au niveau utilisateur.
- Implémentation de mécanismes de synchronisation avancés, augmentant la robustesse du système.

Inconvénients :

- Complexité et difficultés potentielles dans le débogage dues à l'intégration avancée de multithreading.
- Performances et stabilité sous charge élevée restent des zones d'amélioration.

Limitations principales :

- Incomplétude dans la couverture des fonctionnalités avancées de multithreading (Partie II restante).
- Besoin de tests supplémentaires pour évaluer et optimiser les performances dans des environnements multithread complexes.

IV. Tests

Les tests réalisés ont couvert les aspects de base et avancés du multithreading.

Stratégie de tests :

- Tests unitaires pour les nouvelles fonctionnalités de multithreading.
- Tests d'intégration pour évaluer l'interaction entre les threads utilisateurs et le noyau dans des scénarios variés.
- Tests de stress pour identifier les problèmes de performance et de synchronisation sous charge élevée.

Ces tests ont validé les implémentations de base et avancées, tout en soulignant les domaines nécessitant des améliorations futures, notamment en matière de performances et de stabilité dans des scénarios de charge élevée.