

# TP4 - Tâches utilisateurs et appels systèmes

## Programmation Système Avancée

Rudolf Höhn & Axel Fahy

January 22, 2016

1. **Qu'est-ce qui vous a le plus plus dans le développement de ce système d'exploitation ? Pourquoi ?**

La chose qui nous a vraiment le plus plus dans ce développement est le fait d'avoir créer de A à Z un système d'exploitation, et le fait que ce dernier fait tourner un Puissance 4. Nous pensons que c'est la chose la plus importante à garder dans ce TP, il faut que les étudiants aient ce sentiment de vraiment avoir accompli quelque chose de grand à la fin. Malgré le fait que ce fut un de nos plus dur travail pratique, ce fut sans le moindre doute, le plus intéressant.

2. **Qu'est-ce qui vous a le moins plus dans le développement de ce système d'exploitation ? Pourquoi ?**

Pas beaucoup de choses nous ont déplus dans ce TP, mais dans la partie 3, le développement du système du fichiers, il nous manquait une image "propre", sur laquelle nous pouvions nous baser pour développer les tools et être sûr qu'ils fonctionnaient. C'est un détail, mais le développement des tools et la partie PFS dans le kernel aurait été plus rapide.

3. **Quels seraient, selon vous, les 3 étapes suivantes à développer dans votre système d'exploitation ? Pourquoi ?**

Il y a pleins de choses à faire encore pour améliorer notre système d'exploitation. Il n'a pas d'utilisateurs, pas d'arborences de fichiers, pas de connexion réseau, et encore d'autres. Malgré cela, avant de s'emballer dans pleins de choses, nous pensons qu'intégrer un système de fichiers plus complexes semble être la prochaine étape. Lorsque cela sera fait, nous pourrons passer à l'amélioration de notre shell, pour qu'il puisse intégrer la navigation dans un système de fichiers à arborescence (type *FAT32*). Le shell devra aussi pouvoir créer des dossiers, des fichiers, mais des fichiers vides ne servent à rien, donc peut-être implémenter Vi ?

Après le développement de contrôleurs *FAT32*, la mise à niveau du shell, nous pourrons passer au développement de contrôleurs supplémentaires pour pouvoir monter des périphériques, comme par exemple une clé USB.

Cette liste reste peu exhaustive, mais nous pensons que l'objectif des 3 prochaines étapes serait de rendre le système d'exploitation de plus en plus "autonome".

4. **Sachant que les appels systèmes sont relativementement prohibitifs en terme de performance, comment feriez-vous pour proposer aux applications utilisateur un affichage efficace et performant ?**

En lisant quelques documents en ligne, nous sommes tombés sur le *Direct Rendering Manager (DRM)*. C'est un sous-système du kernel Linux qui permet d'interfacer les GPUs et d'exposer une API aux applications utilisateur pour interagir avec l'écran. Dans notre cas, un premier ajustement serait à faire : la gestion de l'écran ne devrait pas être effectuée par le CPU mais plutôt par un GPU. A partir de là, nous déchargerions le kernel des appels systèmes comme la paramétrer la position du curseur, écrire à la position du curseur, et "sous-traiter" ces tâches au GPU grâce à un *DRM*.