

**Programmation système**  
**Epreuve de contrôle continu N°1**  
**Durée 1h30 - 29 novembre 2010**

**Documents autorisés :** transparents et notes de cours uniquement. Tout appareil électronique est strictement interdit.

**Préambule**

Répondez aux questions de manière concise et précise. Evitez le hors-sujet, cela vous pénaliserait.

**Questions de compréhension (8 points)**

Chaque question compte 1 point. Répondre aux questions suivantes en respectant les consignes du préambule :

1. Lorsque l'on crée un système de fichiers on génère automatiquement l'ensemble des inodes. Expliquez pourquoi il n'est pas nécessaire de mémoriser le numéro d'une inode sur le disque.
2. Où est stocké le nom d'un fichier ? En vous aidant de votre réponse à cette question, expliquez comment le système accède au fichier /tmp/x
3. Justifiez le besoin de disposer d'un mode « noyau » et d'un mode « utilisateur »
4. Est-ce que tous les blocs utilisés pour mémoriser le contenu d'un fichier contiennent uniquement les données de ce fichier ? Expliquez pourquoi.
5. Dans quel fichier se trouve l'association entre le numéro d'utilisateur et le nom utilisateur ?
6. Citez une commande UNIX qui utilise les informations du fichier /etc/group, comment ?
7. Dans la table des processus y-a-t-il uniquement des processus non terminés ? Expliquez pourquoi.
8. Admettons que la commande *more* soit lancée à travers un terminal. Quelle est la valeur de sa variable « umask » ?

**Exercice sur les fichiers (6 points)**

Ecrire un programme *afficherCible* qui prenne en paramètre des chaînes de caractères (le nombre n'est pas fixé). Pour chacun des paramètres on vérifiera que le nom correspond à un fichier accessible en lecture et on affichera pour chaque lien symbolique les informations relatives au fichier lié et non pas celles relatives au lien lui-même. Dans le cas où le fichier lié est lui-même un lien symbolique, ce sont les informations de son fichier lié qu'il faudra afficher et ainsi de suite. Pour les autres fichiers passés en paramètre on affichera leur nom suivi du message « n'est pas un lien symbolique ».

*Attention vous ne devez utiliser que des primitives de la section 2 du manuel UNIX sauf éventuellement pour des conversions.*

**Exercice sur les processus (6 points)**

Ecrire un programme *executerFichier* qui prenne en paramètre des chaînes de caractères. Le 1<sup>er</sup> paramètre est un nom de commande et les autres correspondent aux paramètres de cette commande. Il faudra vérifier que le premier paramètre est bien un fichier exécutable et ensuite lancer son exécution de telle manière que si la commande écrit sur le standard de sortie (resp. d'erreur), l'écriture soit faite sur le fichier /tmp/output (/tmp/erreur). Enfin le processus correspondant à *executerFichier* récupérera le code de retour de la commande et l'affichera sur le fichier standard de sortie. On admettra que des informations intéressantes sont disponibles sur le /tmp/output jusqu'à la position 150 et que pour /tmp/erreur on veut conserver la totalité de l'existant.

*Attention vous ne devez utiliser que des primitives de la section 2 du manuel UNIX sauf éventuellement pour des conversions.*