

	Cycle ingénieur 1^{ère} année Examen de système d'exploitation <i>Jean-Philippe Attal, Stefan Bornhofen, Rachid Chelouah, Florent Devin</i>	
	<i>Matière : Système d'exploitation</i>	<i>Date : Juin 2014</i>
	Documents manuscrits autorisés	<i>Durée de l'épreuve : 2 heures</i>
	Ordinateur portable et documents électroniques interdits	<i>Nombre de pages du sujet : 1</i>

Exercice 1 (Cours - 5 points).

- Quelles sont les informations indispensables afin de gérer correctement un processus pour un système d'exploitation ?
- À votre avis, à quoi sert la mémoire virtuelle ? Quelles en sont les applications ?
- À quoi sert la ligne `#!/bin/bash` dans un script shell ?

Exercice 2 (Multiprocesus - 3 points).

- Écrire un programme C, qui provoque la création de deux processus, en plus du processus principal.

Exercice 3 (Script shell - 6 points).

- Écrire un embryon de script shell qui permet de traiter efficacement la ligne de commande. On supposera que la ligne de commande doit pouvoir réagir aux options suivantes :
 - h, --help : affiche de l'aide
 - r : exécution récursive
 - d rep : provoque l'exécution de la commande sur le répertoire rep
 Pour valider le fait que vous traitiez correctement la ligne de commande, vous ferez le traitement dans une fonction shell. L'invocation de cette fonction provoquera un affichage afin de montrer que vous avez détecté correctement les options de la ligne de commande. Si l'option -h ou --help est passée (quelque soit sa position dans la ligne de commande), il ne doit y avoir qu'un seul affichage celui d'aide, peu importe si d'autres options valides sont passées. Toutes options non valides passées à la ligne de commande doit être détectées, et affichées.

Exercice 4 (Multiprocesus et fichiers - 6 points).

Voici un exercice tiré du TD 4.

Pour manipuler des fichiers, il existe deux principaux modes d'accès : bas niveau, haut niveau. Il existe une façon "pratique" de savoir si on utilise une manipulation bas niveau, ou haut niveau : les manipulations bas niveau manipulent des *file descriptor* (fd) tandis que les manipulations haut niveau manipulent des FILE *. Parmi les fonctions bas niveau on trouve open, read, write, ...

Réalisez un programme qui effectue les actions suivantes :

- Ouverture bas niveau d'un fichier (existant, et préalablement rempli par du texte quelconque).
- Création d'un processus fils
 - Dans le processus père :
 - Lecture de deux octets, attente active de 3 sec (sleep), écriture de deux octets.
 - Attente de la terminaison du fils (wait).
 - Clôture du fichier.
 - Dans le fils :
 - Écriture de deux octets, attente active d'une seconde, lecture de deux octets.
 - Fin du fils

Que constatez vous ? Changez les différents temps d'attente, que constatez vous ? Obtenez vous le même résultat en utilisant des manipulations de fichier de haut niveau ? Que pouvez vous en déduire sur la relation père fils, notamment au niveau de l'héritage ?

Question 1 : Donner le code C correspondant à la version utilisant les manipulations bas niveau.