Examen - Systèmes et Réseaux 1
Licence 3 Informatique

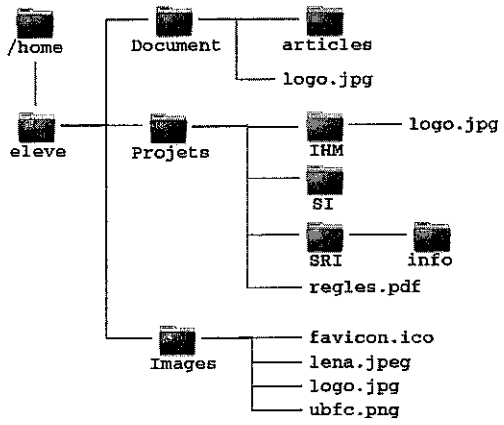
Durée : 2h. Documents personnels autorisés. Le barème est indicatif.

Exercice 1: Questions de cours : commandes (3 pts)

1. Donner la commande qui vous permettra d'obtenir votre nom de login.
2. Donner la commande qui vous permettra d'afficher tous les dossiers contenu dans votre dossier courant.
3. La commande `chmod 764 nomFichier` a été exécutée. Que pourra faire une utilisateur sur le fichier `nomFichier` ?
4. Un utilisateur veut forcer l'arrêt d'un processus par la commande `kill`. Comment va-t-il spécifier le processus à arrêter ?
5. Donner la commande permettant de copier dans un fichier `fich` la liste des éléments contenu dans le dossier `doss`.
6. Quelle est la différence entre les commandes `cp` et `mv` ?

Exercice 2: Système de fichiers (3 pts)
Le répertoire `eleve` est le répertoire courant. Les commandes suivantes sont effectuées successivement. Pour chacune d'entre elles, donner l'action réalisée et le message renvoyé sur la console.

1. `>cd Projets`
2. `>pwd`
3. `>cp ../Images/lena.jpeg SI`
4. `>mv regles.pdf SRI/`
5. `>ls ../Images>contenu`
6. `>wc contenu»contenu`
7. `>cat contenu`
8. `>chmod 644 regles.pdf`
9. `>ls -l` (ne donner que les dix premiers caractères et le dernier mot de chaque ligne)
10. `>find .. -name logo.jpg`
11. `>find . -type d`



Exercice 3: Programmation système (4 pts)
Sur un système Unix, sont gérés des fichiers dont le nom et/ou l'emplacement est susceptible de changer (plusieurs fois) au cours du temps (renommage/déplacement). On souhaite garder une trace des changements successifs. Pour cela, à chaque demande de renommage/déplacement de la forme `mv ancienChemin nouveauChemin`, le couple `(ancienChemin,nouveauChemin)` sera stocké dans un fichier (un fichier de stockage par fichier modifié, placé dans le sous-répertoire `.changes` du répertoire d'accueil).

1. Proposer une façon de nommer le fichier de stockage des changements d'un fichier permettant l'accès à ces changements à tout moment et indépendamment des modifications effectuées sur le nom du fichier.
2. Réécrire la commande mv pour prendre en compte ces fonctionnalités. Le répertoire `.changes` sera crée s'il n'existe pas. `mv` standard du système ?
3. Écrire un script shell/awk `retrouve <fic>` qui devra retrouver le ou les fichiers ayant ou ayant eu pour nom `fic` et afficher leur chemin d'accès actuel.
4. Écrire un script shell/awk `purge` qui va supprimer dans le répertoire `.changes` les fichiers de modifications des fichiers qui n'existent plus.

Exercice 4: Ordonnancement de commandes (3 pts)
Soit un système utilisant 9 formats de fichiers, suffixés par `a`, `b`, `c`, `d`, `e`, `f`, `g`, `h`, `i` et incluant les commandes de conversion de format suivantes : `a2b`, `c2d`, `e2f`, `bd2g`, `df2h`, `gh2i`. On suppose que ces commandes s'utilisent ainsi : pour les trois premières, `x2y fic.x fic.y` va transformer le fichier `fic.x` au format `x` en un fichier `fic.y` au format `y` ; pour les trois dernières, `xy2z fic.x fic.y fic.z` va produire le fichier `fic.z` à partir des fichiers au format `fic.x` et `fic.y`.

1. Pour les fichiers `essai.a`, `essai.c`, `essai.e`,
   (a) Dessiner le graphe de dépendences de la cible `essai.i`.
   (b) Donner le suite de commandes pour obtenir la cible `essai.i`.
2. Proposer un fichier Makefile pour automatiser l'obtention de n'importe quelle cible au format .i à partir des fichiers au format .a .c .e.
3. Si on a exécuté `make essai.i` et que l'on modifie `essai.h`, que se passe-t-il si on relance `make essai.i` une nouvelle fois ?

Exercice 5: Sémaphores (3 pts)
Soit une variable X et deux sémaphores $S1$ et $S2$ dont les valeurs initiales sont données par : `Int X = 0; Sem-Init(S1, 2); Sem-Init(S2, 1);`.

On considère trois processus concurrents $P1$, $P2$ et $P3$ définis comme suit :

Processus P1	Processus P2	Processus P3
Begin	Begin	Begin
P(S1) ;	P(S1) ;	P(S1) ;
P(S2) ;	X = X * X ;	P(S2) ;
X = X * 10 ;	V(S1) ;	X = X + 10 ;
V(S1) ;	V(S1) ;	V(S2) ;
End	V(S2) ;	End
	End	

Quelles sont les valeurs possibles de X ? Expliquer comment vous obtenez les différentes valeurs. Préciser également les cas de blocage de l'ensemble des processus.

Exercice 6: Diffusion simple de valeur par n processus (4 pts)
On considère $n+1$ processus $P_0, P_1, \ldots, P_n$ partageant un tampon ou buffer B. B peut contenir une seule valeur de type quelconque T. Le processus P_0, appelé diffuseur, produit et place dans le buffer B une valeur V (de type T) qui doit être lue par tous les autres processus $P_i, i = 1, 2, \ldots, n$, appelés récepteurs. Chaque valeur produite et déposée dans le buffer B doit être lue par tous les récepteurs avant la production et le dépot de la valeur suivante par le processus diffuseur P_0. Une valeur déposée dans B ne peut être lue qu'une seule fois par un processus récepteur.

1. Expliquer, en utilisant des sémaphores, les règles de synchronisation des processus $P_0, P_1, \ldots, P_n$ pour réaliser la diffusion d'une valeur de P_0 vers les autres processus.
2. Écrire à l'aide de sémaphores un algorithme (pseudo code) pour les processus $P_0, P_1, \ldots, P_n$.
3. En vous basant sur l'exemple de trace d'exécution de processus étudié en TD, illustrer votre solution pour le cas où $n = 3$, c'est-à-dire qu'on a un processus diffuseur P_0 et 3 processus récepteurs P_1, P_2, P_3. Vous devez montrer les évolutions des variables et des sémaphores (valeurs et files d'attente) pour les évènements ou arrivées de processus suivants :
   1. Exécution du Processus Récepteur P_2
   2. Exécution du Processus diffuseur P_0
   3. Exécution du Processus Récepteur P_1
   4. Exécution du Processus Récepteur P_3
   5. Exécution du Processus diffuseur P_0
   6. Exécution du Processus Récepteur P_1
   7. Exécution du Processus Récepteur P_2
   8. Exécution du Processus Récepteur P_2
   9. Exécution du Processus Récepteur P_3
   10. Exécution du Processus diffuseur P_0
   11. Exécution du Processus diffuseur P_2
   12. Exécution du Processus diffuseur P_0
   13. Exécution du Processus diffuseur P_1