

# SYSTÈME

## Devoir Surveillé n°1

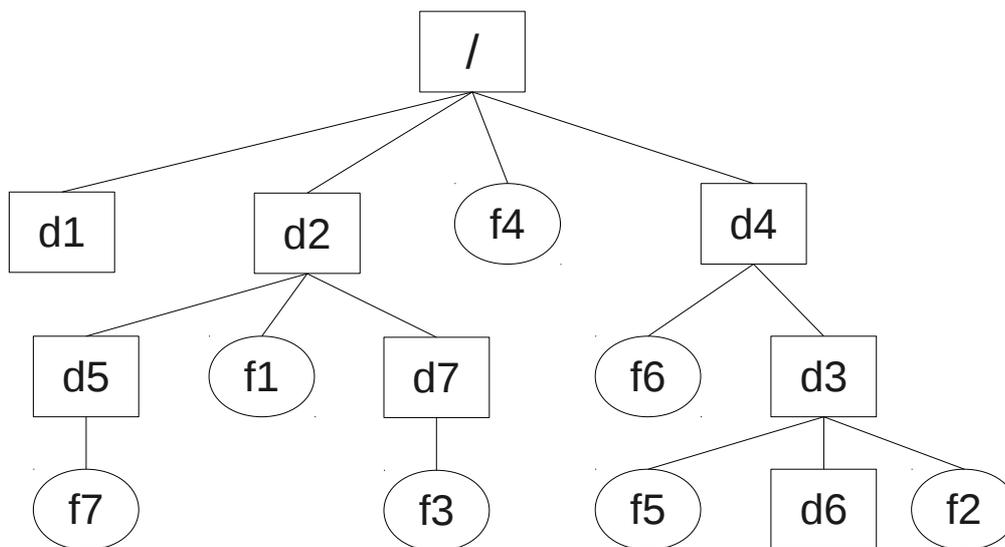
date : 14 octobre 2011, 16h45  
 feuille A4 manuscrite personnelle autorisée.

durée : 1 heure  
 barème : P1/10, P2/3, P3/7.

<b>NOM, Prénom :</b>	<b>Groupe :</b>
----------------------	-----------------

### Partie 1: Fichiers, répertoires et commandes (30 minutes, 10 points)

Voici un arbre de fichiers. Les ovales représentent des fichiers et les rectangles représentent des répertoires.



a) **Donnez le nom absolu de :**

f7	
f4	

b) **Vu de d5, donnez un nom complet relatif de :**

f1	
f6	
f4	
f7	

c) **On se place dans d7. En donnant uniquement des chemins complets relatifs donnez une commande pour :**

supprimer f3	
lister d5	
copier f1 dans d4	

d) On se place dans le dossier d2

- après exécution de la commande : `mv ../d4/d3/f2 ..`

dans quel dossier se trouve le fichier f2 :

- après exécution de la commande : `mkdir ../d1/d8 d9`

dans quel dossier se trouve d9 :

- e) Cette question est complètement indépendante des précédentes. Soit un répertoire où se trouvent les fichiers suivants :

`arbre arbre.o arbre.lst brico.txt forum info.lst`

Quel est ou quels sont les paramètres les plus courts possible, écrit(s) à l'aide des jokers \*, ? et [] de bash, qu'il faut fournir à la commande `rm` pour supprimer ensemble en une fois seulement certains des fichiers de cette liste, ceux qui sont indiqués ci-dessous, mais pas les autres ? Toutes ces questions sont indépendantes. Il n'y a pas d'ordre de difficulté croissante.

fichiers à supprimer avec un seul appel à <code>rm</code>	paramètre(s) le(s) plus court(s) de <code>rm</code>
<code>brico.txt info.lst</code>	
<code>arbre forum</code>	
<code>arbre.lst brico.txt info.lst</code>	
<code>arbre arbre.o arbre.lst</code>	
<code>arbre.o brico.txt</code>	
<code>arbre arbre.o arbre.lst brico.txt</code>	
<code>arbre.o brico.txt forum info.lst</code>	

NOM, Prénom :

Groupe :

## Partie 2: Processus (10 minutes, 3 points)

Soit ce listing, auquel on a enlevé de nombreuses autres lignes, notamment la dernière ligne qui contient la commande qui l'a affiché. Elle a été lancée à 17h54 environ.

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1587	1574	0	Oct03	?	00:00:00	/usr/sbin/winbindd
daemon	1470	1	0	Oct03	?	00:00:00	atd
nerzic	26352	26351	0	13:53	pts/0	00:00:00	-bash
nerzic	26493	26352	0	14:10	pts/0	00:00:27	x-povray -iisocacti.pov
nerzic	26502	26352	18	17:52	pts/0	00:00:23	x-povray -iskyvase.pov
nerzic	26511	26352	02	17:53	pts/0	00:00:01	gedit optics.pov
nerzic	26515	26352	33	17:53	pts/0	00:00:04	x-povray -ilandscape.pov

Quelle est cette commande avec ses options qui a permis de voir ce listing ?

Comment l'un des programmes x-povray a-t-il été lancé dans le terminal ? (Choisissez-en un)  
Remarque : ce programme est lancé de manière à ne rien écrire sur l'écran, mais tout dans stats.

Apparemment, l'un des processus de calcul semble planté ou stoppé. Quels sont les deux indices permettant d'affirmer cela ? Continuez l'exercice même si vous ne trouvez pas pourquoi.

Ce processus a été lancé en arrière-plan dans le terminal actuel. Quelle est la commande qui permet de **connaître l'état des processus** d'arrière-plan et savoir s'ils « vont bien » et quelle est la colonne qu'il faut examiner ?

Quelle commande complète doit-on saisir pour **supprimer le processus** planté de la question au dessus ? Si vous ne savez pas lequel, choisissez l'un des x-povray.

## Partie 3: Redirections et tubes (20 minutes, 7 points)

Le fichier **livres** contient des informations sur des ouvrages : ils sont décrits par le nom de l'auteur, le titre, l'année de parution, sa nature (bd, sf, roman...) et le nombre d'exemplaires possédés. Les champs sont bien alignés pour la lecture et le fichier n'est pas classé. Voici un extrait de ce fichier :

Hergé	Tintin au Tibet	1950	bd	3
Zola	La curée	1871	roman	5
Herbert	Le Messie de Dune	1969	sf	4
Rifflet	Unix	1998	tech	12

<b>Hergé</b>	<b>Tintin au congo</b>	<b>1931 bd</b>	<b>16</b>
<b>Eco</b>	<b>Le nom de la rose</b>	<b>1980 roman</b>	<b>7</b>
<b>Herbert Dune</b>		<b>1965 sf</b>	<b>4</b>
...etc...			

Le but de l'exercice est de décrire ce que font les commandes suivantes. Dans le cas d'une erreur, il faudra expliquer le problème et proposer une correction, si c'est possible, pour que la commande fonctionne. Dans le cas où la commande est correcte, il faut expliquer qualitativement ce qu'elle fait ou donner des exemples des informations qu'elle produit.

Que fait la commande : **sort -k5 < livres | more**, est-ce qu'elle marche bien ?

Que fait la commande : **cut -d' ' -f2 < livres**, est-ce qu'elle marche bien ?

Que fait la commande : **egrep 'Hergé' < livres | wc < livres**

Que fait la commande : **livres > egrep 'Dune' > dune | wc -l**

Que fait la commande : **more livres | egrep -v ' 18.. ' > more**