

# SYSTÈMES D'EXPLOITATION

## Devoir Surveillé n°2

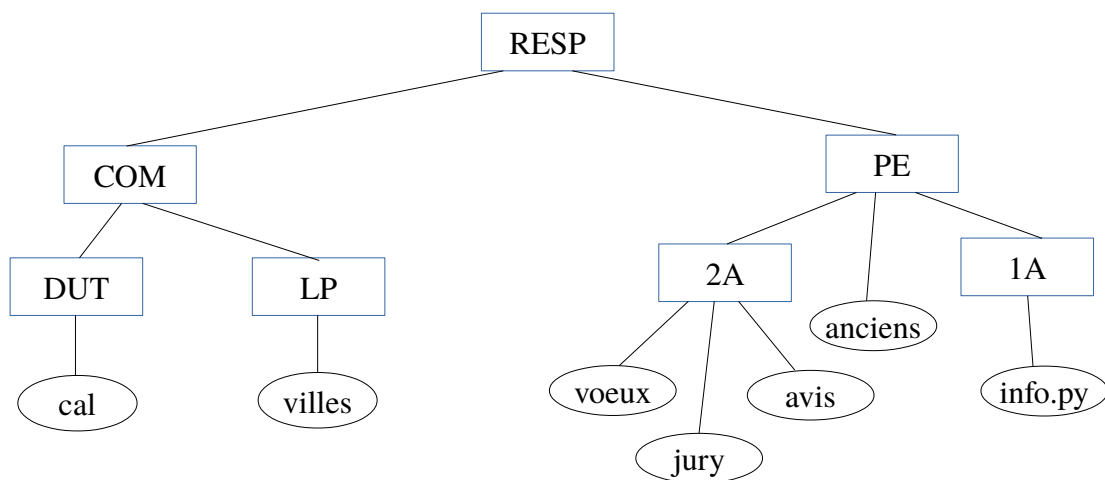
date : 22 janvier 2014 08h00  
feuille A4 manuscrite personnelle autorisée.

durée : 1 heure  
barème indicatif : P1/8, P2/12.

**Lire les questions attentivement et complètement.**

### Partie 1: Protection des fichiers (24 minutes, 8 points)

Voici une partie de l'arborescence du système de fichiers du département informatique.



On s'intéresse à ce que peuvent faire certains utilisateurs de ce système. Voici les noms de ces utilisateurs et les groupes auxquels chacun appartient :

nom	groupe
brigitte	adcom
marie	info, scol
erwann	scol

Voici un ensemble d'informations relatives aux répertoires de cette arborescence (résultat de l'exécution de la commande ls avec les options appropriées).

drwxr-xr-x	1	brigitte	adcom	02	nov	21	14:35	COM
d-wxr-xr-x	1	brigitte	adcom	02	sep	26	17:25	PE
drwxrwxrw-	1	erwann	adcom	02	oct	17	15:44	COM/DUT
drw--wxr-x	1	marie	scol	02	nov	28	18:43	COM/LP
d-wxrwxrw-	1	brigitte	info	02	nov	27	13:34	PE/1A
drwx-wxr-x	1	erwann	scol	02	sep	19	17:43	PE/2A

Ensuite, voici un ensemble d'informations relatives aux fichiers de cette arborescence

```
-rwx-w-rw-   1  erwann    info   02 dec 11 11:10   anciens
-r--rw-r--   1  brigitte  info   02 sep 22 19:20   avis
-rwxr--r--   1  brigitte  scol   02 sep 27 16:53   cal
-rwx-wxr-x   1  erwann    scol   02 dec 11 11:10   info.py
-rwx--xrw-   1  marie     scol   02 nov 14 18:20   jury
-rwxrw-rw-   1  erwann    info   02 sep 16 17:12   villes
-rwxrw-r-x   1  marie     adcom  02 dec 13 19:23   vœux
```

On s'intéresse au problème : qui est autorisé à exécuter telle commande dans tel répertoire .

Répondre aux questions directement sur la feuille de réponse de la partie 1, **en respectant rigoureusement les étapes montrées dans l'exemple fictif de la première ligne.**

**On demande d'appliquer les règles du modèle de base vu en cm, td et tp, sans chercher à supposer un 'pseudo' fonctionnement 'logique', ou inventer une hiérarchie des droits selon la catégorie, par exemple : le propriétaire aurait tous les droits de toutes façons !**

## Partie 2: Scripts Bash (36 minutes, 12 points)

Préparez vos réponses sur papier de brouillon avant de les recopier au propre. Les ratures sales seront pénalisées.

**Proposez un script qui réponde le mieux possible à ce cahier des charges. Faites ce que vous pouvez. Tout élément écrit correct et pertinent rapporte des points.**

On se trouve dans un dossier dans lequel il y a beaucoup de fichiers, certains étant assez anciens. On voudrait écrire un script qui supprime les fichiers qui sont plus anciens qu'une certaine date qu'on passe en paramètre.

Comment savoir si un fichier est plus ancien qu'une certaine date : le système Unix représente les dates en interne sous la forme de nombre de secondes écoulées depuis le 1<sup>er</sup> janvier 1970. Par exemple le matin de ce devoir à 8h, c'était 1 390 374 000.

Tout fichier est daté de cette manière. La commande `stat -c %Y fichier` affiche la date de modification du fichier sous la forme d'un nombre entier. En mettant `%y` au lieu de `%Y` on a la date au format lisible :

```
prompt$ stat -c %Y archive.log
1311607740
prompt$ stat -c %y archive.log
2011-07-25 17:29
prompt$
```

La commande `date -d 'AAAA-MM-JJ HH:MM' +%s` affiche la date indiquée sous forme d'un nombre de secondes, voici un exemple :

```
prompt$ date -d '2014-01-22 08:00' +%s
1390374000
```

```
prompt$
```

Vous avez donc tous les éléments pour écrire un script qui supprime tous les fichiers plus anciens que la date qu'on fournit en paramètre, et il écrit un message pour chaque fichier supprimé.

```
prompt$ SupprAnciens '2012-12-31 23:59'  
Je supprime archive.log (2011-07-25 17:29)  
Je supprime test.txt (2013-04-01 12:54)  
prompt$
```

Ce script doit donc convertir la date fournie en paramètre sous forme d'un nombre puis passer en revue tous les fichiers, et pour chacun obtenir sa date sous forme d'un nombre aussi et les comparer (par  $\leq$ ). S'il décide de supprimer un fichier, alors il affiche sa date sous forme lisible.

On aimerait aussi que ce script puisse éventuellement recevoir un paramètre supplémentaire mais optionnel afin de faire un intervalle de dates :

```
prompt$ SupprAnciens '2013-01-01 00:00' '2013-12-31 23:59'  
Je supprime test.txt (2013-04-01 12:54)  
prompt$
```

Bien comprendre que s'il y a le paramètre de plus, alors on supprime tous les fichiers entre ces deux dates ; s'il est manquant, on supprime les fichiers plus anciens que la date indiquée.

On voudrait aussi qu'on puisse rajouter une option `--simulation` permettant de ne pas réellement supprimer les fichiers, mais seulement indiquer lesquels seraient supprimés si on le mettait pas l'option :

```
prompt$ SupprAnciens --simulation '2012-12-31 23:59'  
Je supprimerais archive.log (2011-07-25 17:29)  
Je supprimerais test.txt (2013-04-01 12:54)  
prompt$
```