

SYSTÈMES D'EXPLOITATION

Devoir Surveillé n°2

date : 28 novembre 2008, 15h45
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)

On se situe dans un répertoire Linux dans lequel la commande `ls -l` affiche ceci :

<code>drwxrwx---</code>	1	claudio	etu	02 sep 20 13:16	projets
<code>drwxr-xr-x</code>	1	jacques	ens	02 sep 23 14:25	docs
<code>drw-rwxr--</code>	1	lucas	etu	02 sep 27 15:34	exos
<code>-rw-r-----</code>	1	claudio	ens	02 sep 29 16:43	texte

Ensuite, sur ces répertoires, voici ce qu'affiche `ls -l projets docs exos` :

<code>-rw-rw-r--</code>	1	lucas	ens	02 sep 26 17:52	projets/desc
<code>-rwxr-x---</code>	1	kristen	admin	02 sep 24 18:01	projets/init
<code>-rw-rw----</code>	1	kristen	ens	02 oct 12 19:10	docs/aide
<code>-rw-r--r--</code>	1	gregory	ens	02 oct 21 20:29	exos/td2

NB : inutile de poser une question sur ce qui est écrit ci-dessus, il n'y a aucune erreur dans l'énoncé. Si quelque chose vous paraît bizarre, c'est vous qui avez des lacunes.

Pour vous aider, vous pouvez faire un schéma sur le brouillon.

On s'intéresse à ce que peuvent faire certains utilisateurs de ce système. Voici les groupes auxquels ils appartiennent :

nom	groupe
claudio	ens
jacques	admin
gregory	ens
lucas	etu
kristen	etu

Encore une fois, il n'y a aucune contradiction avec les listings de `ls -l`. Inutile de poser une question.

Répondre aux questions de la feuille de réponse n° 1.

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.

...page suivante

On voudrait disposer d'une sorte d'agenda. L'idée générale est de noter des événements dans un fichier agenda. Chaque événement a une date. Lorsque la date actuelle est égale à celle de l'événement, il y a une action pour prévenir l'utilisateur : envoyer un message ou un fichier par mail.

Le DS consiste à écrire deux script : l'un pour ajouter un événement à l'agenda, l'autre pour gérer l'agenda et regarder s'il y a des événements aujourd'hui.

Les deux scripts à écrire travaillent sur un fichier commun appelé agenda. Ce fichier contient des lignes contenant plusieurs mots (des caractères séparés par des espaces) :

- date de l'événement au format `jj mm aaaa` les mots sont séparés
- adresse mail de l'utilisateur à prévenir au format `nom@domaine`
- nom complet du fichier ou mots du message à lui envoyer

Voici un exemple de contenu :

```
28 11 2008 info1@iut-lannion.fr devoir surveillance systeme a 15h45
27 11 2008 info1@iut-lannion.fr /home/pierre/systeme/devoir2007.pdf
01 01 2009 ensinfor@iut-lannion.fr /home/pierre/voeux.txt
```

On peut voir que la première ligne contient un message à envoyer tandis que les deux dernières lignes contiennent un nom complet de fichier. On peut dire qu'on a affaire à un message si ce n'est pas un nom de fichier qui existe.

a) script Ajouter

Le script `Ajouter` permet à l'utilisateur de rajouter un événement dans l'agenda. On lui fournit la date en paramètres : jour mois année. Ensuite il demande l'adresse mail du destinataire, puis le message ou bien le nom du fichier. Il ne fait pas de tests sur ces paramètres et réponses, il se contente d'ajouter la ligne adéquate dans le fichier agenda.

- Ecrire le script dans la **feuille de réponse n°2** après l'avoir complètement écrit au brouillon.

b) script Verification

Le script `Verification` reçoit l'agenda en entrée et regarde s'il y a une ligne qui correspond à la date du jour, exécute l'action prévue. Ce script doit faire la différence entre un nom complet de fichier et un simple message.

Le script `Verification` supprime aussi toutes les lignes de l'agenda qui correspondent à la date actuelle ou qui sont périmées. On fait cela en écrivant toutes les autres lignes sauf celles qu'il faut éliminer vers un fichier temporaire qu'on renomme une fois que c'est fini.

Pour savoir si un événement est périmé, il faut programmer un ensemble de tests sur les dates.

La commande `date '+%d %m %Y'` affiche la date actuelle au format exact de l'agenda. On peut avoir seulement le jour avec `+%d`, seulement le mois avec `+%m`, seulement l'année avec `+%Y`.

La commande `mail destinataire` envoie un mail au destinataire avec le texte qui lui arrive sur son entrée standard. On admet que la commande `mail` renvoie un code d'erreur non nul si le destinataire n'existe pas. Il ne sera pas répondu aux questions sur ce que signifie ces deux phrases et sur ce qu'on peut en faire.

Le script doit envoyer un message à `root@iut-lannion.fr` s'il y a une erreur avec le destinataire.

- Ecrire le script dans la feuille de réponse n°2 après l'avoir complètement écrit au brouillon.
- Montrer comment on doit lancer `Verification` pour faire la suppression des événements du jour et ceux qui sont périmés.

Faites ce que vous pouvez – toute ligne bash pertinente est prise en compte dans la notation.

SYSTÈMES D'EXPLOITATION - Devoir Surveillé n°2 FEUILLE DE REPONSE POUR LA PARTIE 1

Nom, Prénom :	Groupe :
---------------	----------

N'écrire ni au crayon à papier, ni au bic rouge.

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

Remplir le tableau suivant en mettant « oui » ou « non » dans les cases correspondantes. En ligne, il y a les noms des utilisateurs, en colonne il y a des commandes tentées par ces utilisateurs.

NB: les commandes sont toutes indépendantes, elles sont tapées dans le répertoire du début.

	<code>more texte</code>	<code>ls projets</code>	<code>cd docs</code>	<code>rm projets/desc</code>	<code>cp texte docs</code>	<code>mv exos/td2 projets</code>	<code>cp docs/aide projets</code>	<code>./projets/init</code>
<code>claude</code>								
<code>jacques</code>								
<code>gregory</code>								
<code>lucas</code>								
<code>kristen</code>								

Faites attention à ne pas tomber dans les pièges vus en TD et en TP.

SYSTÈMES D'EXPLOITATION - Devoir Surveillé n°2 FEUILLE DE REPONSE POUR LA PARTIE 2

Nom, Prénom :

Groupe :

N'écrire ni au crayon à papier, ni au bic rouge.

Partie 2: Scripts Bash (24 minutes, 8 points)

Script Ajouter

Script Verification

