

SYSTÈMES D'EXPLOITATION

Devoir Surveillé n°2

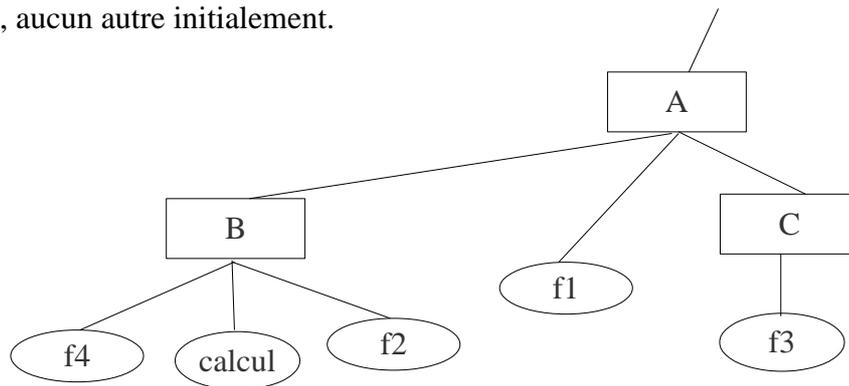
date : 25 novembre 2011, 15h45
feuille A4 manuscrite personnelle autorisée.

durée : 1 heure
barème indicatif : P1/7, P2/6, P3/7.

Lire les questions attentivement et complètement.

Partie 1: Protection des fichiers (21 minutes, 7 points)

Voici une partie de l'arborescence du système de fichiers du département informatique. Il n'y a que ces fichiers, aucun autre initialement.



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	groupes
p1	g1
p2	g2
p3	g2

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

drwxr-x--x	1	p1	g2	02	sep	23	14:25	A
drwxrwx--x	1	p1	g2	02	sep	23	14:25	B
drwxr-x-wx	1	p2	g3	02	sep	27	15:34	C

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

-rwx--xr--	1	p2	g2	02	nov	12	16:10	calcul
-rwxr--r--	1	p1	g1	03	sep	26	17:48	f1
-rwxr-----	1	p2	g3	02	sep	27	17:52	f2
-rwxr--r--	1	p2	g2	09	oct	12	23:10	f3
-rwx--xr--	1	p2	g2	02	oct	15	19:10	f4

1. 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 a tous les droits de toute façon !

2. On souhaite créer deux nouveaux répertoires, D et E avec les accès suivants :

Répertoire D : tout le monde (p1, p2, p3, étudiants) pourra y aller ; p2 et p3 pourront également lister le contenu seul p1 pourra tout y faire, y compris déposer un fichier.

Répertoire E : p1 et p2 pourront seulement lister le contenu, p3 pourra tout y faire, les étudiants auront uniquement le droit d'y déposer un fichier.

Pour chaque dossier, proposer une solution en terme de protections : quel utilisateur en tant que catégorie U ? quel groupe en tant que catégorie G ? quels sont les droits associés pour les catégories U, G et O ? Si ce n'est pas possible, expliquer pourquoi.

Répondre sur la feuille de réponses

Partie 2: Filtrage sur un fichier de données (18 minutes, 6 points)

Voici un extrait d'un très grand fichier de données appelé **paye** sur des salaires versés à des employés. Chaque ligne décrit un des versements. On trouve le nom de l'employé, le service dans lequel il travaille, le mois sur deux chiffres (février = 02), l'année, le montant du versement (salaire mensuel). Les informations sont séparées par des : . Les données montrées pour exemple ci-dessous sont totalement fantaisistes.

```
Albert Henri:publicité:02:2011:8768
Marcel Pascal:production:04:2010:4523
Albert Henri:publicité:03:2011:8771
Marcel Pascal:conception:01:2011:5987
André Jean:publicité:11:2010:6251
Albert Henri:publicité:04:2011:8760
etc... etc...
```

Sans se baser sur ces données uniquement puisqu'il s'agit d'un extrait, donner les commandes Unix permettant de faire les opérations suivantes. Faites ce que vous pouvez à chaque fois. Expliquez ce qu'on obtient si ce n'est pas le résultat demandé.

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

- Afficher les lignes des 5 plus grands versements pour l'année 2010 (lignes entières, celles dont le salaire est le plus élevé en 2010).
- Afficher le nombre d'employés (différents) travaillant dans le service publicité en 2011.
- Comment peut-on s'assurer qu'il y a bien les 12 mois de salaire par an pour chaque employé ? C'est à dire, prenons Albert Henri par exemple en 2011, comment peut-on faire pour vérifier qu'il a été payé chaque mois ? Et comment généraliser à tous les employés ? Afficher les années et les noms des employés pour lesquels ce n'est pas vérifié.
- Afficher les lignes pour lesquelles le salaire est supérieur ou égal à 6000.
- Afficher les noms des employés qui ont changé de service entre 2010 et 2011. C'est le cas de Marcel Pascal dans l'extrait montré.

Partie 3: Scripts Bash (21 minutes, 7 points)

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

On demande de programmer un script bash qui simule le fonctionnement d'une écluse. On rappelle que c'est un système à deux portes dans lequel une seule porte peut être ouverte en même temps. On ne simule pas le niveau d'eau, les bateaux etc, seulement le fait que les deux portes sont ouvertes ou fermées.

```
prompt% ecluse ouvrir portel
portel ouverte, porte2 fermée
prompt% ecluse ouvrir porte2
on ne peut pas ouvrir la porte2 car la portel est ouverte
prompt% ecluse fermer portel
portel fermée, porte2 fermée
prompt% ecluse ouvrir porte2
portel fermée, porte2 ouverte
prompt% ecluse etat
portel fermée, porte2 ouverte
prompt%
```

En résumé, le script reçoit un premier paramètre qui vaut soit etat, soit ouvrir, soit fermer. Dans les deux derniers cas, il faut aussi donner le nom de la porte concernée.

Il faut parvenir à mémoriser l'état des portes entre deux exécutions du script. Pour cela, il y a plusieurs solutions. Par exemple, on peut avoir deux fichiers appelés portel et porte2 dont on regarde s'ils existent ou pas : si portel existe, c'est que la porte1 est fermée, pareil pour porte2. Ou inversement. Ou alors un seul fichier qui contient le nom de la porte ouverte.

Votre script ecluse devra faire tous les tests et changements nécessaires afin de fonctionner comme demandé. Si vous n'arrivez pas à le faire comme demandé, faites comme vous pouvez. Tout ce que vous écrivez de correct et utile est pris en compte dans la notation.