INFO-F201 - Système d'exploitation
Projet 1 - Shell scripting

Année universitaire 2018 - 2019

Énoncé

Vous avez mis en place un pool de photos pour divers photographes amateurs. Lors du premier jet, chaque photographe amateur s'est vu attribué un compte utilisateur (associé à un groupe éponyme) sur la machine hébergeant le pool ainsi qu'un répertoire personnel dans le pool, dont le nom peut différer du compte utilisateur, l'administration de chaque répertoire étant laissé à la charge des photographes (i.e. l'arborescence ne suit pas de pattern prédéfini). L'unique contrainte imposée est le préfixage de chaque photo par un timestamp POSIX indiquant la date de la prise de la photo. Ainsi une photo nommée tux.png, prise le 22 octobre 2018 à $11:26:27$ aura pour nom de fichier :

`1540200387_tux.png`

Malheureusement, une mauvaise gestion des droits par les utilisateurs et l'arborescence hétéroclite entre les répertoires du pool rendent le partage des photos très compliqué.

Afin de faciliter le travail de tout le monde, vous envisagez une seconde version de l'architecture du pool avec les spécifications ci-dessous. Dans ce qui suit, `photo_admin` identifie l'utilisateur en charge de l'administration du pool et `photo_group` identifie le groupe auxquel appatiennent l'ensemble des utilisateurs du pool.

Spécifications

Les propriétaires des fichiers et répertoires sont définis comme suit :

1. chaque répertoire du pool doit appartenir à `photo_admin`,
2. chaque fichier photo doit appartenir à l'utilisateur qui a pris la photo (ie l'actuel propriétaire du fichier),
3. l'ensemble des fichiers et répertoires doivent appartenir au groupe `photo_group`.

Les droits sur chaque répertoire et fichier est défini comme suit :

1. l'arborescence ne peut être modifiée que par l'utilisateur `photo_admin`,
2. les fichiers ne peuvent être modifiés que par leur propriétaire,
3. les fichiers et répertoires de l'arborescence doivent être accessibles et lisibles par l'ensemble des utilisateurs du pool,
4. les fichiers et répertoires doivent être inaccessibles au reste des utilisateurs de la machine.

Une nouvelle arborescence est définie. Ces caractéristiques techniques sont reprises ci-dessous :

1. le répertoire personnel dans le pool doit avoir le même nom que l'utilisateur
2. le chemin relatif d'un fichier par rapport à la racine du pool doit réprésenter la date de prise de vue. Pour ce faire l'arborescence de chaque répertoire personnel est définie sur trois niveaux. Les répertoires de premier niveau représente l'année de la prise de vue (sur quatre chiffres), ceux du deuxième niveau le mois (sur deux chiffres) et le troisième niveau le jour (sur deux chiffres également). Ainsi, la photo tux.png prise le 22 octobre 2018 doit se trouver dans le répertoire :

`/chemin/vers/pool_v2/utilisateur/2018/10/22/tux.png`

Travail demandé

Il est demandé l'écriture du script qui permettra la migration de la V1 du pool vers la V2. Pour se faire, vous devez implémenter un parcours récursif dans le pool_v1 et pour chaque photo, créer, au besoin l'arborescence dans le pool_v2, et déplacer la photo dans le nouveau pool. Le synopsis du script est le suivant :

```
lipap /path/to/pool_v1 /path/to/pool_v2 photo_admin_uid photo_group_gid
```

Le script doit s'assurer l'existence des deux pools, si le répertoire du pool_v2 n'existe pas, il doit le créer. Il doit également s'assurer que le nom du répertoire personnel dans le pool_v1 correspond à un utilisateur existant sur la machine, sans quoi le nom du répertoire correspondant dans le pool_v2 sera nommé en utilisant le propriétaire du répertoire personnel en cours de traitement. Enfin, le pool_v1 ne doit pas être modifié pendant l'opération de migration.

Partie facultative

Pour les étudiants désireux d'aller plus loin, les fonctionnalités suivantes peuvent être implémentées.

1. Rendre le script multithread.
2. Rendre le script à même de gérer les options. Les options suivantes peuvent être de bons candidats à l'implémentation :
   - `-g photo_group_gid` indique le groupid du groupe `photo_group`. Si cette option n'est pas renseignée, le script utilise le gid du groupe users. **L'implémentation de cette option entraîne le retrait du paramètre** `photo_group_gid` **dans le synopsis du script**.
   - `-u photo_admin_group` indique le userid de l'utilisateur `photo_admin`. Si l'option n'est pas renseignée, l'uid de l'utilisateur courant est utilisé. **L'implémentation de cette option entraîne le retrait du paramètre** `photo_admin_uid` **dans le synopsis du script**.
   - `-m` active le `move mode`. Les photos ne sont plus copiées, mais déplacées.
   - `-d` indique si les répertoires traités dans le pool_v1 doivent être effacés après opération. **L'activation de cette option entraîne de facto l'activation du move mode**.
   - `-U utilisateur1[,utilisateur2[,utilisateur3[,...]]]` spécifie une liste d'utilisateurs séparés par une virgule dont le répertoire doit être migré vers la V2, sans cette option l'ensemble des répertoires contenus dans la V1 doivent être traités.
3. Effectuer une vérification sur la validité du timestamp associé à chaque photo. En cas de timestamp invalide, la photo est placée dans un répertoire erreur créé pour l'occasion à la racine du répertoire personnel de l'auteur dans la photo dans le pool_v2.
4. Gérer les conflits possibles entre photographies. Avant la copie/déplacement du fichier, le script s'assure qu'aucun fichier précédemment déplacé/copié ne porte le même nom. Sans quoi, le fichier en cours de traitement est copié/déplacé dans le répertoire conflits (créé à la racine du répertoire personnel de l'auteur dans le pool_v2) en respectant la convention de nommage suivante :

   ```
   YYYYMMDD_hhmmss_nomdufichier_a.png
   ```

   Un lien symbolique vers le fichier préexistant doit être également créé dans le répertoire conflits en respectant la convention suivante :

   ```
   YYYYMMDD_hhmmss_nomdufichier_b.png
   ```

Aide

La commande `date` peut être utilisée pour gérer de manière efficace les timestamps ainsi que leur conversion en différents formats dont le descriptif complet est donné dans la page de manuel.

L'option `-p` est une option intéressante de la commande `mkdir` dans la mesure où son utilisation permet en un seul appel, la création d'une arborescence linéaire. Ainsi un appel à

```
mkdir -p ~/repertoire_existant/repertoire_inexistant/repertoire_a_creer
```

créer l'ensemble des repertoires nécessaire à la création de `repertoire_a_creer`, donc dans l'exemple précédent le répertoire `repertoire_inexistant`. Plus d'informations sont disponibles dans la page de manuel de la commande.

Consignes

Vous pouvez utiliser toutes les commandes vues aux travaux pratiques ainsi que celles citées dans l'énoncé. La clarté, la lisibilité, la qualité d'implémentation et l'efficacité pourront servir de critères d'évaluation. Commentez donc intelligemment votre code et implémentez ce qui est demandé de manière claire et précise. Le shell pour lequel le script doit tourner pour ce projet est bash.

Toute question concernant l'énoncé doit être adressée à Guillerme Duvillié : `gduvilli@ulb.ac.be`.

Remises du projet

Ces consignes sont à respecter scrupuleusement, sous peine de non correction du projet.

1. Votre projet doit indiquer votre nom et votre numéro de matricule en commentaires en dessous de la ligne de spécification du shell, au début de chaque fichier.
2. Votre projet doit être dactylographié. Les projets écrits à la main ne seront pas corrigés.
3. Votre code doit être commenté.
4. Si votre code ne s'exécute pas, votre projet ne sera pas corrigé.
5. Le plagiat sera sanctionné comme il se doit.
6. Aucune défense orale du projet n'est prévue. Toutefois, l'étudiant pourra être convoqué pour fournir des explications dans les semaines qui suivent la remise.
7. Une version électronique est à remettre sur l'université virtuelle avant le 9 décembre $23:59$.