

TP Noté Systèmes Informatiques (Shell)

ENSIIE, 1ère Année - durée : 1h45 (21 pts) – G. Berthelot, S. Genaud et D. Grad

14 octobre 2014

1 Partie 1

1.1 Création de fichier (1 point)

Écrivez le script `exo1.sh` dont le comportement est le suivant: il crée le fichier de nom `dummy` dans le répertoire courant, avec le contenu: `dummy dummy`.

1.2 Création paramétrée de fichier (2 points)

Écrivez le script `exo2.sh` dont le comportement est le suivant:

- si il n'est pas lancé avec 1 argument, il écrit un message d'erreur sur la sortie erreur, et sort en renvoyant la valeur 1.
- si il est lancé avec 1 argument que nous appellerons `arg` et dont la valeur sera notée `<arg>`, alors il crée (ou recrée après écrasement) le fichier de nom `<arg>` (les noms ou chemins relatifs sont interprétés depuis le répertoire courant), le contenu du fichier `arg` devra être `<arg> <arg>`.

1.3 Création paramétrée de fichier généralisée (2 points)

Écrivez le script `exo3.sh` dont le comportement est le suivant:

- si il n'est pas lancé avec 1 argument, il écrit un message d'erreur sur la sortie erreur et renvoie 1.
- si il est lancé avec 1 argument que nous appellerons `arg` et dont la valeur sera notée `<arg>` alors il lit d'abord une valeur `N` sur l'entrée standard (on considérera sans le vérifier que cette valeur est un entier positif), puis il crée (ou recrée après écrasement) le fichier `<arg>`. Le contenu du fichier `<arg>` devra avoir 1 seule ligne contenant `N` fois `<arg>` séparés chacun par un espace: "`<arg> <arg> <arg> ...`".

Rappel: `x=$((x+1))` ou `x=${x+1}` incrémentent la variable `x`.

2 Partie 2

Copiez les données suivantes dans un fichier nommé `data` en séparant les valeurs par une virgule.

Joe	10/12/1990	Strasbourg
Alice	05/03/1998	Nantes
Bob	01/08/2001	Amsterdam
Charles	24/03/1999	Berlin
Paul	11/09/2010	Paris
Alice	03/01/2013	Nice

2.1 Extraction des prénoms, villes (3 points)

Écrivez un script `exo4.sh` dont le comportement est le suivant:

- si il n'est pas lancé avec 1 argument, il écrit un message d'usage et sort avec un code de retour de 1,
- si il est lancé avec 1 argument, affiche une partie des lignes du fichier `data` dans lesquelles figurent l'argument. L'affichage ne devra afficher que la partie *prénom* et *ville*, c'est-à-dire les colonnes 1 et 3 respectivement de chaque ligne sélectionnée. Dans le cas où l'argument n'est pas présent dans le fichier, le script affichera le message "valeur absente".

Exemple:

```
% ./exo4.sh Alice
Alice, Nantes
Alice, Nice
```

2.2 Extraction de personnes selon âge (4 points)

Écrivez un script `exo5.sh` qui prend 1 argument supposé être un entier (on ne demande pas de le vérifier), qui liste tous les prénoms (valeur de la colonne 1) pour lesquels l'année de naissance (prise dans la colonne 2) est supérieure ou égale à la valeur du paramètre.

```
% ./exo5.sh 2001
Bob
Paul
Alice
```

3 Partie 3

3.1 Vérification de fichiers (1,5 points)

Écrivez le script `exo6.sh` ayant le comportement suivant:

- si il n'est pas lancé avec 2 arguments, il écrit un message d'erreur sur la sortie erreur et renvoie 1.
- si il est lancé avec 2 arguments, les arguments sont considérés comme des chemins de fichiers. Si les deux fichiers existent et sont des fichiers réguliers, il écrit sur la sortie standard "oui". Dans le cas contraire il écrit "non".

3.2 Vérification généralisée de fichiers (1,5 points)

Écrivez le script `exo7.sh`, qui est similaire à `exo6.sh`, mais qui accepte N arguments avec $\langle N \rangle$ supérieur ou égal à 1.

3.3 Listing de fichiers (3 points)

Écrivez le script `exo8.sh` qui devra être appelé avec deux arguments nommés `arg1` et `arg2`. Il écrira dans le fichier de nom $\langle \text{arg1} \rangle$ la date du jour au format JJ/MM/AAA. Il ajoutera ensuite dans ce fichier le contenu de tous les fichiers de l'arborescence de l'utilisateur dont les noms se terminent par $\langle \text{arg2} \rangle$. Chaque contenu devra commencer sur une nouvelle ligne. Ce script doit impérativement utiliser la commande `find`.

3.4 Listing récursif de fichiers (3 pts)

Écrire le script `exo9.sh` qui fait la même chose que le script `exo8.sh` mais en utilisant une fonction récursive et non la commande `find`.