UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

Licence INFO
Info0302
2018/2019

Stage Unix : TPtest (1h)

Pour cette épreuve créez un répertoire `TPtest_`*login* dans lequel vous allez composer :

- pour le début du sujet vous produirez vos réponses dans un fichier texte unique, dont le nom sera formé à partir de votre nom et du numéro de votre machine : pour l'étudiant *Besançon à l'Est* qui compose sur la machine 3-05, le fichier sera nommé `debut_besancon-a-l-est_3-05.txt`
- pour la suite [et fin] du sujet vous écrirez les scripts demandés dans des fichiers indépendants (merci de respecter les noms imposés pour les fichiers demandés).
- à la fin de l'épreuve constituez une archive compressée de ce répertoire (`.tgz`)

Exercice 1 (Une commande de rêve ?)

1°) Que fait la commande `rev` ?

2°) Comment doit-on l'utiliser pour appliquer ce traitement à chaque ligne d'un fichier texte `A.txt` et obtenir le résultat dans un autre fichier texte `B.txt` ?

Exercice 2 (La commande `calendar`)

1°) Que fait la commande `calendar` ?

2°) Quelle commande avez-vous invoquée pour obtenir des informations détaillées sur cette commande ?

3°) Comment devez-vous invoquer cette commande pour afficher les résultats allant d'hier à après-demain ?

4°) On souhaite créer une commande `x` permettant de réaliser ce traitement, depuis n'importe quel emplacement de l'arborescence, et n'importe quand (même lors de notre prochaine connexion) :

- donnez une solution qui permette de ne pas écrire de fichier shellscript (détaillez l'ensemble des manipulations à réaliser)
- si vous passez par l'écriture d'un script (dans un fichier `x`), comment faites vous pour qu'on puisse l'invoquer comme une commande [et ce depuis n'importe quel endroit de l'arborescence] ? NB : on ne vous demande pas d'écrire le contenu du fichier `x`.

Exercice 3 (Combinaison de commandes)

1°) Pour un fichier texte `A.txt`, comment peut-on obtenir le nombre de lignes de ce fichier contenant la chaîne de caractères "*bon*" (sans guillemets) ?

2°) Ecrivez un script `ex3.2.sh` prenant deux paramètres, le nom d'un fichier et une chaîne de caractères, et permettant de réaliser le traitement suivant :

- si le fichier n'existe pas ou est vide (taille nulle), afficher un message d'erreur sur *stderr*

- sinon afficher sur la sortie standard le nombre de lignes de ce fichier qui contiennent la chaîne de caractères passée en deuxième paramètre.

NB : on ne demande pas de vérifier que l'utilisateur passe effectivement deux paramètres.

3°) Ecrivez un script `ex3.3.sh` qui prend comme unique paramètre une chaîne de caractères et qui donne les noms de tous les fichiers du répertoire courant dont au moins deux lignes contiennent la chaîne de caractères en question :

- si on passe trop ou trop peu de paramètres à ce script, il doit afficher un message qui en rappelle la syntaxe d'appel
- sinon il doit utiliser le script `ex3.2.sh` , mais sans rien produire sur la sortie d'erreur standard en cas d'erreur sur une des entrées du répertoire courant (un répertoire/...)

NB : en utilisant une boucle *Pour* on peut considérer toutes les entrées du répertoire courant.

Exercice 4 (Paramètres de la ligne de commande)

Ecrivez un script `ex4.sh` qui détermine et affiche le nombre de valeurs positives passées en paramètre, et leur produit :

```
[chj@Ub: TPtest_jaille01]$ ./ex4.sh 3 vingt -2 7
 => 2 valeurs strictement positives ; leur produit est 21
```

- il peut accepter un nombre de paramètres quelconque (éventuellement 0)
- il n'affiche pas de message d'erreur pour les paramètres qui ne sont pas des valeurs entières

---

Exercice 5 (Sauvegardes)

Pour ce problème on demande plusieurs versions d'un script de sauvegarde : vous écrirez la version n° i dans le fichier `save.`i`.sh` (`save.1.sh`, ...)

v1 : on donne le nom d'un fichier à sauvegarder sur la ligne de commande (par exemple *nomfic*) : on vérifie qu'il existe ; si c'est le cas on le copie en *nomfic*`.bak`

v2 : ... mais s'il n'existe pas on le dit

v3 : s'il y a déjà une sauvegarde *nomfic*`.bak` on le dit

v4 : si *nomfic*`.bak` existe on le renomme en *nomfic*`.1.bak` et on fait la sauvegarde en *nomfic*`.bak`

v5 : *idem* mais si nécessaire on va plus loin dans les renumérotations : *nomfic*`.2.bak` ... *nomfic*`.241.bak`

— *plus difficile*

v6 : supprimer toutes les sauvegardes des fichiers existants, mais garder celles dont le fichier d'origine n'existe plus (on ne conserve que la plus récente, *nomfic*`.bak`)

— *bonus !*

v7 : sauvegarde avec la date courante : *nomfic*`_20180921-171243.bak`

v8 : supprimer toutes les sauvegardes antérieures à une certaine date

...