TP 4 - Un peu de lecture

Pour cette séance, nous utiliserons les livres du site Project Gutenberg: `https://www.gutenberg.org/`, et le site RegExr pour nous aider: `https://regexr.com/`.

Exercice 1. Écrivez une fonction `verif_param_script()` prenant un seul paramètre correspondant à un fichier, et sortant du programme en cas d'erreur, c'est-à-dire:

1. si le chemin est vide, ou
2. si le fichier n'existe pas, ou
3. si ce n'est pas un fichier "normal", ou
4. s'il n'est pas lisible.

Utilisez `exit` pour quitter le script, avec un code et un message d'erreur différent dans chaque cas.

Exercice 2. Nous allons écrire pas à pas un script `richesse.sh` qui affiche le nombre de mots réels différents qu'un texte en français donné contient.

1. Installez un dictionnaire français, par exemple: `/usr/share/dict/french` (utilisez votre script `installer_multi.sh` de la séance précédente!).
2. Écrivez une expression régulière décrivant un mot en français. Rappelez-vous que les mots peuvent commencer par une majuscule, et que les caractères peuvent être accentués. Pour simplifier, on ignorera les élisions (le caractère `'`) et les traits d'union (le caractère `-`).
3. Utilisez votre expression avec `grep` pour extraire tous les mots d'un texte et les enregistrer **en minuscules** dans un fichier, triés et sans doublons. La conversion des majuscules accentuées pose problème aux programmes "classiques" comme `tr`; il faudra utiliser `gawk` comme suit pour la conversion:

```
$ grep ... | gawk '{print tolower($0);}'
```

4. Utilisez `comm` pour afficher le nombre de "mots réels" uniques que votre texte contient, c'est-à-dire le nombre de mots qui appartiennent au dictionnaire.
5. Combinez vos réponses aux sous-questions précédentes en un script `richesse.sh`, qui prend en unique paramètre le chemin vers le fichier texte à évaluer. N'oubliez pas de vérifier la présence du paramètre et l'existence du fichier.

Exercice 3 (Renommage)**.** Les fichiers disponibles sur le site du projet Gutenberg sont nommés à l'aide de nombres (par exemple: `14155.txt.utf-8`), ce qui complique leur identification. Écrivez un script `renommer_livres.sh` qui renomme tous les livres dans un répertoire au format:

`Nom complet de l'auteur - Titre.EXTENSION`

où `EXTENSION` est l'extension d'origine (`txt` ou `txt.utf-8`). Le script prendra en paramètre le chemin vers le répertoire contenant les fichiers. Pour vérifier vos réponses, utilisez les données à récupérer ici:

```
wget http://igm.univ-mlv.fr/~alabarre/teaching/shnu/linux/dataset_livres_tp04.zip
```

Les renommages corrects sont:

```
11049.txt.utf-8   →   Honore de Balzac - Eugenie Grandet.txt.utf-8
13951.txt.utf-8   →   Alexandre Dumas - Les Trois Mousquetaires.txt.utf-8
14155.txt.utf-8   →   Gustave Flaubert - Madame Bovary.txt.utf-8
19657.txt.utf-8   →   Victor Hugo - Notre-Dame de Paris.txt.utf-8
42131-0.txt       →   Francois-Marie Arouet - Traité sur la tolérance.txt
```

Lisez les fichiers récupérés pour deviner comment construire les expressions régulières nécessaires. **Faites bien attention aux deux points suivants:**

1. les titres peuvent contenir des caractères non autorisés ou gênants pour les noms de fichiers (par exemple: `/`). Pensez à les remplacer (par exemple par `-`).
2. les champs récupérés se terminent par le caractère `\r` de retour à la ligne: supprimez-les de votre variable `var` à l'aide de la commande : `var=${var//[$'\r']}`.

Exercice 4 (Liste d'auteurs)**.** Écrivez une commande basée sur `curl` qui récupère et affiche tous les auteurs de la page `https://www.gutenberg.org/browse/languages/fr`. Pour vérifier votre réponse, redirigez le résultat de votre commande vers un fichier et comparez-le avec `diff` au résultat obtenu ici:

```
wget http://igm.univ-mlv.fr/~alabarre/teaching/shnu/linux/auteurs_fr
```

Lisez le code HTML de la page pour vous aider à construire votre expression régulière.

Exercice 5 (Répartition)**.** Écrivez un script `repartition.sh` qui réorganise les fichiers texte d'un répertoire passé en paramètre de la façon suivante: on crée un sous-répertoire par auteur, et l'on y déplace tous les livres de cet auteur. On suppose que le répertoire sur lequel on agit contient des livres nommés au format obtenu dans l'exercice 3, et l'on ignore les fichiers qui ne se terminent pas par `.txt` ou `.txt-utf8`. Vérifiez votre réponse sur votre répertoire de l'exercice 3. Attention:

- rappelez-vous que les prénoms peuvent être composés, donc contenir des `-`! Il faut donc extraire la chaîne avant la première sous-chaîne `" - "`, et pas juste `"-"`!
- pour demander à votre expression régulière de s'arrêter avant la première occurrence de `x`, utilisez l'opérateur `?`. Par exemple, pour la chaîne `"axbxc"`:
  - `([a-z]*)x[a-z]*` extrait `"axb"`;
  - `([a-z]*?)x[a-z]*` extrait `"a"`.