

Documents numériques
Majeure Informatique
Exercices 2018–19

Michel Beigbeder

1 Exercices de programmation en sh, 11 septembre 2018

Exercise1 TP : arguments et shell

- 1.1. Écrivez un programme `ex1.c` qui lancé depuis un shell affiche sur sa sortie standard ses arguments avec leur index, à raison d'un argument par ligne.
- 1.2. Testez-le avec un nombre croissant d'arguments (par exemple 1, 10, 100, 1 000, 10 000, 100 000, 1 000 000 ...) jusqu'à obtenir un message d'erreur.
- 1.3. Trouvez (et lisez) une/des page/s sur le Web qui décrit/vent cette limite et permet/tent de répondre aux questions suivantes :
 1. Quelle est la nature de cette limite ?
 2. Qui/quoi est à l'origine de cette limite dans votre système ?
 3. Comment connaître la valeur de cette limite dans son environnement ?
 4. Comment contourner cette limite dans une programmation en shell ?

Exercise2 TP : Données, fichier, format

□ DONNÉES : une liste de nombres.

Dans un premier format le fichier de données :

- ne contient que des chiffres décimaux et des passages à la ligne ;
- et chaque nombre est écrit (formaté) en base 10 à raison de l'écriture d'un nombre par ligne.

Dans un deuxième format, ce fichier :

- ne contient que des chiffres décimaux, des passages à la ligne, des espaces et des tabulations ;
- chaque nombre est écrit (formaté) en base 10 ;
- les nombres sont séparés par un ou plusieurs caractères qui ne sont pas des chiffres.

-
- 2.1. Chargez le fichier `DISTR.tar` depuis `campus.emse.fr`. Extrayez les fichiers de cet archive puis ceux de l'archive `PetitsFichiersPresqueNumeriques.tgz`. Parmi les fichiers `e01 e02 e03 e04 e05 e06 e07 e08 e09 e10` du répertoire `PetitsFichiersPresqueNumeriques` lesquels respectent l'un ou l'autre des formats précités ?
 - 2.2. Précisez un troisième format de fichier dans le cas où ce fichier peut contenir n'importe quels caractères. (Quels seront les nombres considérés ?)

□ TRAITEMENT : calculer la somme des nombres représentés dans un fichier de données dans l'un des formats précédents. ■

- 2.3. Écrivez un programme qui traite l'un des formats de fichier et qui répond au besoin.
 Sans argument, ce programme lit les données sur son entrée standard.
 Avec une liste d'arguments non vide, il lit les données dans les fichiers dont les noms sont les arguments. Vous devrez utiliser *argc* et *argv*.
 Soyez attentifs à l'usage de *stdout* et *stderr*.
 En cas d'erreur votre programme renverra un code d'erreur à son environnement d'exécution. Les erreurs possibles sont :
 — fichier inexistant ou non lisible,
 — fichier de contenu ne respectant pas le format prescrit.
- 2.4. Est-ce que ces formats de fichiers pourraient être étendus pour accepter des nombres écrits dans différentes bases. Faites une proposition.

Exercice3 TP : Statistiques sur des fichiers

Téléchargez le fichier <http://ftp.gnu.org/gnu/coreutils/coreutils-8.25.tar.xz>. Puis désarchivez le.

```
xzcat coreutils-8.25.tar.xz | tar xf -
```

- 3.1. Donnez une définition des opérations de compression et d'archivage.
 3.2. Le rôle de l'utilitaire **tar** est-il l'archivage ou la compression. (Illustrez votre réponse à l'aide d'extrait(s) du manuel.)

□ Pour chacune des questions suivantes, écrivez un script `ex3.3.sh...ex3.7.sh`. ■

- 3.3. Combien de fichiers du répertoire `coreutils-8.25/src` n'ont pas de point dans leur nom ?
 Indications : `ls`, `grep`, `wc`
- 3.4. Combien de suffixes différents y'a-t-il dans les noms de ces fichiers ?
 (Cette question impose que vous donniez explicitement votre définition de « suffixe ».)
 Indications : `ls`, `sed`, `sort`
- 3.5. Pour chacun des suffixes, listez les noms des fichiers qui ont ce suffixe, à condition que ces fichiers contiennent du texte imprimable. Affichez aussi leur nombre de lignes.
 Indications : `ls`, `file`, `grep`, `wc`
- 3.6. Complétez le script précédent pour que pour chaque suffixe il termine par le cumul des lignes des fichiers qui ont ce suffixe en rappelant le répertoire et le suffixe considérés.
 Indications : `expr`, `echo`
- 3.7. Faites une fonction de votre programme précédent et appliquez-là sur tous les sous-répertoires de `coreutils-8.25`.
 Indications : `find`