

Initiation USI, TP12

1 Utilisation de “case”

Utilisez l'exemple 3.9 de la feuille “Unix : shell scripts” qui a été distribuée comme base pour écrire une commande `voir.sh` qui prendra comme paramètre un nom de fichier.

Cette commande affichera normalement le contenu du fichier (`cat`). Si c'est une archive (nom terminé par `.tar`, `.tgz` ou `tar.bz2`, ou `zip`, ZIP, elle affichera le “catalogue” de l'archive (voir options `-t` de `tar`, et `-l` de `unzip`. Si c'est une image (`jpg`, `jpeg`,... elle fera appel à `file` pour afficher ses caractéristiques. Et à `pdftinfo` pour les PDF.

Exemples d'utilisation

```
$ voir.sh archive.zip
Archive: archive.zip
  Length      Date    Time    Name
-----
         0  11-29-10  10:13   vieux/
        194  11-29-10  09:59   vieux/a.html
        868  11-29-10  09:59   vieux/completer.c
       2097  11-29-10  09:59   vieux/gp
        798  11-29-10  09:59   vieux/dead.letter
-----
       3957
                    5 files
```

2 Utilisation de “if”

2.1 Explications

1. Les débutants en *shell* sont souvent surpris par la commande “if”, qui prend comme condition une *commande*, et non une expression logique comme on s'y attendrait. C'est le succès, ou l'échec, de la commande qui conditionne l'exécution des parties “then” et “else”. Exemple

```
if g++ essai.cc -o essai
then
    echo "compilation ok. Exécution !"
    essai
fi
```
2. la commande à tout faire `test` permet d'exprimer les conditions les plus fréquentes, par exemple comparer deux chaînes, tester si un fichier existe, etc. Exemple

```
if test -d ~/.local/share/.Trash
then
    echo -n "Vider la Poubelle o/n ? "
    read reponse
    if test "$reponse" = "o"
    then
        rm -rf ~/.local/share/.Trash/*
    fi
fi
```

3. la commande `[` est un synonyme de `test`, qui allège l'écriture. Penser à mettre des espaces, et mettre un crochet fermant.

```
if [ ! -f $1 ]
then
    le fichier $1 n'existe pas.
fi
```

2.2 Application

Pour apprendre C++, vous faites sans doute des essais, qui se réduisent à un seul fichier source. Ecrivez un script qui imitera le comportement de `make` pour (re)compiler un tel source.

Usage : `fabriquer.sh préfixe`

Exemple : `fabriquer mon-prog`

Fonctionnement

- erreur si le source `mon-prog.cc` n'existe pas ou n'est pas lisible
- compile `mon-prog.cc` si l'exécutable `mon-prog` n'existe pas déjà.
- erreur si `mon-prog` existe mais n'est pas un exécutable (tester si le résultat de `file` contient `LSB executable`);
- recompile si l'exécutable existe, mais est plus ancien que le source.

3 Boucle “while”

3.1 Explications

Comme “if”, la boucle `while` utilise une commande comme condition.

- inspirez-vous de l'exemple 3.16 de la feuille pour faire un script `carres.sh` qui affiche les nombres de 1 à 20 et leurs carrés.

- utilisez `printf` (voir 3.17) pour la présentation : nombres cadrés à droite, carrés à gauche


```
1 1
2 4
3 9
4 16
....
10 100
```
- Écrire un script `devinette.sh` qui
 - tire un nombre au hasard entre 100 et 999. Indication : étudiez la commande `echo $((RANDOM % 10))`
 - demande à l'utilisateur de deviner
 - affiche "trop petit", "trop grand" et redemande,
 - ou s'arrête si c'est trouvé.

voir aussi `break`, en 3.18.

3.2 "while ... read"

- Utilisez l'exemple 3.17 pour améliorer l'affichage du carnet téléphonique, dans le script développé au TD précédent.
- Partant d'un fichier `vacances-2010.txt` qui contient des noms de photos et des légendes

```
dscn1234.jpg:prêts pour le départ
dscn1245.jpg:enfin arrivés
dscn1253.jpg:premier jour, barbecue
```

écrire une commande qui générera un "album", répertoire de pages HTML contenant chacune une photo, sa légende et des liens vers les pages précédente et suivante. Usage :
`album.sh vac2010 "Vacances en Italie"`
 Le fichier `page-3.html` pourra ressembler à ceci

```
<html>
<head>Vacances en Italie</head>
<body>
<h1>premier jour, barbecue</h1>

<p>
<a href="page-2.html">Précédent</a>
```

```
<a href="page-4.html">Suivant</a>
</body>
</html>
Conseils
```

- pour vous simplifier la vie, prévoyez une page 0 pour la couverture, et une page de fin.
- plutôt que de multiplier les "echo", utilisez cette forme de redirection ("here-document") de l'entrée standard, qui autorise l'emploi de variables :


```
echo >$fichier <<FIN
<html>
<head>$titre_album</head>
<body>
<h1>$legende</h1>
....
</html>
FIN
```

4 Boucle "for"

4.1 Explications

Le boucle `for` de *bash* exécute une suite de commandes avec une variable de boucle qui parcourt une liste de valeurs.

4.2 Application

Écrivez un script qui affiche une table qui montre tous les fichiers `.cc` d'un répertoire (liste obtenue par `find $repertoire -name '*.cc'`), avec le nombre de lignes (`wc -l`) et la date de dernière modification (`ls -l, cut`).

Idée de présentation

lignes	date	fichier
23	2 nov 22 08:56	Essais/premier.cc
432	2 nov 27 04:32	Projet/final.cc
...		