TP LO22 / Compléments : Utilisation de diff Eric Bachard septembre 2004

1) But de ce programme

diff est un utilitaire GNU, initialement écrit par Larry Wall (inventeur de Perl), et modifié ensuite par Paul Eggert, de manière à pouvoir comparer des fichiers binaires entre eux (option -a) .
En général, diff sert à comparer deux fichiers, ou deux répertoires (y compris récursivement). Le résultat est par défaut affiché sur la sortie standard.
Dans la pratique, on redirige plutôt le résultat dans un fichier que l'on appelle « patch », et contenant juste les différences entre fichier1 et fichier2. Sinon, il s'agit d'un fichier de log.

Un outil complémentaire, permettant d'appliquer les changements constatés, est patch (voir dans les complements de TP LO22 relatifs à patch)

2) Utilisation

Syntaxe : diff [options] fichier1 fichier2

N.B. : une option de diff commence par un tiret (option courte) ou deux tirets (option longue). en général, on peut ainsi écrire une option de deux manières.

Sans redirection, comment apparaît le résultat, sur la sortie standard (l'écran) par défaut ?

Cas 1) soit les deux fichiers sont identiques, et dans ce cas, l'invite du shell est simplement retournée

Cas 2) soit les différences sont affichées sur la sortie standard, de la manière suivante :

- Toute ligne commençant par < donne ce qui est présent dans fichier1, mais pas dans fichier2

- Toute ligne commançant par > donne ce qui est présent dans fichier2, mais pas dans fichier1

Options les plus importantes : -Naur équivalente à -N -a -u -r

-N ou --new-file : dans la comparaison de répertoires, considère que les fichiers n'existant pas dans un des répertoires sont présents, mais vides.

-a ou --text : Tous les fichiers sont considérés comme des fichiers textes. Cette option permet de comparer deux fichiers binaires.

-u ou --unified : réalise un "diff unifié", qui permet d'afficher dans un seul bloc, et dans leurs contextes, les deux versions de chaque ligne.

-r ou --recursive : permet la comparaison récursive de deux répertoires : cette option est très intéressante quand on veut faire un patch unique alors que plusieurs fichiers ont été modifiés dans une arborescence.

3) Exemples

3.1)

fichier1 contient les lignes suivantes :

```
----------------------------début ----------------------
    1   #!/bin/bash
    2
    3   echo -e "Entrez votre nom \n"
    4   read nom
    5   #ligne différente
    6   echo -e "Entrez maintenant votre prénom \n"
    7   read prenom
    8
    9   clr && sleep 1
   10
   11
   12   echo -e "Bonjour $prenom $nom !"
   13
   14
   15
   16   exit0
-----------------------------fin--------------------------
```

fichier2 contient les lignes suivantes :

```
-----------------------------début----------------------
    1   #!/bin/bash
    2
    3   echo -e "Entrez votre nom \n"
    4   read nom
    5   echo -e "Entrez maintenant votre prénom \n"
    6   read prenom
    7
    8   clr && sleep 1
    9
   10
   11   echo -e "Bonjour $prenom $nom !"
   12
   13
   14
   15   exit0
----------------------------fin-----------------------------
```

Page suivante, figurent les résultats, respectivement de diff fichier1 fichier2, puis de diff -Naur fichier1 fichier2

```
diff fichier1 fichier2
```

```
eric@alube:~$ diff fichier1 fichier2
5d4
< #ligne différente
```

On constate bien que le ligne marquée gauche (<) est dans fichier1, mais pas dans fichier2. le "d" signifie qu'il faut effacer (d comme delete) la ligne 5, qui devient a ligne 4 de fichier2

Remarque : si on avait entré la commande diff fichier1 fichier2, la sortie aurait été :

```
eric@alube:~$ diff fichier2 fichier1
4a5
> #ligne différente
```

Ici, a (qui signifie "append", ajouter) signifie qu'il faut ajouter le texte à droite de > et la ligne ligne 4 devient la ligne 5

```
diff -Naur fichier1 fichier2
```

```
eric@alube:~$ diff -Naur fichier1 fichier2
--- fichier1      2004-09-09 18:30:42.784870456 +0200
+++ fichier2    2004-09-09 18:30:06.362407512 +0200
@@ -2,7 +2,6 @@

 echo -e "Entrez votre nom \n"
 read nom
-#ligne différente
 echo -e "Entrez maintenant votre prénom \n"
 read prenom
```

On voit que dans la seconde version, en plus des lignes qui changent, des lignes supplémentaires, donnant le contexte, sont ajoutées. Les dates aussi sont indiquées...

3.2)

Plus compliqué : voir les exemples étudiés en TP ;-)