TD n°3 – noms complets et paramètres

1) Arbre mystère

Voici différents noms complets de fichiers dans un système fictif dans lequel les éléments portent des noms tous différents (il n'y a qu'un seul d1, un seul f2…). Les répertoires sont nommés d… et les fichiers f… On suppose que l'utilisateur qui voit ces noms se trouve dans un répertoire appelé d1 quelque part dans cet arbre. Initialement, on n'en sait pas plus que ça.

Dessiner l'arbre des fichiers correspondant aux noms qui suivent, les répertoires seront représentés par des rectangles et les fichiers par des cercles :

```
../../../d7/f1          ../../d8/f2          /f3
f4                      /d2/f5               d6/f6
../../../d7/d4/f7       ./f8                 ../d9/f9
/d5/d3/d9/f10
```

Voici maintenant quelques commandes qui s'appliquent à cet arbre. Vous indiquerez quel est ou quels sont les répertoires dans lesquels elles peuvent marcher. Par exemple, `more /f3` peut marcher dans n'importe quel répertoire, mais `more f5` ne peut marcher que dans d2. Si elles sont possibles quelque part, alors quels sont leurs effets ?

```
1)  more f1                 6)  mv d6/f6 .
2)  more d8/f2              7)  cp ../f3 d8/f3
3)  more ../d9/f9           8)  cp ../d2/f5 d3
4)  mv ./f6 ../..           9)  mkdir d8
5)  rm ../f1 f7             10)  mkdir d6/d11
```

2) Emploi des noms complets dans les commandes

Soit votre dossier algo et dedans trois sous-dossiers bin, tp1 et tp2. Dans tp1 et tp2, il y a un programme source en langage C qui porte le nom du sous-dossier : dans tp1, il y a tp1.c et dans tp2, il y a tp2.c.

On veut les compiler ces deux sources et obtenir leur programme exécutable dans le dossier bin de algo. Faire un schéma de la situation et de ce qui est voulu.

On se situe et on reste dans le dossier algo (pas le droit de faire cd).

1. Quelle est la commande de compilation qui crée le programme exécutable tp1 dans bin à partir de tp1.c situé dans tp1 ?
2. Comment lancer l'exécution de tp1 situé dans bin, toujours sans bouger de algo ?
3. Maintenant, quelle est la commande de compilation de tp1.c si on veut que le résultat exécutable se trouve dans algo (c'est à dire dans le dossier courant) ? Est-ce que c'est possible ?
4. Même question pour obtenir le résultat de compilation de tp1.c dans tp1.
5. Initialement, on avait déjà compilé tp1.c dans tp1 et tp2.c dans tp2. On voudrait les déplacer tous les deux en une seule opération vers le dossier bin. Quelle est cette commande ?
6. En fait, on avait oublié l'option -o lors de la compilation de tp2.c, et le programme binaire correspondant s'appelle a.out et est situé dans tp2. Quelle est la commande (unique) pour déplacer ce a.out dans bin, tout en le renommant en tp2 ?

3) Redirections et tubes

Le cours a montré les directives pour changer respectivement la provenance (stdin) et la destination (stdout) des textes resp. lus et écrits par une commande, ainsi que la directive pour connecter par un tube deux commandes entre elles, la sortie de la première sur l'entrée de la seconde.

```
cmde < fic_entrée
cmde > fic_sortie
cmde < fic_entrée > fic_sortie
cmde1 | cmde2
```

On peut mixer les tubes avec les redirections mais il est évident qu'une sortie ne peut pas être connectée simultanément à deux entrées et réciproquement, alors indiquez les erreurs syntaxiques parmi les commandes suivantes (ce que font les commandes n'intervient pas pour cet exercice : sort classe les lignes et head n'affiche que les n premières mais peu importe) :

- `more fichier1 < fichier2 | sort | head -n 1 > fichier3`
- `more fichier1 | sort > fichier2 | head -n 1 > fichier3`
- `more < fichier1 | sort < fichier2 | head -n 1 > fichier3`
- `fichier1 > sort | head -n 1 > fichier2`
- `more fichier1 | sort | head -n 1 < fichier2`
- `more fichier1 | fichier2 | sort | head -n 1 > fichier3`
- `more < fichier1 | sort | head -n 1 | fichier2`
- `more | sort fichier1 | head -n 1`
- `fichier1 | sort | head -n 1`
- `more | sort | head -n 1 < fichier1 > fichier2`
- `more < fichier1 | sort | head -n 1 > fichier2`

À déplorer que bash ne signale aucune de ces erreurs, simplement, la ligne de commande ne fonctionne pas : elle reste bloquée ou elle ne fait pas du tout ce qu'on espère.