

## TE 2

# Compléments sur la hiérarchie de fichiers – Redirections

### Références

- Polycopié UNIX : chapitres 2, 3, 4, 5,10
- Transparents UNIX : sections 3, 4, 5, 6, 7, 8, 9, 10, 11

Créer dans votre répertoire `unix`, un sous-répertoire `te2` pour y placer tous les fichiers du TE 2.

### Ex. 1 : Droits d'accès **AB**

#### Droits sur les fichiers ordinaires **A** 10 min.

1. Afficher le fichier `~lefrere/M1/Config/etudiants.bash_profile`, puis afficher les attributs de ce fichier (droits, propriétaire, taille, date, ...). Repérer quels droits d'accès font que vous pouvez le lire, mais que vous ne pouvez pas écrire dans ce fichier.
2. Afficher les droits du fichier de la commande `ls` (fichier `/bin/ls`).
3. Choisir comme répertoire de travail le répertoire `C` créé (sous `programmes`) dans l'exercice 10 du TE 1. Compiler le fichier `carre+invite.c` avec la commande  
`gcc-mni-c89 carre+invite.c`  
Cela produit un fichier `a.out` que vous pouvez exécuter avec la commande  
`./a.out`  
Repérer les droits de ce fichier. Les modifier afin que vous seul puissiez exécuter ce fichier.
4. Protéger le fichier `carre+invite.c` afin que personne (ni même vous) ne puisse le lire. Vérifier en essayant de l'afficher.

#### Droits sur les répertoires **B** 20 min.

1. Quels droits du répertoire `programmes` permettent de faire la liste des fichiers qu'il contient ? Les modifier temporairement et vérifier l'effet produit, puis redonner ces droits.
2. Quels droits du répertoire `programmes` permettent de le traverser pour atteindre un fichier plus bas dans la hiérarchie ? Les modifier temporairement et vérifier l'effet produit, puis redonner ces droits.
3. Quels droits du répertoire `programmes` permettent de copier ou détruire des fichiers dans ce répertoire ? Les modifier temporairement et vérifier l'effet produit en essayant d'y copier le fichier `ligne.txt`, puis redonner ces droits.
4. Exécuter les commandes suivantes et expliquer.  
`chmod a-r,a+x programmes`  
`ls programmes`  
`cd programmes/C`  
`ls`
5. Créer dans votre répertoire d'accueil un (sous-)répertoire de nom `bin`. Copier le fichier `a.out` dans `bin`. Protéger `bin` afin que personne d'autre que vous ne puisse y accéder.

### Ex. 2 : Recherche de fichiers **AB** 25 min.

#### **AB**

1. En utilisant la commande `find`, chercher sous `~lefrere/M1/` tous les fichiers dont le nom est suffixé par `.c` (fichiers sources de programmes en C).

- Utiliser `find` pour établir la liste de tous les répertoires situés sous votre répertoire d'accueil. Que remarquez-vous ?

**B**

- Afficher la liste de tous les fichiers de taille inférieure à 1000 octets dans votre répertoire d'accueil et ses sous-répertoires. Vérifier que le critère de sélection est bien respecté. Comment procéder pour afficher leurs chemins absolus ?

### Ex. 3 : Archivage et compression **AB** 25 min.

**A**

- L'objectif est de créer dans votre répertoire `unix` un fichier `doc.tar` qui archive la branche `Doc` de la hiérarchie située sous le répertoire `~lefrere/M1/`. On choisit donc le répertoire `~lefrere/M1/` comme répertoire de travail. Quelle est la taille du fichier d'archive en octets ?
- Vérifier la structure de l'arborescence ainsi stockée dans `doc.tar`.
- Reconstruire à partir de l'archive cette arborescence de fichiers sous votre répertoire `te2`.
- Parcourir l'arborescence ainsi restituée et y localiser notamment le fichier `poly-unix.pdf`.

**B**

- Déplacer l'arborescence installée (`Doc`) dans votre répertoire `unix`.
- Compresser le fichier d'archive. Quel est le taux de compression ?
- Vous envoyer le fichier d'archive compressé en fichier attaché par mail.
- Recommencer l'opération 1 en compressant l'archive lors de sa création (suggestion : `man tar`).
- Reprendre l'opération de compression avec `bzip2` (éventuellement avec `unlzma` ou `xz`). Comparer le taux de compression avec celui de `gzip`.

### Ex. 4 : Téléchargement d'une arborescence et extraction **B** 20 min.

L'objectif est d'installer sur votre compte une hiérarchie de travaux pratiques du cours d'unix de l'IDRIS, disponible sous la forme de fichier d'archive compressée à l'URL :

"`http://www.idris.fr/data/cours/unix/user/TP.tar.gz`"

- Copier le fichier `TP.tar.gz` dans votre répertoire `~/tmp/` avec la commande `wget`
- Quelle est la taille du fichier téléchargé ? Décompresser le fichier. Quel était le taux de compression ?
- Vérifier le contenu de l'archive avant d'en extraire l'arborescence.
- Extraire les fichiers de l'archive de façon à ce que le répertoire `TP` soit un sous répertoire de votre répertoire `te2`. Dessiner l'arborescence restituée.
- Détruire tous les fichiers de cette hiérarchie dont le nom commence par `mpp`
- Archiver la nouvelle hiérarchie dans un fichier `TP-IDRIS.tar` placé dans votre répertoire d'accueil.
- Détruire l'arborescence modifiée `TP`
- Restituer l'arborescence modifiée à partir de l'archive `TP-IDRIS.tar`

### Ex. 5 : Sortie standard, sortie d'erreur standard et redirections **A** 15 min.

- Depuis le répertoire `~lefrere/M1/2015-2016/etu/mni/unix/te/te2/`, copier les fichiers `debut` `fin` et `redir.sh` dans votre répertoire `te2`
- Exécuter les commandes suivantes et expliquer le résultat :

```
cat debut
cat debut fin
cat debut fin > sortie puis less sortie
cat debut /inexistant
cat debut /inexistant > sortie puis less sortie
cat debut /inexistant 2> error puis less error
cat debut /inexistant > sortie 2> error puis less sortie et less error
```

- Afficher le fichier texte `redir.sh` : il comporte des commandes, c'est un shell-script. Essayer d'exécuter ce script avec la commande `./redir.sh` Observer l'affichage : que constatez-vous ? Ajuster les droits du fichier `redir.sh` et recommencer.
- Rediriger la sortie de ce script vers un fichier `memo.txt` Vérifier le résultat en affichant `memo.txt`
- Sans passer par un éditeur de texte, ajouter le message "Fin du memo" à la fin de `memo.txt`.

### Ex. 6 : Utilisation du filtre `tr` **AB** 15 min.

Recopier dans votre répertoire de travail le fichier

`~lefrere/M1/2015-2016/etu/mni/unix/te/te2/ligne-ascii.txt`

Puis tester le comportement des commandes suivantes ; noter les résultats et les expliquer en s'aidant du manuel en ligne de `tr` :

```

1 cat ligne-ascii.txt
2 tr aeiou AEIOU < ligne-ascii.txt
3 tr "aeiou" 'AEIOU' < ligne-ascii.txt
4 tr aeiou AEIOU < ligne-ascii.txt > LIGNE-ascii.txt
5 cat LIGNE-ascii.txt
6 cat ligne-ascii.txt | tr aeiou AEIOU
7 tr '0123456789' 'abcdefghij' < ligne-ascii.txt
8 tr '0-9' 'a-j' < ligne-ascii.txt
9 tr "[:lower:]" "[:upper:]" < ligne-ascii.txt
10 tr "a-z" "A-Z" < ligne-ascii.txt
11 tr '0-9' '+' < ligne-ascii.txt
12
13 tr -d 'a-d' < ligne-ascii.txt
14 echo '1a2aa3aaa4bbbb' | tr 'a' 'A'
15 echo '1a2aa3aaa4bbbb' | tr -s 'a' 'A'
16 echo '1a2aa3aaa4bbbb' | tr -s 'a' '\n'
17 tr '0-4' '+=' < ligne-ascii.txt
18 tr -s '0-4' '+=' < ligne-ascii.txt

```

### Ex. 7 : Filtres élémentaires `head`, `tail`, `wc`, `tr` et `tubes` **A** 15 min.

- Rechercher le fichier `comptes-lutes.txt` situé quelque part sous `Doc` sur le compte de l'UE. Le recopier dans votre répertoire `te2`
- Afficher le nombre de lignes de ce fichier.
- Afficher le nombre d'octets de ce fichier. Afficher le nombre de caractères de ce fichier. Comporte-il des caractères accentués ?
- Afficher sa première ligne, puis l'afficher en lettres capitales. Stocker le résultat dans un fichier appelé `Titre`.
- Afficher les lignes 23 à 26 de ce fichier. Afficher le nombre de mots qu'elles comportent.

### Ex. 8 : Programmes et redirections **A** 20 min.

Les programmes `carre.f90` en fortran et `carre.c` en C (fournis au TE2) lisent trois nombres fournis au clavier<sup>1</sup>, calculent leur carré, puis affichent ces carrés.

- Compiler le programme fortran (`gfortran-mni carre.f90`) puis tester le fonctionnement de l'exécutable `a.out` ainsi produit. Renommer le fichier exécutable en `carre-f.exe`.
- Compiler le programme C (`gcc-mni-c89 carre.c`) et tester le fonctionnement du nouvel exécutable `a.out`. Renommer le fichier exécutable en `carre-c.exe`.

<sup>1</sup> Attention : contrairement à `carre+invite.{c,f90}`, ces programmes n'affichent aucun message invitant à saisir les données d'entrée.

3. Les filtres du système unix ne sont pas les seuls à pouvoir utiliser les redirections : les programmes qui lisent au clavier et écrivent à l'écran acceptent aussi les redirections.  
Saisir trois nombres dans un fichier que l'on nommera **in** et faire afficher leurs carrés.
4. Comment stocker les résultats de plusieurs exécutions successives du programme dans un fichier **out** ? Où est envoyé le message d'erreur dans le cas d'une saisie non numérique avec l'exécutable fortran ? Comment le stocker dans un fichier nommé **erreur** ?
5. **AB** Comment calculer les puissances quatrièmes des entrées sans modifier le programme source ?
6. **B** Comment récupérer aussi les carrés dans un fichier ?

### Ex. 9 : Utilisation de l'éditeur vi **AB**

1. À partir du répertoire : `~lefrere/M1/2015-2016/etu/mni/unix/te2/`, recopier dans le répertoire de travail que vous vous êtes créé pour cette séance les fichiers :
  - `ref-vi-vim.txt` qui est le fichier à éditer
  - `instructions-vi-vim.txt` qui contient l'exercice à effectuer et qu'il est conseillé d'imprimer.
2. Ouvrir le fichier `ref-vi-vim.txt` sous `vi` et appliquer les actions décrites dans le fichier texte `instructions-vi-vim.txt` (ce qui suit le caractère #, et le caractère # lui-même, ne doivent pas être entrés au clavier). Commenter l'effet des requêtes proposées.