

## Thème 3

# Redirections, tubes et filtres

### Références

- Polycopié UNIX : chapitres 5, 6, 7, 8, 9, 10
- Transparents UNIX : sections 10, 11, 12, 13, 14, 15

Créer un répertoire `te3` sous votre répertoire `unix` : on y placera tous les fichiers de travail du TE3. En particulier, on y copiera tous les fichiers utiles fournis dans le répertoire `te3` du compte de l'UE.

### Ex. 1 : Sortie standard, sortie d'erreur standard et redirections **A** 15 min.

1. Depuis le répertoire `/home/lefrere/M1/2019-2020/etu/mni/unix/te/te3/`, copier les fichiers `debut`, `fin` et `redir.sh` dans votre répertoire `te3`
2. 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
```
3. 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.
4. Rediriger la sortie de ce script vers un fichier `memo.txt` Vérifier le résultat en affichant `memo.txt`
5. Sans passer par un éditeur de texte, ajouter le message `Fin` du `memo` à la fin de `memo.txt`.

### Ex. 2 : Utilisation du filtre `tr` **AB** 15 min.

Recopier dans votre répertoire de travail le fichier `/home/lefrere/M1/2019-2020/etu/mni/unix/te/te3/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. 3 : Filtres élémentaires `head`, `tail`, `wc`, `tr` et tubes **A** 15 min.

1. Rechercher le fichier `comptes-lutes.txt` situé quelque part sous `Doc` sur le compte de l'UE. Le recopier dans votre répertoire `te3`
2. Afficher le nombre de lignes de ce fichier. Comment afficher ce nombre sans le nom du fichier ?
3. Afficher le nombre d'octets de ce fichier. Afficher le nombre de caractères de ce fichier. Comporte-t-il des caractères accentués ?
4. Afficher sa première ligne, puis l'afficher en lettres capitales. Stocker le résultat dans un fichier appelé `Titre`.
5. Afficher les lignes 23 à 26 de ce fichier. Afficher le nombre de mots qu'elles comportent.

### Ex. 4 : 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.

1. 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`.
2. 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`.
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. 5 : Utilisation de l'éditeur `vi` **AB**

1. À partir du répertoire : `/home/lefrere/M1/2019-2020/etu/mni/unix/te/te3/`, 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.

---

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

**Ex. 6 : Tubes et filtres (more, less, wc, head, tail, sort, grep) A 25 min.**

Pour chaque question de 1 à 7, utiliser la commande `\ls2` avec des options éventuelles, puis un filtre à l'aide d'un tube.

1. Afficher la liste des fichiers du répertoire `/usr/bin/` en contrôlant le défilement écran par écran (tester avec `more` puis avec `less`).
2. Compter le nombre de fichiers de ce répertoire (`wc`).
3. Afficher les noms des 8 derniers fichiers de `/usr/bin/` (`tail`)
4. Afficher les noms des 8 premiers fichiers de `/usr/bin/` (`head`)
5. Afficher les noms des fichiers de `/usr/bin/` classés par ordre alphabétique inverse (`sort`).
6. Compter le nombre des fichiers du répertoire `/home/lefrere/M1/Config/`
7. Afficher la liste des fichiers du répertoire `/home/lefrere/M1/Config/` avec leurs attributs; les classer par ordre de taille décroissante. Compter le nombre de fichiers classés. Pourquoi ne retrouve-t'on pas le nombre précédent? Insérer un filtre pour éviter ce problème.

On s'intéresse maintenant à des fichiers d'entête pour le préprocesseur `cpp`

8. Afficher les lignes du fichier `/usr/include/limits.h` qui comportent la chaîne `MAX` (`grep`). Compter leur nombre.
9. Afficher les 3 premières lignes du fichier `/usr/include/limits.h` qui comportent la chaîne `MAX`.
10. AB Afficher les lignes du fichier `/usr/include/limits.h` qui comportent à la fois l'instruction préprocesseur `#define` en début de ligne (avec un nombre arbitraire de blancs entre `#` et `define`) et la chaîne `MAX`

**Ex. 7 : Fichier de notes avec awk A 15 min.**

Le fichier `notes.txt` contient les notes d'un groupe d'étudiants, précédées d'une ligne d'entête.

```

notes.txt
Nom Prenom Initiales cc1 cc2 examen oral 2011-2012
Legrand Jean JL 10 12 13 14
Dupont Yves YD 9 7 9 12
Durand Claire CD 8 12 11 10
Lefort Corine CL 15 14 14 12
Duran Alberto AD 14 13 11 09

```

Utiliser le filtre `awk` pour :

1. Afficher, pour chaque ligne du fichier `notes.txt`, le numéro de la ligne, le premier champ, le nombre de champs et le dernier champ. Afficher le nombre de lignes du fichier `notes.txt`
2. Afficher le fichier `notes.txt` en permutant les champs `Nom` et `Prenom`.
3. Extraire du fichier `notes.txt` les lignes des étudiants sans la ligne d'entête et stocker le résultat dans le fichier `notes.dat` avec lequel on travaillera par la suite.
4. Calculer la moyenne de l'épreuve d'oral pour l'ensemble du groupe d'étudiants et l'afficher.
5. Calculer la moyenne des notes de l'étudiant dont les initiales sont `YD` et l'afficher.
6. Afficher la liste des étudiants ayant obtenu une note comprise entre 10 et 14 (inclus) au premier contrôle continu et calculer la moyenne de ces notes.

**Ex. 8 : Analyse de fichier image en niveaux de gris AB 40 min.**

Les fichiers d'image de type `pnm` (portable any map qui n'est pas un format) regroupent trois types d'images non compressées, (noir et blanc, niveaux de gris ou 3 couleurs `RGB`) qui peuvent être stockées soit en format texte soit en format binaire plus compact. Ils comportent tous un entête spécifiant le type du fichier (de `P1` à `P7`), la géométrie de l'image et, pour les niveaux de gris et les `RGB`, la dynamique du gris ou des couleurs.

2. La commande `\ls` invoque la commande native `/bin/ls` au lieu de l'alias qui pourrait, selon les configurations de site, utiliser la colorisation automatique et, éventuellement, l'affichage des caractères `*`, `@` ou `/` en fin de nom pour respectivement, les fichiers exécutables, les liens et les répertoires.

|            |                  |                 |       |         |
|------------|------------------|-----------------|-------|---------|
| <b>pnm</b> | portable any map |                 | texte | binaire |
| <b>pbm</b> | portable bitmap  | noir et blanc   | P1    | P4      |
| <b>pgm</b> | portable greymap | niveaux de gris | P2    | P5      |
| <b>ppm</b> | portable pixmap  | couleur RGB     | P3    | P7      |

Ainsi, les fichiers d'image en niveaux de gris (**portable gray map**) existent sous les formats deux P2 (ascii) ou P5 (binaire). Ils comportent un en-tête de trois lignes en texte ascii décrivant successivement :

- ligne 1) le format de fichier sous la forme P2 dans le cas **pgm** ascii par exemple
- ligne 2) le nombre de colonnes (largeur) et le nombre de lignes (hauteur) de l'image
- ligne 3) la valeur maximale du niveau de gris qui représente le blanc

Suivent alors les valeurs des pixels en niveau de gris stockées en ascii pour un fichier de type P2. L'image est parcourue de gauche à droite et de haut en bas ; le nombre de pixels par ligne du fichier ascii ne détermine pas la géométrie de l'image, qui est spécifiée dans l'en-tête (**man pgm** pour plus de détails).

Le fichier **img-asc.pgm**, fourni sur le compte de l'UE, est un fichier ascii dans lequel les niveaux de gris sont des entiers compris entre 0 et 255 (blanc).

```

----- premières lignes du fichier img-asc.pgm -----
1 P2
2 92 124
3 255
4 158 125 133 111 130 140 124 232 255 224 254 255 226 230 226 222 208 194 179 197 144 174 171 162 153 155
5 160 157 158 155 165 173 170 207 237 233 223 216 216 236 250 255 252 242 255 255 248 252 247 253 251 244
6 248 255 254 242 246 242 255 247 255 240 254 255 242 255 255 244 239 214 230 195 223 197 207 202 236 204
7 230 199 238 255 204 172 185 162 217 136 79 57 89 113
8 162 66 102 93 179 188 148 192 177 242 209 198 211 164 225 168 218 131 133 142 182 153 170 144 152 134

```



### Travail sur le fichier **img-asc.pgm** **A**

1. Quelle commande affiche le nombre total de lignes du fichier (et non de l'image) **img-asc.pgm** ?
2. Quelle commande permet d'extraire les trois premières lignes du fichier **img-asc.pgm** et de les écrire dans un fichier **entete.txt** ?
3. Extraire la partie du fichier **img-asc.pgm** qui suit l'en-tête (soit de la ligne 4 à la fin), pour former le fichier **pixels.txt** sur lequel on travaillera dorénavant.

### Travail sur le fichier **pixels.txt**

1. **A** Quelle commande permet de compter le nombre de pixels de l'image ? Avec quelle commande vérifier à partir de l'en-tête ?
2. **AB** À l'aide du filtre **tr**, écrire les valeurs des niveaux de gris des pixels à raison d'un par ligne dans le fichier **listpix.txt**
3. **B** Reprendre les deux questions précédentes en utilisant **awk** sur le fichier **pixels.txt**.

### Travail sur le fichier **listpix.txt** **AB**

1. Quelle commande permet de vérifier le nombre total de pixels dans le fichier **listpix.txt** ?
2. Comment compter le nombre de pixels dont le niveau de gris est 200 dans le fichier **listpix.txt** ? Même question avec le niveau 20.
3. Comment afficher les valeurs des niveaux de gris par ordre décroissant ?
4. Comment afficher la valeur du niveau de gris du pixel le plus clair ? le plus foncé ?
5. Calculer le niveau de gris moyen de l'image.

**Ex. 9 : Image bitmap (éditeur interactif, sed et tr) AB 20 min.**

Le fichier texte `esperluette.pbm` est une image du caractère `&` en noir sur fond blanc au format `pbm` (**portable bitmap**) `ascii`, semblable à celui des images en niveaux de gris de l'exercice 8 :

- les deux premières lignes de ce fichier sont un en-tête que l'on ne modifiera pas : la première indique le type de fichier (`P1` pour `pbm ascii`), la seconde la géométrie de l'image (nombre de lignes et de colonnes) ;
- les lignes suivantes correspondent aux pixels noirs (1) ou blancs (0) de l'image parcourue de gauche à droite puis de haut en bas ; chaque ligne comporte au plus 70 caractères.

Copier ce fichier depuis le compte de l'UE. Afficher l'image qu'il représente avec `display` par exemple. L'objectif est de modifier ce fichier texte pour afficher le caractère `&` en blanc sur fond noir, via différentes méthodes :

1. avec un éditeur de texte interactif ;
2. avec le filtre `sed` ;
3. AB avec le filtre `tr` et d'autres filtres élémentaires.

**Ex. 10 : Trace d'une matrice carrée (awk) A 15 min.**

Le fichier `mat5-5.dat` stocke une matrice carrée 5 par 5. Sans faire d'hypothèse sur la taille de la matrice, écrire les programmes `awk` qui calculent :

1. la somme des éléments de sa deuxième colonne ;
2. la somme des éléments de sa troisième ligne ;
3. AB sa trace (somme des éléments diagonaux).

| mat5-5.dat |    |    |    |    |
|------------|----|----|----|----|
| 11         | 12 | 13 | 14 | 15 |
| 21         | 22 | 23 | 24 | 25 |
| 31         | 32 | 33 | 34 | 35 |
| 41         | 42 | 43 | 44 | 45 |
| 51         | 52 | 53 | 54 | 55 |

**Ex. 11 : Édition d'un fichier fortran 90 (sed, grep) B 15 min.**

On considère le fichier fortran 90 `trinome.f90`. Pour répondre aux questions suivantes, utiliser les filtres `grep`, `sed` ou un éditeur en repartant à chaque fois de la version originale de `trinome.f90`.

1. Faire une copie de sauvegarde du fichier `trinome.f90`.
2. Créer un fichier constitué des lignes 23 à 46 de `trinome.f90`.
3. Créer un fichier ne contenant que les lignes de commentaires de `trinome.f90`. On rappelle qu'en fortran 90, une ligne est un commentaire si son premier caractère non-blanc est un point d'exclamation (!).
4. Créer un fichier omettant tous les commentaires de `trinome.f90`
5. Ajouter des blancs après les virgules.
6. Insérer des blancs de part et d'autres des opérateurs arithmétiques "+-".
7. Passer toutes les occurrences des mots clefs `if`, `then`, `else`, et `endif` en majuscules.

**Ex. 12 : Mise en forme pour graphique (awk) B 20 min.**

On souhaite représenter graphiquement l'évolution de l'endettement consigné dans le fichier `depenses`.

1. Calculer l'axe des temps en jours depuis le début de l'année 2000. Pour simplifier, on prendra des mois de 30 jours et des années non bissextiles.
2. Ordonner les emprunts en fonction de ce numéro de jour.
3. Stocker dans un fichier de trois colonnes nommé `dettes.dat`, le numéro du jour, le montant de l'emprunt et la dette cumulée.
4. Visualiser ce fichier grâce à la commande `plot-dettes.sh`. Le shell-script `plot-dettes.sh` appelle le programme `gnuplot` qui permet de créer des graphiques à partir de la ligne de commande.
5. Repérer le processus associé à l'affichage graphique. Comment terminer l'affichage sans saisir `q` dans la fenêtre `gnuplot` ?

**Ex. 13 : Notes** **AB** **20 min.**

Le fichier `notes1.dat` contient les notes d'un groupe d'étudiants. Le point-virgule ";" peut tenir lieu de séparateur entre colonnes (comme dans les fichiers au format CSV exportés par les tableurs).

```
notes1.dat
Nom Prenom Initiales;cc1;cc2;ex;oral
Legrand Jean JL;10;12;13;14
Dupont Yves YD;9;7;9;12
Durand Claire CD;8;12;11;10
Lefort Corine CL;15;14;14;12
```

1. Utiliser la commande `awk` pour permuter les notes d'oral et d'examen (`ex`). Comment restituer le séparateur « ; » entre les notes en sortie ?
2. Calculer la moyenne par épreuve sur l'ensemble du groupe d'étudiants et l'afficher.
3. **B** Calculer la moyenne des notes pour chaque étudiant et l'afficher.
4. Utiliser la commande `awk` pour permuter nom et prénom.
5. **B** Reprendre la permutation nom-prénom avec la commande `sed` et conclure.
6. **B** Reprendre la permutation nom-prénom avec `awk` puis `sed` sur le fichier `notes1+blancs.dat`. Expliquer.