

---

**TD 08 : UNIX - REDIRECTIONS ET FILTRAGE**

---

**Note :** Ce TD sera effectué depuis une fenêtre de Terminal. Il s'agira d'utiliser les commandes du système pour effectuer chaque étape des exercices suivants. Même si il est possible d'utiliser l'explorateur de fichiers graphique pour comparer/vérifier le comportement des commandes exécutées, il ne faut pas oublier que le but est de comprendre et de maîtriser les commandes dans un environnement UNIX.

**Exercice 1** (*Lecture et Ecriture de fichiers*)

1. Pour ce TD, dans votre dossier personnel, créer un dossier TD8\_UNIX et se placer à l'intérieur.
2. Afficher le contenu du fichier les3M.txt de 2 manières : en utilisant les paramètres de la commande cat, puis en utilisant la redirection de l'entrée standard.
3. Quelle commande permet de compter les lignes/mots d'un fichier ? Afficher le nombre de lignes du fichier les3M.txt en utilisant les paramètres de la commande, puis afficher le nombre de mots du fichier les3M.txt en utilisant la redirection de l'entrée standard. Mise à part la valeur numérique affichée, quelle différence est visible entre les 2 résultats des commandes ?
4. En une seule commande, créer un fichier *hoy* qui contiendra la date du système
5. Créer un fichier vide Dartagnan.txt, ajouter dedans (à l'aide d'une commande) la phrase suivante : *Tous pour un, un pour tous !* et afficher le contenu du fichier pour vérifier.
6. En une seule commande, ajouter le texte suivant à la suite du contenu du fichier Dartagnan.txt : *Amaris* Afficher ensuite le contenu du fichier résultant.
7. Afficher le contenu du fichier soccer.csv, d'abord à l'aide de la commande cat, puis à l'aide de la commande more : Quelle différence entre les deux ?
8. Afficher le contenu du fichier soccer.csv, par groupes successifs de 3 lignes, et ce, à partir de la ligne 15 incluse. L'affichage est-il celui attendu ? Modifier la commande pour afficher les lignes correctement ?
9. Afficher le contenu du fichier soccer.csv en inversant l'ordre des lignes du fichier.

**Exercice 2** (*Tris de lignes*)

1. Trier le contenu du fichier soccer.csv pour afficher les joueurs par ordre alphabétique
2. Trier le contenu du fichier soccer.csv pour afficher les noms des pays par ordre décroissant.
3. Trier le contenu du fichier soccer.csv pour afficher les joueurs par poids, par ordre numérique décroissant (attention : on ne veut pas le résultat par ordre lexicographique). Que remarquez-vous concernant la dernière ligne affichée ?
4. reprendre la question précédente et stocker le résultat dans un fichier soccer\_weight.csv. Afficher le contenu du fichier soccer\_weight.csv.

**Exercice 3** (*Filtrage en ligne*)

1. Afficher seulement la première ligne du fichier soccer.csv.
2. Afficher les 100 premières lignes du fichier soccer.csv.
3. Afficher toutes les lignes du fichier soccer.csv à l'exception de la dernière.
4. Afficher toutes les lignes du fichier soccer.csv à l'exception des 70 dernières.
5. Afficher seulement la dernière ligne du fichier soccer.csv.
6. Afficher les 5 dernières lignes du fichier soccer.csv.
7. Afficher toutes les lignes du fichier soccer.csv à l'exception de la première.
8. Afficher toutes les lignes du fichier soccer.csv à l'exception des 100 premières.
9. Créer un fichier data.csv qui contient les lignes 12 à 14 inclus ainsi que les lignes 33 à 36 inclus du fichier soccer.csv. Il ne faut utiliser QUE les commandes head, tail, cat et les redirections de la sortie standard vers un fichier. Il est autorisé de créer des fichiers intermédiaires mais il faudra les supprimer à la fin des opérations.

**Exercice 4** (*Filtrage en colonne*)

1. Afficher le contenu du fichier */etc/passwd* en ne gardant QUE le nom des utilisateurs.
2. Afficher le contenu du fichier soccer.csv en ne gardant QUE le nom du joueur et sa date de naissance.
3. Afficher le contenu du fichier soccer.csv en ne conservant que les 4 premiers champs de chaque ligne.

4. Afficher le contenu du fichier soccer.csv en supprimant les noms de tous les joueurs.
5. Afficher le contenu du fichier soccer.csv en ne conservant QUE les noms des joueurs et les pays. L’affichage de sortie aura remplacé toutes les virgules par des points-virgules.

**Exercice 5** (*Redirections entre commandes*)

1. Afficher le contenu détaillé du répertoire /bin mais en faisant en sorte de l’afficher petit à petit, page par page à l’écran.
2. Créer un fichier nbBinFiles qui contiendra le nombre d’éléments présents dans le répertoire /bin.
3. Stocker le contenu du fichier soccer.csv entre les lignes 22 et 27 (incluses) dans le fichier results.txt, en une seule ligne de commandes. Ensuite, afficher le contenu du fichier results.txt pour vérifier.
4. Ajouter dans le fichier results.txt, les lignes 3 à 7 du fichier soccer.csv. Vérifier le résultat dans le fichier cible.
5. Afficher uniquement les noms des joueurs du fichier soccer.csv mais dans un ordre aléatoire.
6. Filtrer les noms des joueurs comme dans la question précédente, et vérifier qu’ils sont classés dans l’ordre croissant. Quel résultat obtient-on ? Trier les noms par ordre croissant et vérifier à nouveau. Refaire la manipulation avec un tri en ordre inverse et une vérification que les données sont bien triées en ordre inverse.
7. Afficher à l’écran les nom et age des joueurs présents dans le fichier soccer.csv, triés par age croissant d’abord, puis par nom décroissant ensuite. Le nom et l’age en sortie seront séparés par le caractère ‘:’.
8. Quelle différence y a-t-il entre un ‘pipe’ classique et l’option -exec de la commande find ?