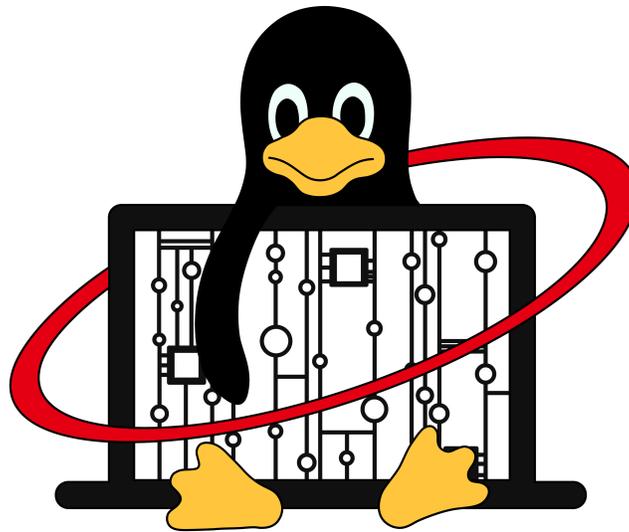


# Unix pour les nuls

Paul Slisse

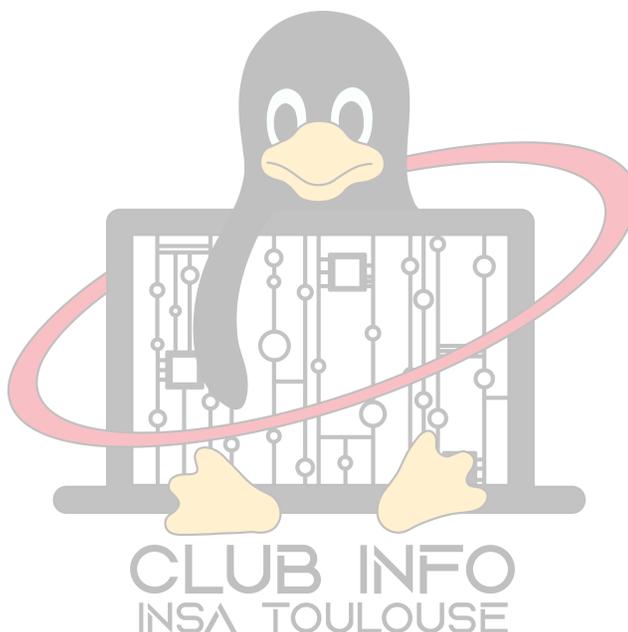
Novembre 2023



**CLUB INFO**  
INSA TOULOUSE

## Table des matières

<b>1</b>	<b>On commence doucement</b>	<b>3</b>
1.1	Manipulation de fichiers et répertoires . . . . .	3
1.2	Correction . . . . .	3
<b>2</b>	<b>Le petit script des familles</b>	<b>4</b>
2.1	Vérification d'Arguments et Traitement de Dossier . . . . .	4
2.2	Correction . . . . .	4



# 1 On commence doucement

## 1.1 Manipulation de fichiers et répertoires

Pour réaliser cette exercice vous pouvez utiliser un terminal pour réellement réaliser les manipulations mais nous vous demandons ensuite pour chaque question de recopier les commandes que vous avez utilisé dans un fichier texte afin de nous faciliter la correction.

1. Création et Navigation :  
Créez un répertoire appelé "ExerciceUnix" dans votre répertoire personnel. Allez dans ce répertoire.
2. Manipulation de Fichiers :  
Créez un fichier vide appelé "monFichier.txt". Ajoutez quelques lignes de texte à ce fichier en utilisant un éditeur de texte en ligne de commande, écrivez au moins une fois "Cybersecurity\_prodigy" dedans. Copiez ce fichier dans un nouveau fichier appelé "copieFichier.txt".
3. Manipulation de Répertoires :  
Créez un sous-répertoire dans "ExerciceUnix" appelé "SousDossier". Déplacez le fichier "monFichier.txt" dans le sous-répertoire "SousDossier".
4. Recherche et Filtrage :  
Utilisez la commande grep pour rechercher toutes les occurrences du mot "Cybersecurity\_prodigy" dans le fichier "monFichier.txt".
5. Permissions :  
Changez les permissions du fichier "monFichier.txt" pour que seul le propriétaire puisse le lire et le modifier.
6. Archivage et Compression :  
Créez une archive (au format tar) de tout le répertoire "ExerciceUnix". Compressez cette archive.
7. Nettoyage :  
Supprimez le répertoire "ExerciceUnix" et son contenu.

## 1.2 Correction

```

1.
1 cd
2 mkdir ExerciceUnix
3 cd ExerciceUnix

1 touch monFichier.txt
2 emacs (ou vi ou pluma ou tout diteur de texte) monFichier.txt &
3 (si pas le & ne pas oublier le Ctrl+Z apr s pour interrompre le
   processus)
4 cp monFichier.txt copieFichier.txt

2.
1 mkdir SousDossier
2 mv monFichier.txt SousDossier/

1 grep Cybersecurity_prodigy monFichier.txt

3.
1 chmod 600 monFichier.txt

1 tar -cvf ExerciceUnix.tar ExerciceUnix/
2 gzip ExerciceUnix.tar

4.
1 rm -r ExerciceUnix

```

## 2 Le petit script des familles

### 2.1 Vérification d'Arguments et Traitement de Dossier

Écrivez un script Bash appelé analyse.sh. Ce script devrait prendre deux arguments en ligne de commande : un répertoire et un fichier de sortie. Le script doit effectuer les étapes suivantes :

1. Vérification des Arguments :  
Assurez-vous que le nombre d'arguments est égal à 2. Vérifiez que le premier argument est un répertoire existant. Vérifiez que le deuxième argument est un fichier ou peut être créé.
2. Calcul du Poids du Dossier :  
Utilisez une commande pour obtenir le poids du répertoire sous forme lisible (conversion Ko Go Mo en fonction de la taille). "Indication : utilisez la commande cut et le man est votre meilleur ami" Ecrivez le nom du dossier et son poids dans le fichier de sortie (fichier de sortie= 2ème paramètre)
3. Nombre Total de Caractères dans Chaque Fichier :  
Utilisez une boucle pour parcourir tous les fichiers du répertoire. Pour chaque fichier, comptez le nombre total de caractères. Écrivez le nom du fichier et le nombre total de caractères dans le fichier de sortie.

Assurez-vous d'inclure des messages informatifs pour les utilisateurs en cas d'erreurs ou d'arguments invalides. N'oubliez pas de donner les droits d'exécution à votre script (chmod +x analyse.sh) avant de le tester. Vous pouvez ensuite tester avec un répertoire existant et un fichier.

### 2.2 Correction

```

1 #!/bin/bash
2
3 # Verification du nombre d'arguments
4 if [ "$#" -ne 2 ]; then
5     echo "Usage: $0 <repertoire> <fichier_de_sortie>"
6     exit 1
7 fi
8
9 # Verification que le premier argument est un repertoire existant
10 if [ ! -d "$1" ]; then
11     echo "Le premier argument doit etre un repertoire existant."
12     exit 1
13 fi
14
15 # Verification que le deuxieme argument est un fichier ou peut etre cree
16 if [ -e "$2" ] && [ ! -f "$2" ]; then
17     echo "Le deuxieme argument doit etre un fichier ou un fichier creable."
18     exit 1
19 fi
20
21 # Calcul du poids du repertoire et ecriture dans le fichier de sortie
22 poids=$(du -sh "$1" | cut -f1)
23 echo "Poids du repertoire '$1': $poids Ko." >> "$2"
24
25 # Traitement de chaque fichier
26 for fichier in "$1"/*; do
27     if [ -f "$fichier" ]; then
28         # Calcul du nombre total de caracteres
29         caracteres=$(wc -c < "$fichier")
30

```

```
31     # ecriture dans le fichier de sortie
32     echo "Fichier: $fichier, Caracteres: $caracteres" >> "$2"
33     fi
34 done
35
36 echo "Analyse terminee. Resultats ecrits dans '$2'."
```

