

Exercice 1 – Fichiers et arborescence (2 pts)

1) Créez dans votre répertoire personnel un dossier nommé `Examen_R108`.

`mkdir Examen_R108`

2) À l'intérieur, créez les sous-dossiers `work` et `result`.

`cd Examen_R108` : permet de se déplacer directement dans le répertoire

`mkdir work`

`mkdir result`

3) Créez dans `work` trois fichiers (`t1.txt`, `t2.txt`, `t3.txt`) contenant chacun une phrase différente.

`cd work`

`touch t1.txt, t2.txt, t3.txt`

4) Copiez tous les fichiers `.txt` de `work` vers `result`.

`cp -r work/. result`

`cp` permet de copier les fichiers du répertoire `work` vers le `result`

5) Affichez la structure complète du dossier `Examen_R108` avec les chemins absolus.

`ls -R Examen_R108`

affiche tout les répertoires du dossier `Examen_R108` ainsi que ces sous répertoires

Exercice 2 – Recherche et critères avec find et grep (2 pts)

1) Recherchez dans le répertoire `/bin` tous les fichiers exécutables dont le nom contient la lettre s.

```
cd /bin
```

```
find . -iname "s*" -type f
```

```
romain_droques@j004-11:/bin$ find . -iname "s*" -type f
./sensible-terminal
./sha224sum
./sccmap
./setpriv
./spd-conf
./ssh-import-id
./sort
./sb
./swtpm_setup
./snmpnetstat
./systemd-ask-password
./sbsignify
./snmptranslate
./setterm
./systemd-stdio-bridge
./ss
./scilab-cli
./switcherooctl
./snmpcheck
./systemd-dissect
./split
./sdptool
./sudo
```

2) Affichez uniquement les 5 premiers résultats.

```
find . -iname "s*" -type f | head -n 5
```

```
romain_droques@j004-11:/bin$ find . -iname "s*" -type f | head -n 5
./sensible-terminal
./sha224sum
./sccmap
./setpriv
./spd-conf
```

3) Sauvegardez ces résultats dans le fichier `executables.txt` dans le dossier `result`.

```
find . -iname "s*" -type f > /home/romain_droques/Examen_R108/executable.txt
```

copie tout les fichier exécutable de `/bin` vers un fichier `executable.txt`

4) Ajoutez à la fin du fichier la date du jour (commande `date`) et votre nom d'utilisateur (`whoami`).

```
find . -iname "*s*" -type f >  
/home/romain_droques/Examen_R108/result/executables.txt ; date >>
```

```
find . -iname "*s*" -type f >  
/home/romain_droques/Examen_R108/result/executables.txt ; whoami >>
```

Exercice 3 – Droits, comparaisons et redirections (2 pts)

1) Dans le dossier `Examen_R108`, créez un fichier `identite.txt` contenant votre nom, prénom et adresse mail.

```
cd Examen_R108
```

```
touch identite.txt
```

```
echo "Romain" > identite.txt
```

```
echo "Droques" >> identite.txt
```

```
echo "droquesr@gmail.com" >> identite.txt
```

2) Modifiez ses droits pour qu'il soit :

lisible et modifiable uniquement par le propriétaire,

```
chmod u+r identité.txt
```

lisible par le groupe,

```
chmod u+r,g+r identité.txt
```

invisible pour les autres.

```
chmod u+r,g+r,o-rwx identité.txt
```

3)Créez une copie `identite_copie.txt` et comparez les deux fichiers avec `diff`.

```
cp identité.txt identité_copie.txt
```

```
diff identite.txt identite_copie.txt
```

4)Enregistrez le résultat de la comparaison dans `comparaison.log`.

```
diff identite.txt identite_copie.txt > comparaison.log
```

5)Rediriger toute erreur éventuelle vers un fichier `erreurs.txt` (en l'ajoutant à la suite du fichier s'il existe déjà).

```
diff identite.txt identite_copie.txt > comparaison.log 2> erreur.txt
```

Exercice 4 – Filtres, tri et traitement de texte (2 pts)

1)Créez un fichier `employes.txt` contenant les lignes suivantes :

```
Alice 2300
```

```
Karim 1900
```

```
Sophie 2500
```

```
Hugo 1800
```

```
Alice 2300
```

```
Léa 2100
```

```
touch employes.txt
```

2)Triez le fichier selon la deuxième colonne (salaire) par ordre croissant.

```
sort -k2 employes.txt
```

3)Affichez uniquement les noms sans doublons.

```
cut -d" " -f1 employes.txt | sort | uniq
```

4)Affichez le nombre total de lignes du fichier original.

```
nl employes.txt
```

5)Cherchez toutes les lignes contenant la lettre “e” (insensible à la casse).

```
grep -E "e" employes.txt
```

Exercice 5 – Variables, scripts et gestion des processus (2 pts)

1) Créez une variable `LOGIN` contenant votre nom d'utilisateur et affichez sa valeur.

```
login=romain_droques
```

```
echo $login
```

2) Créez un script `infos_user.sh` qui :

affiche le nom d'utilisateur courant et le shell utilisé ;

affiche la date et l'heure du jour ;

```
echo $USER  
echo $SHELL  
date
```

demande à l'utilisateur son âge et affiche un message personnalisé :

Bonjour <user>, vous avez <âge> ans. Nous sommes le <date>.

```
read -p "quel est ton age " age  
"Bonjour $USER, vous avez $age ans. nous sommes le $(date)"
```

3) Rendez ce script exécutable et exécutez-le.

4) Lancez une commande `sleep 45` en arrière-plan, affichez son PID avec `ps`, puis terminez-la proprement avec `kill`.

5) a) Vérifiez qu'elle a bien été arrêtée.