Informatique *TP*
Système d'exploitation Ipesup

Attention

Travaillez *exclusivement* dans un répertoire d'atelier sous votre `$HOME`. Avant chaque commande « risquée », vérifiez le chemin avec `pwd` et un `ls` sec.

Exercice — Arborescence et chemins

1. Créez `$HOME/lab-os` puis les sous-répertoires `projets/os/{notes,tp}`, `donnees/{images,textes}`, `partage`.
2. Sans changer de répertoire, affichez le **contenu détaillé** de `lab-os/projets/os` (indice : `ls -l` avec un *chemin relatif*).
3. Donnez un *chemin absolu* et un *chemin relatif* vers `donnees/images` depuis `projets/os/tp`. Expliquez `.` et `...`

Exercice — Utilisateurs, groupes et permissions de base

1. Affichez vos identifiants avec `id`. Quel est votre `uid`, `gid` et vos groupes secondaires ?
2. Dans `donnees/textes`, créez `lisez-moi.txt` (contenu libre via `echo` & redirection). Donnez-lui les droits `rw-r---`. Montrez `ls -l` et expliquez chaque caractère.
3. Pour `donnees/images`, accordez au **groupe** le droit d'écriture, et retirez tout accès aux « autres ». Justifiez l'effet précis des bits `r/w/x` *sur un répertoire*.

Exercice

(20 min) Bits spéciaux : SUID, SGID, sticky (observation)

1. Inspectez `ls -ld /tmp`. Que signifie la lettre finale `t` ? Dans quel cas protège-t-elle contre les suppressions ?
2. Observez `ls -l /usr/bin/passwd`. Identifiez la lettre `s` et reliez-la à l'*EUID*.
3. Créez `$HOME/lab-os/partage/groupe` puis appliquez `chmod 2775 groupe`. Créez un fichier dedans et vérifiez le **groupe hérité**. Expliquez le rôle du bit **SGID** sur répertoire.

Exercice — Inodes et liens physiques

1. Dans `donnees/images`, créez `img_001.jpg` et `img_002.jpg` (contenu arbitraire via `echo`).
2. Créez **un lien physique** `a_imprimer/photo.jpg` vers `img_001.jpg`. Comparez `ls -li` des deux noms (même inode ? même compteur de liens ?).
3. Retirez la lecture pour groupe & autres sur `photo.jpg` et observez l'effet sur `img_001.jpg`. Expliquez pourquoi *les droits* sont partagés.
4. Supprimez `img_001.jpg`. Les données existent-elles encore ? Quand sont-elles réellement libérées ?

Exercice — Liens symboliques

1. Créez `projets/os/courant` → `projets/os/notes`. Interprétez la ligne `ls -l` (lettre `l`, flèche, « taille = longueur de la cible »).
2. Chainez `actuel` → `courant`, puis `dernier` → `actuel`. Que donne `cd dernier` ?
3. Renommez `notes` en `notes_old`. Qu'advient-il des liens ? Comparez lien **relatif** et **absolu**.

Exercice — Globbing et expansion

1. Donnez l'expansion de `img_???.jpg` et `img_[0-2][0-9][0-9].jpg` dans `donnees/images` (créez au besoin `img_012.jpg`, `img_223.jpg`…).
2. Proposez un motif qui *exclut* les fichiers dont le nom contient une majuscule (indice : classes de caractères et `[^]`).
3. Prouvez que l'expansion est faite par le **shell** (indice : `echo`).

Exercice — Redirections : stdout vs stderr

1. Exécutez `ls donnees/images img_999.jpg` en redirigeant la **sortie standard** vers `ok.txt` et la **sortie d'erreur** vers `err.txt`. Vérifiez le contenu des deux fichiers.
2. Quelle différence entre `>` et `»` ? Montrez-la concrètement.
3. Expliquez pourquoi l'ordre `2>&1` vs `> fichier 2>&1` change le résultat (indice : duplication de descripteurs).

Exercice — Pipes et petits traitements sur texte

1. Listez *tous* les fichiers directement sous `lab-os` et `donnees` avec `ls -l` ; triez par **taille décroissante** et affichez les **5** plus gros (`sort -k 5 -n -r | head -n 5`).
2. Calculez le **nombre total** d'entrées listées à l'aide d'un pipe vers `wc -l`. Écrivez ce nombre *en ajout* dans `stats.txt`.
3. À partir de `ls -l donnees/textes`, extrayez « taille ; nom » puis triez par taille croissante (`cut` puis `sort -n`). Ajoutez une ligne d'en-tête avec `echo »`.

Exercice — Options à découvrir dans le manuel

1. En consultant `man ls`, trouvez l'option qui liste **récursivement** (`-R`) et produisez la liste récursive de `lab-os` dans `arbre.txt`.
2. En consultant `man sort`, expliquez le sens de `-k 5 -n -r`. Donnez une variante qui trie par *nom de fichier* (ordre alphabétique).
3. En consultant `man chmod`, donnez l'écriture **octale** des modes suivants : `rw-r---`, `rwxr-sr-x`, `rwxrwxrwt`.

Exercice — Inventaire, liens et permissions

Objectif

Produire dans `$HOME/lab-os` un dossier `rapport/` contenant `rapport.txt` (lisible) et `inventaire.csv` (données « point-virgule »), en utilisant **exclusivement** les commandes de la leçon et leurs options (`ls`, `pwd`, `id`, `echo`, `cat`, `cut`, `sort`, `head`, `tail`, `wc`, `chmod`, `ln`, `cp`, `mv`, `rm`, `mkdir`) plus la redirection et les pipes.

Contenu attendu de `rapport.txt` :

1. **Contexte** : répertoire courant (`pwd`), utilisateur et groupes (`id`).
2. **Comptages** :
   — nombre total de fichiers et de répertoires sous `lab-os` (`ls -lR` + `wc -l`, méthode au choix) ;
   — nombre de fichiers dans `donnees/images` dont le nom matche `img_???.jpg`.
3. **Top 10** des entrées les plus volumineuses visibles via `ls -l` (tri par colonne taille, puis `head -n 10`).
4. **Liens physiques** : tableau listant, pour chaque inode dont le *compteur de liens* ≥ 2, les noms associés. (Indice : `ls -li` puis tri et regroupement avec `sort` et `cut` / `uniq -c` ou équivalent.)
5. **Permissions sensibles** : liste des fichiers *monde-écrits* dans `lab-os` (piste : repérez « `w` » dans la troisième triade de `ls -l`) et indiquez s'ils se trouvent dans un répertoire protégé par sticky (si vous en avez créé un).
6. **Résumé** : 5 lignes synthétiques expliquant ce que vous avez observé (liens, tailles, modes).

Contenu attendu de `inventaire.csv` (séparateur « ; ») :
une ligne par entrée `lab-os`, avec colonnes `inode;permissions;taille;nom`.
(Indice : `ls -liR` puis `cut` / `tr` pour formater.)

Contraintes :

— Pas de scripts externes (pas de C/Python). Uniquement redirections & pipes.
— Chaque commande clé utilisée pour générer le rapport doit apparaître dans `rapport.txt` (copiez la ligne de commande avec `echo` avant son exécution ou récapitulez-la en fin de section).
— Le projet doit être **rejouable** : supprimer `rapport/` et regénérer sans intervention manuelle.

Pistes d'évaluation (*non à rendre*) : clarté du pipeline, robustesse des tri/extractions, interprétation correcte des permissions et des liens.