

Examen Commandes UNIX

Nom :

Prénom:

Les machines et les documents sont interdits. Vous devez donc répondre aux questions suivantes en écrivant les commandes nécessaires sur cette même feuille. Vous supposerez que vous venez de lancer votre terminal, vous êtes donc placés dans le répertoire : ~

Vous N'AVEZ PAS LE DROIT d'utiliser la commande suivante : cd

Barème : 0.5 points par questions (il n'y a pas de 0.25 points)

1. Créer un nouveau répertoire : **Exam**, et son sous répertoire **SousExam**. . (2 commandes)

```
mkdir Exam
mkdir Exam/SousExam
```

2. Pouvez-vous faire l'action précédente avec une seule commande? Si oui laquelle, si non pourquoi ?

```
mkdir -p Exam/SousExam
```

3. Créer un fichier vide **notes.txt** et le placer dans **Exam** (2 commandes).

```
touch notes.txt
mv notes.txt Exam
```

4. Pouvez-vous faire l'action précédente avec une seule commande? Si oui laquelle, si non pourquoi ?

```
touch Exam/notes.txt
```

5. Effacer le répertoire **SousExam**. (1 commande)

```
rmdir Exam/SousExam
```

On suppose qu'on possède trois fichiers **notesA.txt**, **notesB.txt** et **notesC.txt**, placés dans votre répertoire utilisateur, qui contiennent les notes des élèves des 3 groupes A, B et C selon le format décrit ci-dessous :

```
Pierre Popaul : 20
Paul Pinel : 14
Pierre Blanche : 10
Thomas Gabriel : 12
Céline Done : 20
Cécile Blanche : 16
```

6. Copier le contenu de ces 3 fichiers à la suite dans votre fichier **notes.txt** (1 commande).

```
cat notesA.txt notesB.txt notesC.txt > Exam/notes.txt
```

7. Lister sur la sortie standard les lignes du fichier **notes.txt** contenant le chiffre 2 ou le chiffre 0 (1 commande).

```
grep -E '2|0' Exam/notes.txt ( grep [20] Exam/notes.txt )
```

8. Lister sur la sortie standard les lignes du fichier **notes.txt** qui contiennent le nombre 20, en faisant apparaître au début de chaque ligne le numéro de la ligne correspondant dans le fichier d'origine (1 commande).

```
grep -n 20 Exam/notes.txt
```

9. Modifier la commande précédente pour sauvegarder son résultat dans un fichier nommé **élèves_brillants.txt** (1 commande).

```
grep -n 20 Exam/notes.txt > élèves_brillants.txt
```

10. Renommer le fichier **notes.txt** en **notes2.txt** et placer ce dernier dans votre répertoire utilisateur (1 commande).

```
cp Exam/notes.txt notes2.txt
```

11. Supprimez le fichier **notes.txt** puis le répertoire **Exam** (2 commandes).

```
rm Exam/notes.txt  
rmdir Exam
```

12. Pouvez-vous faire l'action précédente avec une seule commande? Si oui laquelle, si non pourquoi ?

```
rm -R Exam
```

13. Afficher sur la sortie standard la liste des élèves de **notes2.txt** par ordre croissant de leurs notes (1 commande).

```
sort -k4n notes2.txt
```

14. Trier le fichier **notes2.txt** par ordre alphabétique des noms et des prénoms, et sauvegarder le résultat dans le fichier **fichier_trie.txt** (1 commande).

```
sort -k2 -k1 notes2.txt > fichier_trie.txt
```

15. Sauvegarder la liste des élèves triés sans leurs notes dans un fichier nommé **listing.txt** (1 commande).

```
sort -k2 -k1 notes2.txt | cut -f1,2 > listing.txt
```

16. Faites une recherche sur tout le disque afin de trouver le fichier nommé **fichier_trie.txt**. (1 commande).

```
find / -name "fichier_trie.txt"
```

17. La commande précédente génère des erreurs. Pourquoi ? (Justifier).

```
Permission Denied
```

18. Comment faire pour ne plus voir ces erreurs ? (1 commande).

```
find / -name "fichier_trie.txt" 2>/dev/null
```

19. Modifiez la commande précédente, afin de changer les droits de ce fichier, si celui-ci est trouvé. Les droits doivent être affectés ainsi :

- Le groupe propriétaire du fichier a le droit en lecture et en exécution
- L'utilisateur propriétaire du fichier a tous les droits
- Les autres utilisateurs n'ont aucun droit

```
find / -name "fichier_trie.txt" -exec chmod 750 {} \; 2>/dev/null
```

20. Trouver dans l'arborescence les fichiers de notes qui contiennent l'étudiant se prénommant : « Sacha ». (1 commande).

```
find / -name "notes*.txt" -exec grep -H Sacha {} \;
```

21. Extraire de tous les fichiers de notes, les notes et uniquement les notes des étudiants. (1 commande).

```
find / -name "notes*.txt" -exec cut -f4 {} \;
```

22. Comment faire pour compter le nombre de notes de la question précédente. (1 commande).

```
find / -name "notes*.txt" -exec cut -f4 {} \; | wc -l
```

23. À partir du fichier **notes2.txt**, faites apparaître dans l'ordre alphabétique les étudiants ayant eu **20**.

```
sort -k2 -k1 notes2.txt | grep 20
```

24. Que signifie le droit d'exécution sur un fichier ?

Exécuter un fichier exécutable

25. Et sur un répertoire ?

Droit de traverser ce répertoire

26. Que signifie le droit d'écriture sur un fichier ?

Modifier/Supprimer un fichier

27. Et sur un répertoire ?

Pour pouvoir effacer un fichier ou un répertoire vide, il suffit d'avoir les droits **wx** sur le répertoire contenant (avoir uniquement de droit **w** ne sert à priori à rien pour un répertoire).

Notez que dans un répertoire avec droit **wx**, on peut effacer les fichiers (ou répertoires vides) sur lesquels on n'a pas les droits (par contre, on ne peut pas effacer les répertoires non vides sur lesquels on n'a pas les droits **wx**).

28. Je veux donner tous les droits au propriétaire du fichier **notes2.txt** sans modifier les droits des autres utilisateurs. Quelle commande dois-je entrer ? (1 commande)

```
chmod u+rwx notes2.txt
```

29. À quoi sert le fichier **/etc/bash_profile** ?

À configurer les bashes utilisateurs. C'est le fichier de configuration qui est propagé par root.

30. Et le fichier **~/.bashrc** ?

Fichier de configuration utilisateur.

31. Pour la commande **grep**, à quoi correspond le motif : « **A?** » ?

Soit A suivi d'un caractère, soit un caractère

32. Et le motif : « **^a** » ?

Une ligne qui commence par a

33. Même question pour le motif : « **[^a]** » ?

Un autre caractère que a

34. Enfin pour en finir avec les motifs, à quoi correspond le motif : "**[sS]{,1}(ac){1}h+**" ?

Soit s ou S (0 ou 1 fois) suivi de ac suivi d'au minimum un h

35. Comment feriez vous pour rechercher *le caractère* : '\n' ?

```
grep '\n'
```

36. À quoi sert la commande **su** ?

Changer d'utilisateur, sans paramètre permet de devenir root

37. À quoi sert la commande **ssh** ?

La commande **ssh** permet de se connecter à un ordinateur distant et d'y exécuter des commandes.

38. À quoi sert la commande **scp** ?

La commande **scp** permet de transférer des fichiers entre des ordinateurs au moyen d'une connexion cryptée sécurisée.

39. Quelle est la différence entre un lien symbolique et un lien physique ?

Pour un lien physique la cible doit exister. C'est un autre nom permettant d'appeler un fichier. Pour supprimer le fichier, il faudra supprimer tous les liens physiques pointant sur ce fichier.

Un lien symbolique crée un "raccourci" vers un fichier. La cible n'est pas obligée d'exister. Si la cible n'existe pas ou plus, on dit que le lien est "cassé".

40. Quelle commande dois-je entrer pour créer un lien symbolique nommé **link** sur le fichier **notes2.txt** ? (1 commande).

```
ln -s notes2.txt lien
```