

Feuille de travaux pratiques – Corrigé

Ligne de commande

Ce TP a pour objectif de vous faire utiliser les commandes de base sous UNIX. Il se réalisera uniquement en ligne de commande.

1 Gestion de fichiers

Commandes pour cet exercice `cd`, `cp`, `mkdir`, `mv`, `ls`, `rm`, `rmdir`, `touch`.

1. Ouvrez un terminal, et déplacez vous dans vos documents.

▽ **Correction**

```
cd Documents
```

2. Créez un répertoire nommé `tp_infople`.

▽ **Correction**

```
mkdir tp_infople
```

3. Affichez le contenu du répertoire que vous venez de créer.

▽ **Correction**

```
ls tp_infople
```

4. Créez un fichier `tp1.txt`, quelle est la taille de ce fichier ?

▽ **Correction**

```
touch tp1.txt
```

5. Déplacez le fichier `tp1.txt` dans le répertoire `tp_infople`.

▽ **Correction**

```
cp tp1.txt tp_infople/.
```

6. Copiez le fichier `tp1.txt` dans votre répertoire de connexion sous le nom `tp1_copie.txt`.

▽ **Correction**

```
cp tp1.txt ~/tp1_copie.txt
```

7. Effacez le fichier `tp1_copie.txt`.

▽ **Correction**

```
rm ~/tp1_copie.txt
```

8. Créez un répertoire nommé `tmp` dans le répertoire `tp_infople`.

▽ **Correction**

```
mkdir tp_infople/tmp
```

9. Copiez le fichier `tp1.txt` dans le répertoire `tmp`.

▽ Correction

```
cp tp1.txt tp_infople/tmp/.
```

10. Renommez le fichier `tp1.txt` en `tp2.txt`.

▽ Correction

```
mv tp1.txt tp2.txt
```

11. Copiez dans le répertoire `tmp` tous les fichiers qui se trouvent dans `/usr/lib` dont le nom commence par `lib` et qui ont l'extension `.so`.

▽ Correction

```
cp /usr/lib/lib*.so tp_infople/tmp/.
```

12. Affichez la liste de tous les fichiers du répertoire `tmp` dont le nom comporte exactement 7 caractères.

▽ Correction

```
ls tp_infople/tmp/???????
```

13. Effacez du répertoire `tmp` les fichiers ayant l'extension `.so`.

▽ Correction

```
rm tp_infople/tmp/*.so
```

14. Copiez dans le répertoire `tmp` les fichiers qui se trouvent dans `/usr/lib` et dont le nom commence par `m`, `p` ou `l`.

▽ Correction

```
cp /usr/lib/[m,p,l]* tp_infople/tmp/.
```

15. Effacez tous les fichiers du répertoire `tmp`.

▽ Correction

```
rm tp_infople/tmp/*
```

16. Effacez le répertoire `tmp`.

▽ Correction

```
rmdir tp_infople/tmp
```

2 Redirection et affichage

Commandes pour cet exercice `cat`, `head`, `less`, `more`, `tail`, `wc`.

17. Déplacez vous dans le répertoire `tp_infople`.

▽ Correction

```
cd ~/tp_infople
```

18. Affichez le contenu du répertoire `/usr/lib`.

▽ Correction

```
ls /usr/lib/
```

19. Stockez la liste du contenu du répertoire `/usr/lib` dans le fichier `fichier.txt`.

▽ Correction

```
ls /usr/lib/ > fichier.txt
```

20. Ajoutez la liste du contenu de votre répertoire de connexion dans le fichier `fichier.txt`.

▽ Correction

```
ls ~ >> fichier.txt
```

21. Affichez le nombre de lignes dans `fichier.txt`.

▽ Correction

```
wc -l fichier.txt
```

22. Affichez le nombre de caractères dans `fichier.txt`.

▽ Correction

```
wc -c fichier.txt
```

23. Affichez le nombre de fichiers et dossiers de votre répertoire de connexion sans utiliser un fichier pour stocker le contenu.

▽ Correction

```
ls ~ | wc -l
```

24. Affichez le contenu du fichier `fichier.txt`.

▽ Correction

```
cat fichier.txt
```

25. Affichez le contenu du fichier `fichier.txt` avec le numéro de ligne.

▽ Correction

```
cat -n fichier.txt
```

26. Affichez les 25 premières lignes du fichier `fichier.txt`.

▽ Correction

```
head -n 25 fichier.txt
```

27. Affichez les 25 dernières lignes du fichier `fichier.txt`.

▽ Correction

```
tail -n 25 fichier.txt
```

28. Affichez le contenu du fichier `fichier.txt` “page” par “page”.

▽ Correction

```
more fichier.txt OU less fichier.txt
```

29. Affichez le contenu du fichier `fichier.txt` “page” par “page” à partir de la ligne 25.

▽ Correction

```
more +25 fichier.txt OU less +25 fichier.txt
```

3 Variables

La commande `echo` permet d’afficher ce qui est passé en argument.

30. Exécutez `echo "Ça change des maths"`.

▽ Correction

```
echo "Ça change des maths"
```

31. Affichez votre login.

▽ Correction

```
echo $USER
```

32. Affichez votre répertoire de connexion.

▽ Correction

```
echo $HOME
```

33. Exécutez `echo $PS1`.

▽ Correction

```
echo $PS1
```

34. Stockez la valeur de la variable PS1 dans le fichier `tmp.txt`.

▽ Correction

```
echo $PS1 > tmp.txt
```

35. Modifiez la variable PS1 pour que l'invite de commande soit "Bonjour toto >" où toto est votre nom d'utilisateur.

▽ Correction

```
Bash : PS1="Bonjour \u >", Zsh : PS1="Bonjour %n >"
```

36. Exécutez `echo -e "Bonjour\033[35m $USER \033[0m"`.
37. Exécutez `echo -e "Bonjour\033[1;36m $USER \033[0m"`.
38. Exécutez `echo -e "Bonjour\033[5;37;46m $USER \033[0m"`.

Pour changer la couleur et le style on commence par `\033[` les codes de changement, puis `m` et à la fin `\033[0m`. Les styles possible sont :

Couleur	Code texte	Code fond	Effet	Code
Noir	30	40	Gras	1
Rouge	31	41	Léger	2
Vert	32	42	Italique	3
Orange	33	43	Souligné	4
Bleu	34	44	Clignotant	5
Magenta	35	45	Surligné	7
Cyan	36	46		
Blanc	37	47		

4 Pour aller plus loin

39. Exécutez `man grep`.

▽ Correction

```
man grep
```

40. Récupérez le fichier <https://norvig.com/big.txt>.

▽ Correction

```
wget https://norvig.com/big.txt
```

41. Trouvez les lignes qui commencent par M ou m dans le fichier.

▽ Correction

```
grep "[Mm]" fichier
```

42. Trouvez les 2 lignes précédant celles qui commencent par P ou p dans le fichier.

▽ Correction

```
grep "[Pp]" -B 2 fichier
```

43. Trouvez la ligne suivant celles qui commencent par L ou l dans le fichier.

▽ Correction

```
grep "[Ll]" -A 1 fichier
```

44. Comptez les lignes qui finissent par E ou e dans le fichier.

▽ Correction

```
grep "[Ee]$" fichier | wc -l OU grep -c "[Ee]$" fichier
```

5 Connexions (sur les machines de l'université)

Commandes pour cet exercice [ssh](#), [ssh-keygen](#), [scp](#), [logout](#).

Pour se connecter à une machine distante, vous pouvez utiliser la commande [ssh](#). Pour ce faire il faut au préalable avoir généré et déposé sur le serveur une clé publique.

45. Créez une clé publique et privée à l'aide de la commande [ssh-keygen](#). On vous demandera d'entrer un mot de passe celui-ci est facultatif, afin de s'assurer de ne pas l'oublier laissez-le vide.

▽ Correction

```
ssh-keygen
```

46. Consultez les fichiers créés dans le répertoire `.ssh`, quelle est la clé publique ? Quelle est la clé privée ?

▽ Correction

```
ls .ssh, clé publique id_rsa.pub, clé privée id_rsa.
```

47. Copiez votre clé publique sous le nom `authorized_keys` dans le répertoire `.ssh`.

▽ Correction

```
cp .ssh/id_rsa.pub authorized_keys
```

48. Relevez le nom de la machine à côté de vous. Connectez-vous à cette machine à partir de son nom. Quelle est son adresse IP ?

▽ Correction

```
ssh rome 134.59.2.162
```

49. Une fois la connexion établie, créez un fichier ayant pour nom `jeSuisIci.txt`.

▽ Correction

```
touch jeSuisIci.txt
```

50. Déconnectez-vous de la machine voisine. Le fichier `jeSuisIci.txt` est-il toujours là ?

▽ Correction

```
logout, Oui
```

51. Depuis votre ordinateur personnel, connectez-vous à `sesame-mips.unice.fr`. Pour cela il faudra préciser votre nom d'utilisateur, par exemple [ssh](#) `ab123456@sesame-mips.unice.fr`.

▽ Correction

```
ssh ab123456@sesame-mips.unice.fr
```

52. Depuis votre ordinateur personnel, récupérez un fichier se trouvant dans votre répertoire de connexion sur les machines des salles de TP de l'université.

▽ Correction

```
scp ab123456@sesame-mips.unice.fr:fichier .
```

53. Depuis votre ordinateur personnel, copiez un fichier dans votre répertoire de connexion sur les machines des salles de TP de l'université.

▽ Correction

```
scp fichier ab123456@sesame-mips.unice.fr:.
```

6 Installation (sur votre machine personnelle)

Sous Debian ou Ubuntu, il est possible d'installer des programmes à l'aide de la commande `apt`. Sous Mac vous pouvez utiliser la commande `brew`, si vous ne l'avez pas encore installez-la <https://brew.sh/>.

54. S'il n'est pas déjà installé, installez `python3`.
55. Installez le programme `cowsay`.

▽ Correction

```
apt install cowsay OU brew install cowsay
```

56. Installez le programme `lolcat`.

▽ Correction

```
apt install lolcat OU brew install lolcat
```

57. Exécutez la commande suivante `cowsay -f stegosaurus "I am a legend"`.

▽ Correction

```
cowsay -f stegosaurus "I am a legend"
```

58. Utilisez la commande précédente et faites une séquence avec redirection avec la commande `lolcat`.

▽ Correction

```
cowsay -f stegosaurus "I am a legend" | lolcat
```

7 Exercice supplémentaire : les permissions

Commandes pour cet exercice `chmod`, `cp`, `gedit`, `ls`, `mkdir`, `touch`, `umask`.

59. Créez un fichier (vide) `fichier1` dans le répertoire `tp_infople`.

▽ Correction

```
touch ~/Documents/tp_infople/fichier1
```

60. Créez un répertoire (vide) `rep1` dans le répertoire `tp_infople`.

▽ Correction

```
mkdir ~/Documents/tp_infople/rep1
```

61. Affichez des informations détaillées sur les fichiers et répertoires contenus dans `tp_infople`. Repérez les permissions accordées à `fichier1` et `rep1`.

▽ Correction

```
ls -l ~/Documents/tp_infople Pour fichier1, -rw-rw-r-- (644), pour rep1, drwxr-xr-x (755).
```

62. Ouvrez le fichier `fichier1` avec `gedit`, écrivez votre nom dans ce fichier, puis sauvegardez.

▽ Correction

```
gedit ~/Documents/tp_infople/fichier1
```

63. Copiez `fichier1` en `fichier2`.

▽ Correction

```
cp ~/Documents/tp_infople/fichier1 ~/Documents/tp_infople/fichier2
```

64. Utilisez la commande `chmod` pour enlever le droit d'écriture à `fichier2`.

▽ Correction

```
chmod -w ~/Documents/tp_infople/fichier2
```

65. Pouvez-vous alors ouvrir et modifier `fichier2` avec `gedit` ?

▽ Correction

On peut lire le fichier, mais pas le modifier.

66. Tapez `umask` (sans paramètre). Quelle est la valeur obtenue ?

▽ Correction

022

67. En utilisant la commande `umask`, changez la valeur du masque à 136. Créez un fichier `fichier3` et un répertoire `rep2`. Affichez leurs permissions.

▽ Correction

```
umask 136; touch ~/Documents/tp_infople/fichier3; mkdir ~/Documents/tp_infople/rep2 Les permissions sont -rw-r----- et drw-r-----x respectivement.
```

68. Utilisez la commande `umask` pour qu'à l'avenir, les permissions des fichiers soient

- (a) `-r--r-----`
- (b) `-rw-rw----`

69. Avez-vous bien compris ?

- (a) Vous voulez modifier les permissions d'un fichier qui existe déjà : utilisez-vous `chmod` ou `umask` ?

▽ Correction

`chmod`

- (b) Vous voulez créer de nombreux fichiers avec les mêmes permissions : utilisez-vous `chmod` ou `umask` ?

▽ Correction

`umask`, même si on peut aussi utiliser `chmod`

8 Exercice supplémentaire : fichier de script

70. Dans le répertoire `tp_infople`, créez un fichier `tp1.sh` : sa première ligne doit être `#!/bin/sh` et il contient ensuite les commandes pour

- (a) afficher "Bonjour",
- (b) écrire dans le fichier `jour.txt` la date et l'heure. Vous pourrez utiliser la commande `date`.

▽ Correction

```
echo "Bonjour" rm -f jour.txt date > jour.txt
```

71. En utilisant la commande `sh`, exécutez le fichier `tp1.sh`.

▽ Correction

```
sh tp1.sh
```

72. Changez les permissions de `tp1.sh` pour pouvoir exécuter `tp1.sh` sans utiliser `sh`.

▽ Correction

```
chmod +x tp1
```

9 Rappel de quelques commandes

<code>man</code>	retourne le manuel de la commande s'il existe. <code>man</code> [options] nom_commande ex : <code>man man</code>
<code>cd</code>	permet de se déplacer dans le système de fichiers. <code>cd</code> repertoire ex : <code>cd toto</code>
<code>cd ..</code>	permet de se déplacer dans le répertoire parent.
<code>ls</code>	liste le contenu du répertoire courant. <code>ls</code> [option] [fichier] ex : <code>ls</code>
<code>ls -a</code>	liste le contenu du répertoire courant, sans ignorer les fichiers commençant par le symbole . <code>ls -a</code> [fichier] ex : <code>ls -a</code>
<code>ls -l</code>	liste le contenu du répertoire courant, en utilisant un format d'affichage long. <code>ls -l</code> [fichier] ex : <code>ls -l</code>
<code>ls -al</code>	liste le contenu du répertoire courant, sans ignorer les fichiers commençant par le symbole . et en utilisant un format d'affichage long. <code>ls -al</code> [fichier] ex : <code>ls -al</code>
<code>pwd</code>	affiche le nom du répertoire courant. <code>pwd</code> [option] ex : <code>pwd</code>
<code>mkdir</code>	crée un répertoire. <code>mkdir</code> [option] repertoire ex : <code>mkdir toto</code>
<code>rmdir</code>	supprime un répertoire vide. <code>rmdir</code> [option] repertoire ex : <code>rmdir essai</code>
<code>cp</code>	copie un fichier. <code>cp</code> [option] source destination ex : <code>cp essai TP1/</code>
<code>cp -R</code>	copie un répertoire de manière récursive. <code>cp -R</code> source destination ex : <code>cp -R TP1 TP2</code>
<code>mv</code>	déplace/renomme un fichier. <code>mv</code> [option] source destination ex : <code>mv essai TP1</code>

<code>rm</code>	efface un fichier. <code>rm [option] fichier</code> ex : <code>rm</code> essai
<code>rm -R</code>	efface récursivement un répertoire et son contenu. <code>rm -R fichier</code> ex : <code>rm -R</code> TP1
<code>echo</code>	affiche une ligne de texte. <code>echo [option_courte] [chaine_de_caracteres]</code> ex : <code>echo</code> "bonjour"
<code>history</code>	montre à l'utilisateur les dernières commandes utilisées. <code>history [nombre]</code> ex : <code>history</code> 5
<code>touch</code>	modifie les horaires des dernières modifications du fichier. Si le fichier passé en argument n'existe pas, le fichier est créé et est vide. <code>touch fichier</code> ex : <code>touch</code> essai
<code>cat</code>	concatène des fichiers et affiche le résultat sur la sortie standard. <code>cat [option] [file]</code> ex : <code>cat</code> essai essai
<code>wc</code>	compte les caractères, les mots et les lignes en entrée. <code>wc [option] [file]</code> ex : <code>wc</code> essai
<code>more</code>	affiche à l'écran l'entrée standard (page par page). <code>more [option] [file]</code> ex : <code>more</code> essai
<code>less</code>	idem avec possibilité de retour en arrière. <code>less [option] [file]</code> ex : <code>less</code> essai
<code>head</code>	affiche les n première lignes. Par défaut n vaut 10. <code>head [option] [file]</code> ex : <code>head</code> essai
<code>tail</code>	affiche les n dernières lignes. Par défaut n vaut 10. <code>tail [option] [file]</code> ex : <code>tail</code> essai
<code>grep</code>	recherche des chaînes de caractères. <code>grep [option] [file]</code> ex : <code>grep</code> toto