

Feuille de travaux pratiques

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.
2. Créez un répertoire nommé `tp_infople`.
3. Affichez le contenu du répertoire que vous venez de créer.
4. Créez un fichier `tp1.txt`, quelle est la taille de ce fichier ?
5. Déplacez le fichier `tp1.txt` dans le répertoire `tp_infople`.
6. Copiez le fichier `tp1.txt` dans votre répertoire de connexion sous le nom `tp1_copie.txt`.
7. Effacez le fichier `tp1_copie.txt`.
8. Créez un répertoire nommé `tmp` dans le répertoire `tp_infople`.
9. Copiez le fichier `tp1.txt` dans le répertoire `tmp`.
10. Renommez le fichier `tp1.txt` en `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`.
12. Affichez la liste de tous les fichiers du répertoire `tmp` dont le nom comporte exactement 7 caractères.
13. Effacez du répertoire `tmp` les fichiers ayant l'extension `.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`.
15. Effacez tous les fichiers du répertoire `tmp`.
16. Effacez le répertoire `tmp`.

2 Redirection et affichage

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

17. Déplacez vous dans le répertoire `tp_infople`.
18. Affichez le contenu du répertoire `/usr/lib`.
19. Stockez la liste du contenu du répertoire `/usr/lib` dans le fichier `fichier.txt`.
20. Ajoutez la liste du contenu de votre répertoire de connexion dans le fichier `fichier.txt`.
21. Affichez le nombre de lignes dans `fichier.txt`.
22. Affichez le nombre de caractères dans `fichier.txt`.
23. Affichez le nombre de fichiers et dossiers de votre répertoire de connexion sans utiliser un fichier pour stocker le contenu.
24. Affichez le contenu du fichier `fichier.txt`.
25. Affichez le contenu du fichier `fichier.txt` avec le numéro de ligne.
26. Affichez les 25 premières lignes du fichier `fichier.txt`.
27. Affichez les 25 dernières lignes du fichier `fichier.txt`.
28. Affichez le contenu du fichier `fichier.txt` "page" par "page".
29. Affichez le contenu du fichier `fichier.txt` "page" par "page" à partir de la ligne 25.

3 Variables

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

30. Exécutez `echo "Ça change des maths"`.
31. Affichez votre login.
32. Affichez votre répertoire de connexion.
33. Exécutez `echo $PS1`.
34. Stockez la valeur de la variable `PS1` dans le fichier `tmp.txt`.
35. Modifiez la variable `PS1` pour que l’invite de commande soit “Bonjour toto >” où `toto` est votre nom d’utilisateur.
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`.
40. Récupérez le fichier <https://norvig.com/big.txt>.
41. Trouvez les lignes qui commencent par `M` ou `m` dans le fichier.
42. Trouvez les 2 lignes précédant celles qui commencent par `P` ou `p` dans le fichier.
43. Trouvez la ligne suivant celles qui commencent par `L` ou `l` dans le fichier.
44. Comptez les lignes qui finissent par `E` ou `e` dans le 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.
46. Consultez les fichiers créés dans le répertoire `.ssh`, quelle est la clé publique ? Quelle est la clé privée ?
47. Copiez votre clé publique sous le nom `authorized_keys` dans le répertoire `.ssh`.
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 ?
49. Une fois la connexion établie, créez un fichier ayant pour nom `jeSuisIci.txt`.
50. Déconnectez-vous de la machine voisine. Le fichier `jeSuisIci.txt` est-il toujours là ?

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/>.

51. S’il n’est pas déjà installé, installez `python3`.

52. Installez le programme `cowsay`.
53. Installez le programme `lolcat`.
54. Exécutez la commande suivante `cowsay -f stegosaurus "I am a legend"`.
55. Utilisez la commande précédente et faites une séquence avec redirection avec la commande `lolcat`.

7 Exercice supplémentaire : les permissions

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

56. Créez un fichier (vide) `fichier1` dans le répertoire `tp_infople`.
57. Créez un répertoire (vide) `rep1` dans le répertoire `tp_infople`.
58. Affichez des informations détaillées sur les fichiers et répertoires contenus dans `tp_infople`. Repérez les permissions accordées à `fichier1` et `rep1`.
59. Ouvrez le fichier `fichier1` avec `gedit`, écrivez votre nom dans ce fichier, puis sauvegardez.
60. Copiez `fichier1` en `fichier2`.
61. Utilisez la commande `chmod` pour enlever le droit d'écriture à `fichier2`.
62. Pouvez-vous alors ouvrir et modifier `fichier2` avec `gedit` ?
63. Tapez `umask` (sans paramètre). Quelle est la valeur obtenue ?
64. En utilisant la commande `umask`, changez la valeur du masque à 136. Créez un fichier `fichier3` et un répertoire `rep2`. Affichez leurs permissions.
65. Utilisez la commande `umask` pour qu'à l'avenir, les permissions des fichiers soient
 - (a) `-r--r-----`
 - (b) `-rw-rw----`
66. Avez-vous bien compris ?
 - (a) Vous voulez modifier les permissions d'un fichier qui existe déjà : utilisez-vous `chmod` ou `umask` ?
 - (b) Vous voulez créer de nombreux fichiers avec les mêmes permissions : utilisez-vous `chmod` ou `umask` ?

8 Exercice supplémentaire : fichier de script

67. 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`.
68. En utilisant la commande `sh`, exécutez le fichier `tp1.sh`.
69. Changez les permissions de `tp1.sh` pour pouvoir exécuter `tp1.sh` sans utiliser `sh`.

9 Rappel de quelques commandes

<code>man</code>	retourne le manuel de la commande s'il existe. <code>man [options] nom_commande</code> ex : <code>man man</code>
<code>cd</code>	permet de se déplacer dans le système de fichiers. <code>cd repertoire</code> 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 [option] [fichier]</code> ex : <code>ls</code>
<code>ls -a</code>	liste le contenu du répertoire courant, sans ignorer les fichiers commençant par le symbole <code>.</code> <code>ls -a [fichier]</code> 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 [fichier]</code> 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 [fichier]</code> ex : <code>ls -al</code>
<code>pwd</code>	affiche le nom du répertoire courant. <code>pwd [option]</code> ex : <code>pwd</code>
<code>mkdir</code>	crée un répertoire. <code>mkdir [option] repertoire</code> ex : <code>mkdir toto</code>
<code>rmdir</code>	supprime un répertoire vide. <code>rmdir [option] repertoire</code> ex : <code>rmdir essai</code>
<code>cp</code>	copie un fichier. <code>cp [option] source destination</code> ex : <code>cp essai TP1/</code>
<code>cp -R</code>	copie un répertoire de manière récursive. <code>cp -R source destination</code> ex : <code>cp -R TP1 TP2</code>
<code>mv</code>	déplace/renomme un fichier. <code>mv [option] source destination</code> ex : <code>mv essai TP1</code>
<code>rm</code>	efface un fichier. <code>rm [option] fichier</code> ex : <code>rm essai</code>
<code>rm -R</code>	efface récursivement un répertoire et son contenu. <code>rm -R fichier</code> ex : <code>rm -R TP1</code>
<code>echo</code>	affiche une ligne de texte. <code>echo [option_courte] [chaine_de_caracteres]</code> ex : <code>echo "bonjour"</code>
<code>history</code>	montre à l’utilisateur les dernières commandes utilisées. <code>history [nombre]</code> ex : <code>history 5</code>
<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 essai</code>
<code>cat</code>	concatène des fichiers et affiche le résultat sur la sortie standard. <code>cat [option] [file]</code> ex : <code>cat essai essai</code>
<code>wc</code>	compte les caractères, les mots et les lignes en entrée. <code>wc [option] [file]</code> ex : <code>wc essai</code>
<code>more</code>	affiche à l’écran l’entrée standard (page par page). <code>more [option] [file]</code> ex : <code>more essai</code>
<code>less</code>	idem avec possibilité de retour en arrière. <code>less [option] [file]</code> ex : <code>less essai</code>
<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</code> [option] [file] ex : <code>tail</code> essai
<code>grep</code>	recherche des chaînes de caractères. <code>grep</code> [option] [file] ex : <code>grep</code> toto