
Correction des TPs unix

ANNE CADIOU, LAURENT POUILLOUX

2017-11-7

Atelier Introduction aux commandes Unix

Table des matières

- Prise en main de l'environnement
- Commandes de base
 - Naviguer dans les répertoires
 - Enchaîner des commandes
 - Code d'erreur d'un processus
 - Entrée/Sortie
 - Nettoyage
 - Quelques commandes pour faire une pause
- Manipulation de fichiers
 - Afficher le contenu d'un fichier
 - Éditer des fichiers
 - Trouver un fichier

Prise en main de l'environnement

- Chercher dans les menus le logiciel Terminal ou Console pour ouvrir un terminal.
- Ouvrir un terminal par un clic droit sur le bureau.
- Ouvrir 2 terminaux. En fermer un avec la souris, l'autre au clavier.

```
1 ~$ exit
```

Commandes de base

L'accès au manuel d'usage des commandes se fait par la commande **man**.

```
1 ~$ man man
```

Quel est d'après le manuel le descriptif (NAME) de cette commande ?

```
1 man - an interface to the on-line reference manuals
2 man - Interface de consultation des manuels de référence en ligne
```

Naviguer dans les répertoires

- Vous venez d'ouvrir un terminal. Dans quel répertoire vous trouvez-vous ?

```
1 À la racine de mon compte.  
2  
3 ~$ pwd
```

- Exécutez les commandes suivantes :

```
1  
2 ~$ pwd  
3  
4 ~$ cd /tmp  
5  
6 ~$ pwd  
7  
8 ~$ cd
```

Où êtes-vous ?

```
1 À la racine de mon compte.
```

- Affichez le contenu des variables définissant votre répertoire à la racine de votre compte et celui où vous êtes actuellement

```
1 ~$ echo $HOME  
2  
3 ~$ echo ~  
4  
5 ~$ echo $PWD
```

- Sous quel shell travaillez-vous ?

```
1 ~$ echo $SHELL
```

- Quel est le nom de votre machine ?

```
1 ~$ uname -n  
2  
3 ~$ hostname  
4  
5 ~$ echo $HOSTNAME
```

- Quel est votre login ?

```
1 ~$ whoami
2
3 ~$ echo $USER
```

- À quels groupes unix appartenez-vous ?

```
1 ~$ groups
2
3 ~$ id
```

- Quel est votre numéro d'utilisateur ?

```
1 ~$ id
2
3 ~$ echo $UID
```

- Créez un répertoire de nom UNIX à la racine. Y créer le dossier TP1. Descendez dans ce répertoire TP1.

```
1 ~$ mkdir ~/UNIX
2
3 ~$ mkdir ~/UNIX/TP1
4
5 ~$ cd ~/UNIX/TP1
6
7 ~/UNIX/TP1$ pwd
```

- Remontez d'un répertoire pour vous retrouver dans le répertoire UNIX.

```
1 ~/UNIX/TP1$ cd ..
2
3 ~/UNIX$ pwd
```

- Placez-vous à la racine et faites un aller-retour dans le répertoire TP1 en une seule ligne.

```
1 ~/UNIX$ cd; cd ~/UNIX/TP1; cd
```

- Créez le répertoire TP2 dans UNIX. Créez un dossier exo1 dans les répertoires TP1 et TP2.

```
1 ~$ mkdir ~/UNIX/TP1/exo1
2
3 ~$ mkdir ~/UNIX/TP2; mkdir ~/UNIX/TP2/exo1
```

- Placez-vous dans le répertoire `exo1` de TP2 et faites un aller-retour dans le répertoire `exo1` de TP1.

```
1 ~$ cd ~/UNIX/TP1/exo1
2
3 ~/UNIX/TP1/exo1$ pushd ../../TP2/exo1
4 ~/UNIX/TP2/exo1$ pushd
```

- Placez-vous dans à votre racine. Listez le contenu du répertoire UNIX avec la commande `ls`.

```
1 ~$ ls -l ~/UNIX
```

- Affichez l'arborescence du répertoire UNIX à l'aide de la commande `tree`.

```
1 ~$ tree ~/UNIX
```

- Listez les fichiers ou dossiers cachés.

```
1 ~$ ls -la ~/UNIX
```

- Détruisez le répertoire `exo1` de TP2 de deux manières différentes

```
1 ~$ rm -r ~/UNIX/TP2/exo1
2
3 ou
4
5 ~$ rmdir ~/UNIX/TP2/exo1
```

- Descendez dans le répertoire UNIX. Copiez TP1 dans un nouveau répertoire TP1bis à l'aide de la commande `cp`.

```
1 ~/UNIX$ cp -r TP1 TP1bis
2
3 ou
4
5 ~/UNIX$ cp -R TP1 TP1bis
```

- Copiez à nouveau TP1 dans TP1bis. Que se passe-t'il ?

```
1 ~/UNIX$ cp -r TP1 TP1bis
2 ~/UNIX$ tree
3 .├──
```


- Afficher la liste des répertoires du plus récent au plus ancien puis inversement.

```
1 ~/UNIX$ ls -lt
2
3 ~/UNIX$ ls -lrt
```

- Afficher la taille des répertoires en bloc et en kilo octet.

```
1 ~/UNIX$ ls -s
2
3 ~/UNIX$ ls -l
4
5 ~/UNIX$ ls -lh
```

Enchaîner des commandes

- Quel que soit le résultat de la commande

```
1 ~/UNIX$ cd /tmp ; ls -l ; cd -
```

- si la commande précédent s'est correctement effectuée

```
1 ~/UNIX$ cd /tmp && ls -l && cd -
```

- si la commande précédent ne s'est pas correctement effectuée

```
1 ~/UNIX$ cd /tmp && ls -l || cd -
```

Code d'erreur d'un processus

- Connaitre le code de retour

```
1 ~/UNIX$ ls -l ; echo $?
2 0 si le processus s'est déroulé correctement
3 ~/UNIX$ ls -l no* ; echo $?
4 une autre valeur sinon
```

Entrée/Sortie

Chaque commande a une **entrée standard**, une **sortie standard** et une **sortie d'erreur**. Par défaut, l'entrée standard est le clavier, la sortie standard est l'écran, et la sortie d'erreur est

aussi l'écran. Cela peut être modifié par des **redirections**.

- Redirigez la sortie standard de la commande `ls -l` vers le fichier `res.txt`. Vous pouvez en visualiser le contenu avec la commande `cat`, explicitée à la section suivante.

```
1 ~UNIX$ ls -l > res.txt
2 ~UNIX$ cat res.txt
```

- Quelles est la différence entre

```
1 ~UNIX$ ls -l > res.txt
```

et

```
1 ~UNIX$ ls -l >> res.txt
```

```
1 > redirige la sortie en écrasant le contenu du fichier initial alors que
  >> ajoute la sortie de la commande à la fin du fichier initial.
```

- À l'aide de la commande `cat` et des chevrons d'entrée et sortie, copiez le contenu de `res.txt` dans un nouveau fichier `new.txt`

```
1 ~UNIX$ cat > new.txt < res.txt
```

- Essayez de faire afficher la liste de fichiers qui n'existent pas (par exemple commençant par `no`). Que se passe-t'il ?

```
1 ~UNIX$ ls -l no*
2 ls: cannot access 'no*': No such file or directory
3
4 Un message d'erreur s'affiche à l'écran.
```

- Recommencez en redirigeant la sortie standard vers le fichier `res.txt`

```
1 ~UNIX$ ls -l no* > res.txt
2 ls: cannot access 'no*': No such file or directory
3
4 Un message d'erreur s'affiche à l'écran.
5 Le fichier res.txt est vide
```

- Recommencez en redirigeant la sortie standard et la sortie d'erreur vers `res.txt`

```
1 ~UNIX$ ls -l no* > test.txt 2>&1
```

```
2  
3 Rien ne s'affiche à l'écran mais test.txt contient le message d'erreur  
4 ls: cannot access 'no*': No such file or directory
```

- Recommencez en affichant un message si vous en avez trouvé

```
1 ~UNIX$ ls -l no* > test.txt 2>&1 && echo "Found!"
```

- Recommencez en affichant un message si vous n'en avez pas trouvé

```
1 ~UNIX$ ls -l no* > test.txt 2>&1 || echo "Not Found!"
```

Nettoyage

- Supprimer tous les répertoires de UNIX. **ATTENTION : pas de “corbeille”**

```
1 ~/UNIX$ rm -fr *
```

Quelques commandes pour faire une pause

- Quel jour sommes-nous ? Utilisez la commande **date**.

```
1 ~$ date
```

- Quel jour de la semaine êtes-vous né ? Utilisez la commande **cal**.

```
1 Alan Turing est né le 23 juin 1912.  
2  
3 ~$ cal 6 1912  
4  
5 C\ 'était un dimanche.
```

- Où se trouve la commande **man** ?

```
1 ~$ whereis man
```

- Quelle commande **man** utilisez-vous ?

```
1 ~$ which man
```

Manipulation de fichiers

Afficher le contenu d'un fichier

- Afficher le contenu du répertoire `/etc/dictionaries-common`

```
1 ~$ ls -l /etc/dictionaries-common
```

- À qui appartient le fichier nommé `words` ?

```
1 à root
```

- Quel type de fichier est-ce ?

```
1 un fichier ascii, alias qui pointe sur /usr/share/dict/american-english  
(chez moi)
```

- Qui a les droits en lecture sur ce fichier ?

```
1 tout le monde.
```

- Descendre dans ce répertoire `/etc/dictionaries-common` et afficher le fichier `words` à l'aide des commandes `cat`, `less` et `more`.
- À l'aide de la commande `less` chercher les mots commençant par `Sco`

```
1 /etc/dictionaries-common$ less words  
2  
3 puis tapez /Sco  
4  
5 quittez less en tapant q
```

- Combien de lignes comporte ce fichier en tout ? Utilisez la commande `wc`.

```
1 /etc/dictionaries-common$ wc -l  
2 99171 words
```

- Combien de lignes comprenant les caractères `Sco` il y a t'il dans ce fichier ? Utilisez les commandes `grep` et `wc`

```
1 /etc/dictionaries-common$ grep Sco words |wc -l  
2 35 words
```

- Combien de lignes comprenant les caractères `Sco` ou `sco` il y a t'il dans ce fichier ?

```
1 /etc/dictionaries-common$ grep -i Sco words |wc -l
2 456 words
```

- Retournez dans le répertoire UNIX créé précédemment à la racine de votre compte. Copiez-y les fichier words de /etc/dictionaries-common.

```
1 /etc/dictionaries-common$ cd ~/UNIX
2 ~/UNIX$ cp /etc/dictionaries-common/words .
```

- À qui appartient ce fichier ?

```
1 à vous.
```

- Quels sont les droits unix sur ce fichier ?

```
1 -rw-r--r--
2
3 l'utilisateur peut lire et écrire, le groupe et les autres peuvent lire
  le fichier.
```

- Supprimer les droits en lecture des autres

```
1 ~/UNIX$ chmod o-r words
2
3 ou
4
5 ~/UNIX$ chmod 640 words
```

- Faire un lien symbolique de ce fichier vers un nouveau fichier nommé mots.

```
1 ~/UNIX$ ln -s words mots
2
3 ~/UNIX$ ls -l
```

- Afficher les 3 premiers mots de la liste words puis les 5 derniers à l'aide des commandes **head** et **tail**.

```
1 ~/UNIX$ head -3 words
2
3 ~/UNIX$ tail -5 words
```

- Rediriger les 3 premiers mots de la liste words dans un nouveau fichier nommé extrait

```
1 ~/UNIX$ head -3 words > extrait
2
3 ~/UNIX$ cat extrait
```

- Ajouter les 5 derniers mots de words au fichier extrait.

```
1 ~/UNIX$ tail -5 words >> extrait
2
3 ~/UNIX$ cat extrait
```

- Ajouter le mot FIN à la fin du fichier mots avec la commande **cat**.

```
1 ~/UNIX$ cat << EOF >> mots
2 FIN
3 EOF
4 ~/UNIX$ tail mots
```

- Que se passe-t'il sur la liste words ?

```
1 Elles est également modifiée car le fichier mots est un alias de words.
```

Éditer des fichiers

- Créer avec la commande **cat** un fichier nommé file.txt contenant les lignes

Ce fichier

contient 2 lignes.

```
1 ~/UNIX$ cat > file.txt
2 Ce fichier
3 contient 2 lignes.
4 ^D
5 ~/UNIX$ cat file.txt
```

- Créer un autre fichier nommé fileNum.txt contenant les mêmes lignes mais numérotées.

```
1 ~/UNIX$ cat -n > fileNum.txt < file.txt
2 ~/UNIX$ cat fileNum.txt
```

- Éditer le fichier file.txt avec **vi**, **nano** et **gedit**. Écrire le nom de l'éditeur utilisé, sauver et quitter l'éditeur.

Trouver un fichier

- Trouver un fichier via son nom

```
1 ~/UNIX$ locate bashrc
```

- Trouver avec la commande **find**, rechercher les fichiers de type .pdf de votre répertoire Documents

```
1 ~/UNIX$ find ~/Documents/. -name '*.pdf'
```

- Recommencez en affichant les droits sur ces fichiers et sa taille en format M, K etc.

```
1 ~/UNIX$ find ~/Documents/. -name '*.pdf' -exec ls -lsh {} \;
```

- Recommencez en affichant les droits sur ces fichiers et sa taille en format M, K etc. et en les triant du plus petit au plus gros.

```
1 ~/UNIX$ find ~/Documents/. -name '*.pdf' -exec ls -lsh {} \; |sort -n
```

- Recommencez en affichant les droits sur ces fichiers et sa taille en format M, K etc. et en affichant les 10 plus gros par ordre croissant

```
1 ~/UNIX$ find ~/Documents/. -name '*.pdf' -exec ls -lsh {} \; |sort -n |  
tail -n 10
```

- Trouver tous les fichiers de ce répertoire de taille plus grande que 500M et les afficher par taille croissante

```
1 ~/UNIX$ find ~/Documents/ -size +500M -exec du -sh {} \; |sort -nr
```