Répertoires et chemins

- Objet : Répertoire et chemins en détail
- Niveau requis :
  débutant, avisé
- Commentaires : *Introduction pour l'utilisation des* lignes de commande *par l'exemple.*
- Débutant, à savoir : Utiliser GNU/Linux en ligne de commande, tout commence là !
- Suivi :
  - Création par smolski le 08/11/2010
  - Testé par milou le 18/12/2015
- Commentaires sur le forum : C'est ici[1)]

Partons maintenant à la découverte des chemins absolus et relatifs...



Pour les exemples, nous allons utiliser pour le nom de la machine : `machine` et pour le nom user : `toto`, pour exécuter les commandes en exemple, transposez ces noms avec ceux que vous utilisez.

Préparation du terrain

Dans votre */home/toto*,créez un nouveau dossier avec la commande mkdir, puis placez-vous dans ce nouveau dossier avec la commmande cd.

Exemple pratique :

```
mkdir -p tmp/df_tp_chemins_01/
```

```
cd tmp/df_tp_chemins_01/
```



L'invite de commande devient

```
toto@machine:~/tmp/df_tp_chemins_01$
```

Ce qui signifie que nous sommes placés maintenant dans le répertoire créé avec mkdir à l'intérieur du répertoire `/home/toto`

Avec la commande touch, on va maintenant créer dans ce nouveau répertoire les fichiers qui nous serviront pour la suite.

Exemple des commandes de la création des fichiers :

```
touch phaute_dorthografe
```

```
touch notes_temporaires
```

```
touch quisuisje
```

Le répertoire courant, ou répertoire de travail



Nous utilisons le nom `toto` pour indiquer l'*user* des exemples et `machine` pour le nom donné à la machine, transposez *toto* avec vos propres nom user et *machine* pour le nom de votre machine pour réaliser chez les exemples qui suivent.

Le répertoire courant, ou répertoire de travail est ce qui est indiqué juste à gauche du $ dans le prompt[2] bash ainsi :

```
~/tmp/df_tp_chemins_01$
```

Vous pouvez l'afficher en tapant la commande pwd :

```
toto@machine:~/tmp/df_tp_chemins_01$ pwd
```

retour de la commande

```
/home/toto/tmp/df_tp_chemins_01
```

On change de répertoire de travail avec la commande CD.



Le répertoire de travail est également stocké dans la variable $PWD du shell.

La commande echo à savoir pour la suite.

```
toto@machine:~/tmp/df_tp_chemins_01$ echo $PWD
```

retour de la commande

```
/home/toto/tmp/df_tp_chemins_01
```

Chemin relatif / chemin absolu



Pour identifier de manière unique un fichier sur un ordinateur sous Linux, on donne ce qu'on appelle un chemin vers le fichier.

Ce chemin peut être :

- *absolu* ou
- *relatif*

Quelques chemins particuliers

- `..` (deux points) désigne le répertoire parent d'un répertoire considéré
- `.` (un point) désigne le répertoire considéré

Exemples :

- `/home/../var` est un chemin pour le répertoire `/var`
- `ls .` liste le répertoire courant
- `../../../.` est un chemin vers le dossier parent, équivalent à `..`

Chemin absolu



Un **chemin absolu** est un chemin depuis la racine du système de fichier, `/`

Exemples de chemin *absolu* :

- `/var/cache/apt/archives/` qui est le dossier contenant les paquets téléchargés pour installation ou mise à jour par `aptitude`, ou `apt-get`, ou `synaptic` etc.
- `/home/../var` est aussi un chemin absolu puisqu'il donne le chemin depuis `/`

Chemin relatif



Un **chemin relatif** est un chemin depuis le **répertoire de travail** : `$PWD`

Exemple de chemin *relatif* :

- `../../` est un chemin relatif vers le dossier parent du dossier parent.

Si vous êtes toujours dans `~/tmp/df_tp_chemins_01/`, la commande

```
ls ../..
```

doit vous lister votre *home*.

- `./plouf/../tchac/` est un chemin relatif vers le dossier `$PWD/tchac`



`Attention !` : Virtuellement, un chemin relatif commence **toujours** par :`./`.



Cependant, en règle générale, on peut l'omettre.

Exemple

```
ls -lh quisuisje
```

Signifie en fait :

```
ls -lh ./quisuisje
```

Exemple pour renommer des fichiers

Lorsque l'on veut renommer un fichier, on utilise la commande mv qui prend deux arguments :

1. D'abord un **chemin** vers le fichier à déplacer,
2. puis le nouveau **chemin** du fichier.

Ce chemin peut être :

1. soit *absolu*
2. soit *relatif*

Dans la construction des fichiers d'exemple, nous pouvons exécuter la commande mv :

Soit avec deux chemins relatifs :

```
toto@machine:~/tmp/df_tp_chemins_01/$ mv ./phaute_dorthografe
./faute_dorthographe
```

qui signifie donc :

```
mv $PWD/./phaute_dorthografe $PWD/./faute_dorthographe
```

Soit avec un chemin relatif et un chemin absolu

```
toto@machine:~/tmp/df_tp_chemins_01/$ mv ./notes_temporaires
/tmp/notes_temporaires.txt
```

qui signifie :

```
toto@machine:~/tmp/df_tp_chemins_01/$ mv $PWD/./notes_temporaires
/tmp/notes_temporaires.txt
```

Effacer le bazar

Pour enlever les fichiers du tuto :

```
cd ~/tmp
```

```
toto@machine:~/tmp$ rm -r df_tp_chemins_01
```

```
toto@machine:~/tmp$ cd
```

L'invite de commande redevient

```
toto@machine:~$
```

Liens et remerciements

Allez donc voir ici :

- Les outils d'exécution exec et execdir de la commande find

Vous n'en reviendrez pas comme ça devient maintenant facile ! 😀

Un gros merci au **captnfab** *qui nous guide et nous permet de reproduire ici le contenu de sa substantifique moëlle, ce que faire se peut.* 😉

1)
N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !
2)
terminal

From:
http://debian-facile.org/ - **Documentation - Wiki**

Permanent link:
http://debian-facile.org/doc:programmation:shell:repertoires

Last update: **29/12/2019 08:49**