

Les variables d'environnement sous Linux

Table of Contents

- [Les variables d'environnement sous Linux](#)
- [Les principales variables](#)
 - [HOME](#)
 - [PATH](#)
 - [PWD](#)
 - [PS1](#)
 - [PS2](#)
 - [TERM](#)
 - [LOGNAME](#)
- [Manipulation des variables d'environnement](#)
 - [Consultation des variables d'environnement](#)
 - [Création d'une variable d'environnement](#)
 - [Suppression d'une variable d'environnement](#)
- [Les fichiers d'environnement](#)
 - [En sh/ksh](#)
 - [En bash](#)

Une variable d'environnement est une variable dynamique utilisée par des processus ou des applications afin de définir des chemins ou des raccourcis d'information.

Les variables d'environnement sont dynamiques. C'est à dire qu'elles peuvent chacune changer en fonction de divers paramètres que l'utilisateur demande, l'endroit où l'on se trouve sur le serveur, le langage qu'on utilise ...

Les principales variables

HOME

Cette variable contient le chemin du répertoire d'accueil de l'utilisateur.

PATH

Cette variable contient une liste de répertoire dans lesquels le shell recherche toutes les commandes qui sont exécutées. Si une commande est exécutée et qu'elle ne se trouve dans aucun des répertoires indiqués dans la variable PATH, une erreur sera retournée en indiquant que la commande est introuvable. Si l'on souhaite exécuter une commande qui se trouve dans un répertoire non indiqué

dans la variable PATH, il suffit tout simplement de modifier le contenu de la variable.

```
$ PATH=$PATH:/leRepertoireDeMaCommande
```

Pour modifier la variable PATH définitivement, il suffit d'ajouter une ligne au fichier ~/.bashrc :

```
> export PATH=$PATH:/mon_autre_repertoire
```

Ou modifier directement le fichier /etc/environment :

```
> PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/mon_autre_repertoire"
```

Pour exécuter une commande présente dans le répertoire courant, il suffit d'ajouter à la variable PATH la valeur “.” ou “:.”.

PWD

Cette variable contient le chemin du répertoire courant.

PS1

Cette variable contient la chaîne de caractères correspondant au prompt (\$ en règle générale).

Pour faire apparaître la valeur du répertoire courant dans le prompt :

```
$PS1=' $PWD$ '
```

ou

```
$ PS1='\w$ '
```

Le tableau ci-dessous présente les séquences d'échappement permettant de paramétrer la variable PS1

Séquence échappement	Valeur
\u	Nom de l'utilisateur
\h	Nom de la machine
\w	Répertoire courant
\W	Partie terminale du répertoire courant

PS2

Cette variable contient la chaîne de caractères correspondant au prompt secondaire (> en règle générale).

```
$ echo 'ceci est le début
> de ma chaîne de texte
> et tant que je ne saisi pas
> le caractère apostrophe
> ça continue
> '
ceci est le début
de ma chaîne de texte
et tant que je ne saisi pas
le caractère apostrophe
ça continue
$
```

TERM

Cette variable contient le type du terminal de l'utilisateur.

LOGNAME

Cette variable contient le nom de l'utilisateur connecté.

Manipulation des variables d'environnement

Dans cette section, nous allons apprendre à voir, définir et supprimer des variables d'environnement sous Linux.

Consultation des variables d'environnement

La commande `set` permet d'obtenir la liste des variables pour le shell courant..

```
$ set
```

Le soucis avec “set” est que la commande liste aussi les fonctions, la commande `env` permet d'obtenir une sortie plus classique.

```
$ env
BASH=/bin/bash
HOME=/home/toto
HOSTNAME=myhostname
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
PS1='\[\e]0;\u@h:
\w\a\]${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@h$
PS2='> '
PS4='+ '
SHELL=/bin/bash
...
```

Le caractère \$ permet d'obtenir la valeur d'une variable :

```
$ echo $BASH
/bin/bash
```

On peut également filtrer la sortie de la commande set

```
$ set | grep EDITOR
EDITOR=/usr/bin/nano
```

ou de la commande env

```
$ env | grep EDITOR
EDITOR=/usr/bin/nano
```

ou env

Création d'une variable d'environnement

On peut facilement créer une variable d'environnement pour le shell en cours avec la commande suivante :

```
$ export DIR_APPLI=/etc/appli
```

On peut vérifier ensuite que la variable est créée :

```
$ echo $DIR_APPLI
```

Cela devrait renvoyer selon l'exemple “/etc/appli”. On pourra alors utiliser cette variable dans un fichier de configuration (par exemple). Il est important de savoir que la création d'une variable d'environnement n'est valable que pour la session en cours. Pour rendre cette création permanente, il faut saisir cette même commande dans le fichier “/home/\$USER/.bashrc” de l'utilisateur en cours (si

l'on veut que la variable ne soit que pour l'utilisateur) ou dans le fichier `"/etc/bash.bashrc"`, pour tous les utilisateurs.

Suppression d'une variable d'environnement

La suppression d'une variable peut se faire avec la commande `"unset"` qui est faite pour effacer complètement une variable d'environnement (pour la session en cours !) :

```
$ unset $VAR
```

On pourra ensuite vérifier que la variable à été correctement supprimée avec la commande d'affichage des variables d'environnement :

```
$ env
```

Si la variable à juste été définie pour une session, elle sera effacée au prochain redémarrage.

Les fichiers d'environnement

Les fichiers d'environnement sont utilisés pour stocker de manière permanente tout le paramétrage de l'environnement de l'utilisateur.

Ce sont des scripts shell qui contiennent un ensemble de commandes Linux.

Certains scripts sont exécutés uniquement par le shell de connexion et d'autres par un shell ordinaire.

Le shell de connexion est lancé immédiatement après l'identification de l'utilisateur.

Le shell de connexion exécute en premier le script `/etc/profile`. Ce script contient le paramétrage commun à tous les utilisateurs.

Il recherche ensuite dans le répertoire d'accueil de l'utilisateur un script dont le nom dépend du shell utilisé.

En sh/ksh

Le script de connexion se nomme `~/.profile`

En bash

L'un des 3 scripts suivants dans l'ordre :

```
~/.bash_profile
```

```
~/.bash_login  
~/.profile
```

En bash, le shell de connexion exécute en premier le script `/etc/profile`, puis le script `~/.profile` et enfin le script `~/.bashrc`.

C'est dans ce dernier fichier, `~/.bashrc`, que l'on peut définir de nouveaux alias, exporter automatiquement de nouvelles variables, paramétrer l'auto-complétion, définir les variables `PS1` et `PS2`, paramétrer le nombre de commandes à historiser, etc etc.

Par exemple, en BASH, pour utiliser le programme nano (éditeur de texte) comme éditeur par défaut du programme mutt (client mail en ligne de commande), il suffit d'inscrire dans son fichier `~/.bashrc` :

```
export EDITOR=/usr/bin/nano
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=debian:linux-variables-environnement>

Last update: **2025/02/19 10:59**

