

Variables - Droits

Cours 3

E. Paviot-Adet

Emmanuel.Paviot-Adet@parisdescartes.fr
Bureau B2-2

IUT Paris Descartes
DUT Informatique - 1^{ère} année

Sommaire

- 1 Variables
 - Généralités
 - Manipulation
- 2 Modification des droits
- 3 Commandes

Plan

- 1 Variables
 - Généralités
 - Manipulation
- 2 Modification des droits
- 3 Commandes

Problématique

Ce que doit connaître un processus

- Le système sur lequel il s'exécute
- La machine sur laquelle il s'exécute
- Le détail de l'installation des logiciels
 - Où trouver des binaires, bibliothèques...
- L'utilisateur
- ...

Ce que doit connaître le système

- Où se trouvent les exécutables, les bibliothèques...
- L'utilisateur
- ...

Problématique

Ce que doit connaître un processus

- Le système sur lequel il s'exécute
- La machine sur laquelle il s'exécute
- Le détail de l'installation des logiciels
 - Où trouver des binaires, bibliothèques...
- L'utilisateur
- ...

Ce que doit connaître le système

- Où se trouvent les exécutables, les bibliothèques...
- L'utilisateur
- ...

Solution

Caractéristiques du système de communication

- Accessible par l'utilisateur et les processus
- Facilement consultable/modifiable
- Utilisable dans les commandes de l'interface textuelle

Variables

- Ne contiennent que des chaînes de caractères
- Certaines sont gérées directement par le système
 - Nom du système, caractéristiques de la machines...
- D'autres sont utilisées par le système
 - Emplacement des exécutables,...
- D'autres sont spécifiques à certains programmes

Solution

Caractéristiques du système de communication

- Accessible par l'utilisateur et les processus
- Facilement consultable/modifiable
- Utilisable dans les commandes de l'interface textuelle

Variables

- Ne contiennent que des chaînes de caractères
- Certaines sont gérées directement par le système
 - Nom du système, caractéristiques de la machines...
- D'autres sont utilisées par le système
 - Emplacement des exécutable,...
- D'autres sont spécifiques à certains programmes

Deux types de variables

Variables d'environnement

- Transmises aux fils
 - Un processus (*père*) en crée un autre (*fils*)
- Nom généralement en majuscule
- Créées par le système ou dans un fichier de configuration
 - Possibilité d'en créer à tout moment

Variables locales à un shell

- Non transmises aux fils
- Nom généralement en minuscules
- Proches des variables locales en programmation

Deux types de variables

Variables d'environnement

- Transmises aux fils
 - Un processus (*père*) en crée un autre (*fils*)
- Nom généralement en majuscule
- Créées par le système ou dans un fichier de configuration
 - Possibilité d'en créer à tout moment

Variables locales à un shell

- Non transmises aux fils
- Nom généralement en minuscules
- Proches des variables locales en programmation

Manipulation des variables

Accès au contenu

- Un $\$$ doit précéder la variable dans la ligne de commande
 - Indique qu'il s'agit d'une variable
 - A utiliser pour les variables locales et d'environnement

Définition/modification d'une variable locale

- Forme générale : *variable='valeur'*
 - Pas d'espace autour de l'égalité
 - `test='Ceci est un exemple'`
- Cas particulier : *variable=valeur*
 - *valeur* ne contient pas d'espace
 - `test=monNom`

Manipulation des variables

Accès au contenu

- Un $\$$ doit précéder la variable dans la ligne de commande
 - Indique qu'il s'agit d'une variable
 - A utiliser pour les variables locales et d'environnement

Définition/modification d'une variable locale

- Forme générale : *variable*='valeur'
 - Pas d'espace autour de l'égalité
 - test='Ceci est un exemple'
- Cas particulier : *variable*=valeur
 - *valeur* ne contient pas d'espace
 - test=monNom

Commande de base - export

Commande *export* : Transforme une variable locale en une variable d'environnement

Syntaxe simplifiée

```
export [variable[=valeur]]
```

En pratique

- Notes
 - Sans argument *export* affiche la liste des variables exportées
 - Toujours pas d'espace autour de =
- Exemple : `export PATH=$PATH:~/bin`

Commande de base - unset

Commande *unset* : Suppression d'une variable

Syntaxe simplifiée

unset *variable*

Exemple

- *mavar=test*
 - Création de la variable *mavar* avec la valeur *test*
- *unset mavar*
 - La variable *mavar* n'existe plus

Commande de base - set

Commande *set* : Affichage des variables locales et d'environnement

Syntaxe simplifiée

set

Note

- Permet de positionner, ou non, les attributs du shell
 - *set -x* active le mode trace
 - *set +x* annule le mode trace

Commande de base - env

Commande *env* : Affichage des variables d'environnement
(signification : ENVironment)

Syntaxe simplifiée

`env`

Note

- Les variables locales ne sont pas affichées

Exemples de création

- *VARIABLE=test*
 - Création d'une variable locale
- *set*
 - *VARIABLE* apparaît dans la liste
- *env*
 - *VARIABLE* n'apparaît pas dans la liste
- *export VARIABLE*
 - *VARIABLE* est maintenant une variable d'environnement
- *env*
 - *VARIABLE* apparaît dans la liste
- *export VARIABLE2=test2*
 - Création d'une variable d'environnement

Exemples d'exportations

- *VARIABLE=test*
 - Création d'une variable locale
- *bash*
 - On utilise un nouveau shell (création d'un processus fils)
- *set*
 - *VARIABLE* n'apparaît pas dans la liste (nouveau shell)
- *exit*
 - On quitte le nouveau shell
- *set*
 - *VARIABLE* apparaît dans la liste
- *export VARIABLE*
- *bash*
- *set*
 - *VARIABLE* apparaît dans la liste

Quelques variables d'environnement

PATH Liste des répertoires que le shell explore pour trouver les commandes externes

- Répertoires séparés par ':'
- Exemple : `PATH=/bin:/usr/bin`

USER Nom de l'utilisateur

SHELL Type de shell utilisé

MANPATH Liste des répertoires que la commande *man* explore pour trouver les documentations

- Répertoires séparés par ':'

HOME Répertoire domicile

Plan

- 1 Variables
 - Généralités
 - Manipulation
- 2 Modification des droits
- 3 Commandes

Commande de base - chmod

Commande *chmod* : Modification des autorisations des fichiers et répertoires (signification : CHange MODe)

Syntaxe simplifiée

`chmod [-R] [-v] mode nom`

- `-R` : modifications récursives
- `-v` : indique les modifications effectuées (verbose)

Notes

- mode : description des nouveaux droits
 - Forme symbolique ou numérique
 - Sous forme symbolique
 - mode peut être remplacé par mode [,mode]....
 - Pas d'espace après la virgule
- Seul le propriétaire peut utiliser cette commande

Forme symbolique

[utilisateur]modificateur[droits]

- utilisateur n'importe quelle combinaison de :
 - a All
 - u User (propriétaire)
 - g Group
 - o Others
- modificateur un parmi :
 - + Ajout de droits
 - Suppression de droits
 - = Spécification de droits
- droits n'importe qu'elle combinaison de :
 - r Read (lecture)
 - w Write (écriture)
 - x eXecute (exécution)

Forme symbolique - exemples

- `chmod a+x monexec` Ajoute le droit en exécution au fichier *monexec* à tous les types d'utilisateurs
- `chmod go-rw fic.txt` Retire les droits en lecture et écriture sur le fichier *fic.txt* aux membres de groupe et aux autres utilisateurs
- `chmod u=rw fic1.txt` Le propriétaire a les droits en lecture et écriture (et pas les droits en exécution) sur le fichier *fic1.txt* après exécution de la commande
- `chmod u+x,go=rx fic3.txt` Ajout des droits en exécution pour le propriétaire, le groupe et les autres n'ont que le droit en lecture et en exécution (pas en écriture)

Forme symbolique - Comportement par défaut

utilisateur n'est pas spécifié

- Restauration, pour tous, des droits spécifiés par défaut pour +
et =

```
chmod a-rwx fic - - - - -
```

```
chmod =rw fic -rw-r- -r- -
```

- Suppression, pour tous, des droits pour -

droits ne sont pas spécifiés

Aucun droit

```
chmod a= fic - - - - -
```

```
chmod = fic - - - - -
```

Forme symbolique - Comportement par défaut

utilisateur n'est pas spécifié

- Restauration, pour tous, des droits spécifiés par défaut pour +
et =

```
chmod a-rwx fic - - - - -
```

```
chmod =rw fic -rw-r- -r- -
```

- Suppression, pour tous, des droits pour -

droits ne sont pas spécifiés

Aucun droit

```
chmod a= fic - - - - -
```

```
chmod = fic - - - - -
```

Forme numérique

[[[d_0] d_u] d_g] d_o

- d_0 n'est pas détaillé ici
- S'il n'y a pas 4 chiffres, le système complète à GAUCHE avec des 0
- $d_u d_g d_o$: Droits pour le propriétaire, les membres du groupe et les autres respectivement
- Chaque droit est représenté en base 8 :
 - 1 bit pour chacun des droits rwx
 - 3 bits, donc 8 valeurs possibles
 - 4₈ : droit en lecture (car 100 en binaire)
 - 2₈ : droit en écriture (car 010 en binaire)
 - 1₈ : droit en exécution (car 001 en binaire)
 - Si plusieurs droits sont spécifiés, faire la somme des valeurs

Forme numérique - exemples

`chmod 111 monexec` Uniquement les droits en exécution pour tout le monde

- Equivalent à : `chmod 0111 monexec`
- Equivalent à : `chmod a=x monexec`

`chmod 64 fic.txt` Plus de droits pour le propriétaire, droits *rw* pour le groupe, droit *r* pour les autres

- Equivalent à : `chmod 0064 monexec`
- Equivalent à : `chmod u=,g=rw,o=r monexec`

Commande de base - chown

Commande *chown* : Changement de propriétaire (signification : CHange OWNer)

Syntaxe simplifiée

chown [-R] nouveau propriétaire nom

- -R : modifications récursives

En pratique

- Exemple : *chown darche fic.txt*
 - *darche* sera le nouveau propriétaire de *fic.txt*
- Note : seul le propriétaire du fichier peut exécuter la commande

Commande de base - chgrp

Commande *chgrp* : Changement de groupe (signification : CHange GRouP)

Syntaxe simplifiée

chown [-R] nouveau_groupe nom

- -R : modifications récursives

En pratique

- Exemple : *chgrp wheel fic.txt*
 - *wheel* sera le nouveau groupe de *fic.txt*
- Note : seul un membre du nouveau groupe peut exécuter la commande

Plan

- 1 Variables
 - Généralités
 - Manipulation
- 2 Modification des droits
- 3 Commandes

Commande de base - echo

Commande *echo* : Afficher une ligne de texte

Syntaxe simplifiée

`echo [-n] [chaîne de caractères]`

- `-n` : pas de saut de ligne final

En pratique

- Exemple : *echo ceci est un exemple*
- Note : aussi utilisé pour afficher le contenu des variables

Commande de base - file

Commande *file* : Détermination du type d'un fichier

Syntaxe simplifiée

file [-z] fichier_ou_répertoire...

- -z : essaie de regarder à l'intérieur des fichiers compressés

En pratique

- Exemple : *file bin/bash*

Commande de base - find

Commande *find* : Recherche de fichiers dans une hiérarchie de répertoires

Syntaxe simplifiée

find [-L] chemin -name nom

- -L : suivre les liens symboliques
 - Par défaut ils ne sont pas suivis

En pratique

- Exemple : *find . -name "*.txt"*
 - Les guillemets sont indispensables pour éviter l'expansion avant l'exécution de la commande

Commande de base - alias

Commande *alias* : Créer un synonyme

Syntaxe simplifiée

```
alias synonyme='commande_équivalente'
```

En pratique

- Exemple : *alias ll='ls - l'*
 - La commande *ll* est ensuite reconnue
- Notes :
 - Surtout utilisé dans les fichiers de configuration type *.bashrc*
 - Si un alias remplace une commande existante
 - 'commande' pour utiliser la commande d'origine
 - Pour supprimer un alias : la commande *unalias*