

Introduction aux Scripts Shell

Soumis par VieuxProf

01-01-1999

Dernière mise à jour : 02-02-2008

- Un script est un fichier exécutable par le shell qui contient des commandes externes UNIX, des commandes internes du shell et des commentaires. Ce n'est pas un binaire mais le fichier source ASCII qui sera interprété par le shell. Il est donc portable.

- Le script se comporte comme l'entrée standard du shell qui l'exécute. Si le shell détecte une erreur, il arrête la lecture et affiche le numéro de la ligne qui a provoqué l'erreur. Le shell père exécute le script dans un shell fils qui hérite de l'environnement du shell courant mais n'a pas d'influence sur l'environnement du shell père.

- La structure d'un script est généralement la suivante :

- signal du démarrage (echo commence ici)

- réalisation de la tâche

- suppression des fichiers temporaires

- signal de fin (echo finit ici)

- Le caractère # permet d'insérer des commentaires. Le caractère #! en début de script spécifie le shell à utiliser (ex : #!/bin/sh).

- Un script peut être exécuté par l'une des procédures suivantes :

\$ /bin/sh monscript (lance monscript dans un shell donné, droits r suffisants)

\$ sh < monscript (redirige l'entrée standard vers monscript)

\$ chmod u+x monscript; ./monscript (lance monscript comme une commande à partir du répertoire courant, droits x nécessaires)

\$ chmod u+x monscript; monscript (lance monscript comme une commande si variable PATH correctement définie)

\$. monscript (lance monscript à partir du shell courant).

{mospagebreak title=Mise au point des scripts}

- Le shell réalise les opérations dans l'ordre suivant :

- décomposition des lignes en mots (parsing)

- remplacement des alias

- remplacement des variables par leur valeur

- remplacement des commandes par leur résultat

- expansion des noms de fichiers (jokers)

- L'activation ou désactivation du mode trace est réalisée par la commande interne 'set' pour les commandes générales.

Pour les scripts, le mode trace s'active par la commande 'set' ou en indiquant les options sur la ligne de commandes au lancement du shell fils.

set -x/set +x

active/désactive l'affichage des étapes de substitution avant exécution

set -v/set +v

active/désactive l'affichage de la ligne telle qu'elle est dans le script avant exécution

set -xv

affiche la commande telle qu'elle est écrite dans le script, le résultat de l'interprétation et le résultat de l'exécution

set -n/set +n

active/désactive la vérification de syntaxe

set -e

active/désactive l'arrêt d'exécution du script en cas d'erreur (code retour différent de 0)

set -

désactive les options x et v

- La commande 'echo' affiche des messages sur la sortie standard (voir section "Commandes"). C'est une commande interne du shell qui possède 2 implémentations en commandes externes (/bin/echo, /bin/posix/echo). Elle est très utile pour le débogage.

- La commande 'print' (ksh) affiche également des messages sur la sortie standard mais est une commande interne au shell sans implémentation externe.

{mospagebreak title=Options du shell}

- Le comportement du shell peut être modifié par des options activées ou désactivées par la commande interne 'set' ou en indiquant les options au lancement du shell fils :

sh -o option monscript/sh +o option monscript
lance monscript en activant/désactivant l'option 'option'

set -o
affiche l'état des options (on/off)

set -o option/set +o option
active/désactive l'option 'option' dans le shell courant

- Les principales options sont :

ignoreeof
interdit/autorise la déconnexion par ctrl-D

noclobber
interdit/autorise l'écrasement d'un fichier avec >

vi, emacs
active/désactive l'édition de commandes en mode vi ou emacs

nounset
génère une erreur quand on utilise une variable non définie

noglob
désactive/active l'interprétation des jokers