

Travaux pratiques : Écriture de scripts de base sous Windows et Linux

Objectifs

Au cours de ces travaux pratiques, vous allez écrire des scripts de base dans différents langages de script afin de comprendre comment chaque langage gère les tâches d'automatisation.

Contexte/scénario

L'écriture de scripts pour automatiser les fonctions d'administration courantes permet de gagner du temps et offre à l'administrateur la flexibilité nécessaire pour effectuer d'autres tâches. Au cours de ces travaux pratiques, vous allez écrire trois types de scripts qui effectuent des tâches similaires. Comparez les différents langages à mesure que vous automatisez une tâche simple.

Ressources requises

- PC Window
- Une machine virtuelle exécutant une distribution Linux

Instructions

Étape 1 : Créez un script de commandes Windows.

- Dans un éditeur de texte tel que le Bloc-notes, enregistrez un fichier texte nommé **info.bat** dans votre répertoire de base (C:\Utilisateurs*votre_nom_d'utilisateur*) avec le texte suivant :

```
@echo off
echo Computer Name is: %computername%
echo Windows version is:
ver
echo CPU is: %PROCESSOR_IDENTIFIER%
echo Total Memory is:
rem Windows Management Instrumentation Command (WMIC) is a command line
utility that can retrieve information about local or remote computers. For
more inline information, enter help wmic or wmic /? at the command prompt.
wmic ComputerSystem get TotalPhysicalMemory
echo The disks that are installed and their freespace:
wmic logicaldisk get size,freespace,caption
echo All the %computername% IP addresses
rem netsh is a command line scripting utility that allows the users to view
or modify the network configurations of a running computer. For more inline
information, enter nesh /? at the command prompt.
rem findstr is used for searching for a text string in files. For more inline
information, enter findstr /? at the command prompt.
netsh interface ip show address | findstr "IP Address"
```

- b. Ouvrez une invite de commandes et accédez à votre répertoire de base.
- c. Répertoriez le contenu de votre répertoire de base et vérifiez que le fichier **info.bat** est enregistré avec le fichier approprié. Si ce n'est pas le cas, renommez le fichier, par exemple **renommez info.bat.txt info.bat**.
- d. À l'invite, saisissez **info.bat** pour exécuter le script.

Questions :

Quel résultat obtenez-vous ?

Saisissez vos réponses ici

Quels sont les noms (%name%) utilisés dans le script ?

Saisissez vos réponses ici

Identifiez les commandes suivantes dans le script :

echo :

Saisissez votre réponse ici.

findstr :

Saisissez votre réponse ici.

netsh :

Saisissez votre réponse ici.

ver :

Saisissez votre réponse ici.

wmic :

Saisissez votre réponse ici.

Étape 2 : Créez un script PowerShell ISE.

- a. Cliquez sur **Démarrer**, recherchez **PowerShell ISE**, cliquez avec le bouton droit sur la sélection, puis cliquez sur **exécuter en tant qu'administrateur**.
- b. Vérifiez que vous êtes dans le répertoire de base : PS C:\Utilisateurs*votre_nom_d'utilisateur*.
- c. Pour autoriser le script à s'exécuter, saisissez **Set-ExecutionPolicy RemoteSigned** à l'invite. Cliquez sur **Oui** pour autoriser l'exécution du script. Les paramètres peuvent être redéfinis sur **Non** une fois le script terminé.
PS C:\Utilisateurs*votre_nom_d'utilisateur*> **Set-ExecutionPolicy RemoteSigned**
- d. Sélectionnez **Fichier -> Créer** et créez un nouveau script.
- e. Saisissez le texte suivant dans la fenêtre **Untitled.ps1** et enregistrez-le sous le nom **info.ps1** dans votre répertoire de base.

```
Write-Output "Computer name is:"  
get-content env:computername  
Write-Output "Windows version is:"  
(Get-WmiObject -class Win32_OperatingSystem).Légende  
Write-Output "CPU is:"  
Get-WmiObject Win32_Processor | findstr "Name"  
Write-Output "Total Memory is:"  
[Math]::Round((Get-WmiObject -Class win32_computersystem -ComputerName  
localhost).TotalPhysicalMemory/1Gb)
```

```
Write-Output "The Disks that are installed and their freespace:"  
Get-WmiObject -Class Win32_logicaldisk -Filter "DriveType = '3'"  
Write-Output "IPv4 addresses"  
Get-NetIPAddress -AddressFamily IPv4 | Sort-Object -Property InterfaceIndex |  
Format-Table
```

Remarque : la commande **Get-NetIPAddress** n'est pas disponible sous Windows 7.

Remarque : dans PowerShell ISE, vous pouvez appuyer sur F1 ou sélectionner **Aide > Windows PowerShell ISE** pour obtenir plus d'informations.

- f. Pour afficher les fonctions de chaque commande, cliquez sur **Modules complémentaires**, puis vérifiez que l'option **Commande** est sélectionnée. Dans l'onglet Commande, saisissez le nom de la commande dans le champ **Nom**. Sélectionnez la commande souhaitée, puis cliquez sur ? pour en savoir plus sur la commande souhaitée.

Sous Windows 7, cliquez sur **Aide** et sélectionnez **Aide de Windows PowerShell**. Sélectionnez **Rubriques d'aide des applets de commande Windows PowerShell**. Recherchez la commande souhaitée.

- g. Saisissez **.info.ps1** à l'invite PS. **Remarque :** assurez-vous d'utiliser la barre oblique correcte.

```
PS C:\Utilisateurs\votre_nom_d'utilisateur> .\info.ps1
```

Question :

Quel est le résultat du script ?

Saisissez votre réponse ici.

- h. Comparez les deux scripts. Associez la commande batch aux commandes PowerShell ci-dessous :

Commande batch Windows	Commande PowerShell
echo Computer Name is: %computername%	
echo Windows version is: ver	
echo CPU is: %PROCESSOR_IDENTIFIER%	
echo Total Memory is:	
wmic ComputerSystem get TotalPhysicalMemory	
echo The disks that are installed and their freespace:	
wmic logicaldisk get size,freespace,caption	
echo All the %computername% IP addresses	
netsh interface ip show address findstr "IP Address"	

Étape 3 : Créez un script BASH.

Un éditeur de texte permet de créer un script exécutable. L'un des outils de l'éditeur de texte, vi, ou la version vi améliorée (vim), utilise des commandes basées sur des lettres et sur des chiffres pour modifier le texte. Par exemple, **dd** supprime toute la ligne sur laquelle le curseur est placé. **5dd** supprimerait 5 lignes. Lorsque vi est en mode de commande, la saisie est interprétée comme une commande.

Pour passer en mode d'insertion à l'emplacement actuel du curseur, tapez **i**. Pour ajouter du texte à la fin de la ligne en cours, tapez **a**. Pour insérer le texte sur une nouvelle ligne en dessous de la ligne actuelle, tapez **o**. Utilisez la touche Échap pour quitter le mode d'insertion en mode commande.

Pour enregistrer un fichier dans l'éditeur vi, utilisez le mode de commande **:w**. Pour enregistrer et quitter, tapez **:wq**. Pour quitter sans enregistrer, tapez **:q!**.

Selon votre version du système d'exploitation UNIX, vous disposerez peut-être d'un autre outil d'édition de texte, tel que nano, pico et gedit. Les outils d'édition de texte tels que vi, nano et pico sont accessibles via la ligne de commande, alors que les éditeurs de texte basés sur l'interface graphique, comme gedit, se situent normalement dans le menu de l'application ou dans la ligne de commande.

- a. Démarrez un ordinateur Linux ou une machine virtuelle.
- b. À l'aide d'un éditeur de texte, créez un fichier nommé **info.sh** dans votre répertoire de base avec le texte suivant :

```
#!/bin/bash
echo "Computer name is: " $HOSTNAME
echo "Operating System is:"
cat /etc/os-release | grep PRETTY_NAME
echo "CPU is"
lscpu | grep "Model name:" | sed -r 's/Model name:\s{1,} //g'
echo "Total Memory is"
cat /proc/meminfo | grep "MemTotal"
echo "The disks that are installed and their freespace"
df -h
echo "All the" $HOSTNAME "IP addresses"
hostname -I
```

- c. Ouvrez un terminal et accédez à votre répertoire de base. Pour rendre le script exécutable, saisissez **chmod 755 info.sh** à l'invite.
- d. À l'invite, saisissez **./info.sh** pour exécuter le script.

Questions :

Quel est le résultat du script ?

Saisissez votre réponse ici.

Que signifie « **#!/bin/bash** » au début du script ?

Saisissez votre réponse ici.

Quelle commande utiliseriez-vous pour en savoir plus sur les commandes **df** et **lscpu** ?

Saisissez votre réponse ici.