



Ingeniería en Sistemas Computacionales.

Intérpretes de Comandos.

junio 18

2019

Explicar un intérprete de comandos de un SO y sus requisitos y sentencias

Tarea # 3

Profesora: Ing. Stephanie Fallas Retana

Alumnos:

Fabio Sandí Sánchez

Curso: Sistemas Operativos

Descripción del trabajo.

1. El estudiante debe investigar sobre algún intérprete de comandos (Configuración, prerequisites, SO etc.) (Valor dado por cumplimiento de la tarea)
2. Según la investigación anterior el estudiante debe de configurar alguna sentencia, tarea o ejecución por medio de la utilización de dicho intérprete de comandos y exponerlo en clase.

Desarrollo.

Windows PowerShell

Windows PowerShell es una interfaz de consola (CLI) con posibilidad de escritura y unión de comandos por medio de instrucciones (scripts en inglés). Es mucho más rica e interactiva que sus predecesores. Esta interfaz de consola está diseñada para su uso por parte de administradores de sistemas, con el propósito de automatizar tareas o realizarlas de forma más controlada. Originalmente denominada como MONAD en 2003, su nombre oficial cambió al actual cuando fue lanzada al público el 25 de abril de 2006.¹ El 15 de agosto de 2016, Microsoft abrió el código de PowerShell en GitHub.² La versión 6 se ofrece con licencia MIT.³

Funciones

Powershell no solo permite interactuar con el sistema operativo, sino también con programas de Microsoft como SQL Server, Exchange o IIS. La principal utilidad de Powershell es permitir automatizar tareas administrativas al usuario.

El lenguaje de la consola incluye declaración de variables, variables especiales predefinidas, operadores matemáticos, incluyendo igualdades y desigualdades numéricas, manejo de vectores, comparación entre estos, operadores de asignación, vectores asociativos (*hashtables*), valores booleanos, ciclos y ruptura de los mismos, operadores de expansión para simplificación de ejecuciones complejas (creación de vectores por medio de llamados a procedimientos, creación dinámica de vectores, etc.); comentarios, operadores de comparación binaria, caracteres de escape, orden de ejecución, ciclos del tipo "foreach", creación de procedimientos y funciones, creación de filtros, estructuras condicionales complejas (if/then/else/elseif/switch), operador de invocación dinámica del contenido de variables (\$p = "MiProceso" --> &\$p ejecuta MiProceso), llamado a métodos de tipo "\$p.ToUpper()", acceso a propiedades de instancias de objetos, redirección de salida normal de consola a archivos, retorno de valores, manejo de cadenas de caracteres por medio de operadores, manejo de excepciones y conversión explícita de tipos.

Una de las principales funciones de Powershell es tratar de igualar al famoso lenguaje [Perl](#) de [UNIX](#). El cual está considerado versátil, potente y con facilidad para interactuar con el sistema operativo. Exchange server 2007 utiliza Powershell internamente. La tendencia es que todas las aplicaciones tengan su sección visual y una opción para ver el código generado en Powershell.

Características

La característica distintiva de PowerShell, es que es un intérprete de comandos orientado a objetos. La información de entrada y de salida en cada etapa del proceso (*cmdlet*, "comándulo") es un conjunto de instancias de objeto, a diferencia de lo que ocurre con los intérpretes de comandos tradicionales, que solo devuelven y reciben texto.

Alias de comandos

En Powershell se pueden crear alias al estilo de [Unix](#), es decir, renombrar o nombrar de diferentes maneras a los comandos.

Por ejemplo para mostrar directorios se usa dir, ls, gci. El programador puede apodar el comando como quiera. Esto facilita trabajar con el entorno de programación.

Utilizando una única sentencia, mediante Powershell se pueden realizar fácilmente las siguientes acciones:

- Monitorizar contadores
- Apagar o detener servicios
- Modificar registros del sistema

Desarrollador(es) Microsoft

Lanzamiento inicial 14 de noviembre de 2006

Última versión estable 6.2.0

28 de marzo de 2019 (2 meses y 20 días)

Última versión en pruebas v6.2.0 RC1

05 de marzo de 2019 (3 meses y 12 días)

Género Línea de comandos

Programado en C#

Sistema operativo Windows XP (última 2.0)

Windows Server 2003 (última 3.0)

Windows Vista

Windows Server 2008

Windows 7

Windows Server 2008 R2

Windows 8

Windows 8.1

Windows 10,

Cmdlet	Alias	Bash Equivalent	Description
<code>Get-ChildItem</code>	<code>gci</code>	<code>ls</code>	Enlista los directorios y archivos en la ubicación actual.
<code>Set-Location</code>	<code>sl</code>	<code>cd</code>	Cambia al directorio en la ruta de acceso dada. Si escribes <code>..</code> en lugar de una ruta te moverá hacia arriba un directorio.
<code>Push-Location</code>	<code>pushd</code>	<code>pushd</code>	Cambiar al directorio.
<code>Pop-Location</code>	<code>popd</code>	<code>popd</code>	Regresa al directorio previo despues de usar <code>pushd</code>
<code>New-Item</code>	<code>ni</code>	<code>(touch)</code>	Crea un nuevo ítem. De no utilizarse un parámetro, el ítem será un archivo por defecto. El uso de <code>mkdir</code> es una abreviatura para incluir el parámetro <code>-ItemType dir</code> .
<code>mkdir</code>	<code>none</code>	<code>mkdir</code>	Crea un nuevo directorio. (Ver <code>New-Item</code> .)
<code>Explorer</code>	<code>none</code>	<code>(open)</code>	Abre algo utilizando el Explorador de archivos (la GUI)
<code>Remove-Item</code>	<code>rm</code>	<code>rm</code>	Borra algo... ide manera permanente!
<code>Move-Item</code>	<code>mv</code>	<code>mv</code>	Mueve algo. Necesita dos argumentos. Primero un nombre de archivo (i.e. su ruta actual), luego la ruta de nueva nueva locación (incluido el nombre que debe tener ahí). Si no se cambia la ruta, puede

Cmdlet	Alias	Bash Equivalent	Description
			usarse para renombrar archivos.
Copy-Item	cp	cp	Copia un archivo en una nueva ubicación. Requiere los mismos argumentos que mover, pero mantiene el archivo original en su ubicación.
Write-Output	write	echo	Exporta lo que escribas. Utiliza la redirección para enviarlo a un archivo. La redirección con >> añadirá texto al archivo en lugar de sobrescribir el contenido.
Get-Content	gc	cat	Obtiene el contenido de un archivo y lo imprime en la pantalla. La adición del parámetro -TotalCount seguido de un número x sólo imprime las primeras x líneas. Añadiendo el parámetro -Tail seguido de un número x sólo imprime las x líneas finales.
Select-String	sls	(grep)	Busca contenido específico.
Measure-Object	measure	(wc)	Obtiene información estadística sobre un objeto. Utiliza Get-Content y dirige la salida a Measure-Object con los parámetros -line, -word y -character para obtener información sobre el recuento de líneas, palabras o caracteres.
>	none	>	Redirección. Pone la salida del comando a la izquierda de > en un archivo a la derecha de >.
	none		Canalizar. Toma la salida del comando a la izquierda y la usa como entrada para el comando a la derecha.

Cmdlet	Alias	Bash Equivalent	Description
<code>Get-Help</code>	<code>none</code>	<code>man</code>	Obtiene el archivo de ayuda de un <i>cmdlet</i> . La adición del parámetro <code>-online</code> abre la página de ayuda en TechNet.
<code>exit</code>	<code>none</code>	<code>exit</code>	Salir de PowerShell

El ejemplo a utilizar es el siguiente:

Programar una tarea con power Shell.

```
schtasks /create /tn "Calculadora"/sc monthly /d 1-10 /m JUN /st 12:00 /tr "c:\windows\system32\calc.exe"
```

```
schtasks /run /tn Calculadora
```