

12. Shell en UNIX/Linux

Sunday, August 30, 2020 6:42 PM

Como se mencionó anteriormente, el Shell es el encargado de establecer una comunicación entre el núcleo del sistema operativo (kernel) y el usuario. Linux es un sistema operativo basado en UNIX, el cual posee varios intérpretes de comandos (shells), siendo los más usados:

- **Bourne Again Shell (BASH).** /bin/bash
- **Tcsh.** /bin/tcsh

BASH es una versión extendida del intérprete "Bourne shell" de UNIX.

Características importantes

- Sintaxis para introducción de comandos en la terminal:
comando [parámetro1] [parámetro2] ...
- A diferencia de PowerShell, BASH si es sensible a mayúsculas y minúsculas.
- Al igual que con PowerShell, contamos con un historial de comandos, al cual podemos acceder con *flecha arriba* y *flecha abajo* del teclado.
- Puede dividir una línea larga de comando (que posea muchos parámetros) con el símbolo "\" y un Enter.
- Al igual que PS, la tecla TAB nos ayudará a autocompletar comandos.
- Las comillas admitidas por la sintaxis son:
 - Comillas inversas (') - Indican a la shell que tendrá que reemplazar lo que está encerrado entre ellas con su resultado.
 - Comillas sencillas (') - Le dicen al sistema que no haga ninguna expansión
 - Comillas dobles (") - Todo lo que esté dentro de llas será una cadena simple de texto, con excepción de lo que además esté entre comillas inversas, acompañado de el singo de pesos (\$) y la diagonal (\).
- BASH cuenta con variables de entorno, las cuales son variables configuradas por el sistema (es el equivalente a las variables automáticas de PS). Puedes ver las variables de entorno con el comando

\$env

- Los comandos básicos reciben parámetros usando un guion (-) seguido de una o más letras, qué además se pueden acompañar del valor de dicho parámetro.
- En ocasiones, se tienen los parámetros de palabra completa, las cuales se deben usar con dos guiones (--).

Comandos básicos

Igual que PS y cmd, en BASH ya existen comandos básicos los cuales puedes consultar aquí: <https://dev.to/awwsmm/101-bash-commands-and-tips-for-beginners-to-experts-30je>

Sintaxis básica

A continuación veremos los elementos básicos y la sintaxis para BASH scripting.

Shebang

También llamada hash-bang o sharpbang, es una instrucción que nos indica la ruta al intérprete de órdenes.

Para BASH, la shebang más habitual es:

```
#!/bin/bash
#!/bin/python3
#!/bin/perl
```

Pero si quieres tener un script con portabilidad, puede ser recomendable usar:

```
#!/usr/bin/env bash
#!/usr/bin/env python3
#!/usr/bin/env perl
```

Variables y operaciones

Variables

Las variables se declaran directamente cuando se van a usar, ya sea que se les inicialice o se usen directamente en un comando. Ejemplo:
`STR = "Hola mundo!"`

Como se puede observar, no es necesario definir el tipo de dato.

Entrada y salida estándar

Para la entrada de información desde el teclado podemos ejecutar:

```
read -p "Entra un dato:" variable
```

Para desplegar información en terminal:

```
echo "Variable capturada: $variable"
```

Operadores

Las operaciones básicas en BASH se pueden ejecutar con los operadores mostrados aquí:

https://es.wikibooks.org/wiki/El_Manual_de_BASH_Scripting_B%C3%A1sico_para_Principiantes/Operadores

Condicionales y ciclos

Condicionales

La sintaxis de un condicional simple (if elif else) es:

```
if [condición1];  
then  
Instrucción1  
elif [condición2];  
then  
Instrucción2  
else  
Instrucción3  
fi
```

Siendo posible colocar cero o más elif y, cero o un else. Observa que la estructura se cierra con fi y, si haces anidación, debe escribirse un fi por cada nivel de anidamiento.

Por otro lado, un equivalente al switch es case, cuya sintaxis es:

```
case $variable in  
  Valor1)  
    Instrucciones1  
  ;;  
  Valor2)  
    Instrucciones2  
  ;;  
  *)  
    Instrucciones3  
  ;;  
esac
```

Los casos para los diferentes valores pueden ser tantos como necesitemos y el caso de * implica una opción default, por si no coincide con ninguna otra opción. Observa que cada caso se "cierra" con doble ; y el case se cierra con esac.

Ciclos

BASH cuenta con el ciclo for, el cual puede ser utilizado de 3 formas diferentes:

```
for VARIABLE in VALORES  
do  
  Comandos  
done
```

El ejemplo anterior VARIABLE va tomando el valor de cada elemento indicado en VALORES.

Otro uso es:

```
for VARIABLE in ARCHIVOS
do
    Comandos
done
```

Donde ARCHIVOS puede ser uno o más archivos, los cuales iremos leyendo y cada línea se asignará a VARIABLE.

Finalmente, el for puede asociarse con un comando de BASH:

```
for OUTPUT in $(Comando-BASH)
do
    Comandos
done
```

Además de for, contamos con while:

```
while [condición]
do
    Comandos
done
```

Y, también, el ciclo until:

```
until [condición]
do
    Comandos
done
```

Referencias:

- Para revisar más información básica con BASH consulta [El Manual de BASH Scripting Básico para principiantes](#)