

Prácticas de Linux

Vamos a realizar cuatro prácticas del bloque de Linux. Estas cuatro prácticas versarán sobre aquellos apartados que hemos dejado un poco de lado en la parte teórica. Los puntos son:

Configuración de la red

1. Haz un guión que configure un PC con GNU/Linux para que pueda conectarse dentro del aula y hacia el exterior.
2. Haz un guión que haga ping a una dirección ip del aula, si esa dirección responde deber mostrarse esta encendido por pantalla.
3. Haz guión que tras ejecutarse muestre por pantalla la dirección física de eth0 dirección física

Argumentos

Esta parte es sumamente importante para continuar en nuestro camino hacia el certificado de profesionalidad. Hemos visto que los puntos fundamentales de cualquier programa son:

1. Recogida de datos
2. Procesado de datos
3. Muestra de los resultados (información)

Vamos a profundizar más en el tema de la recogida de datos. De momento sabemos recoger datos desde el teclado de la siguiente forma.

```
echo "Introduzca dato:"
```

```
cat>fl.....<CTRL + D>
```

```
dato_introducido=`cat fl`
```

Salta a la vista que esta no es una forma muy eficiente para hacerlo, en primer lugar requiere que usuario presione <CTRL + D>, lo cual no es nada normal y de otra parte requiere un procesado para recoger el valor desde un fichero a una variable. Esto es demasiado complicado.

La forma normal de realizar la recogida de datos en GNU/Linux es usando los argumentos. No penséis que esto es algo nuevo, ya que llevamos haciendolo desde que comenzamos con GNU/Linux.

Por ejemplo:

1. ls -l/ Este comando recibe dos argumentos “-l” y “/”
2. cat fichero1 fichero2 Este recibe otros dos argumentos, ¿Cuáles son?
3. cp -R carpeta1 carpeta2. Este recibe tres argumentos, ¿Cuáles son?

Por tanto lo único que tenemos que hacer es cambiar nuestra forma de recibir los datos, para ello tenemos que saber:

- \$#, número de argumentos
- \$1...\$9, argumentos del 1 al 9
- \$0, nombre del shell
- \$*, cadena con todos los argumentos

Por ejemplo: Implementar un guión que reciba como entrada dos argumentos y devuelva la suma de ellos.

```
#!/bin/bash
if [ $# -ne 2 ]
then
    echo "Debe introducir 2 Argumentos"
    exit 0
fi
```

```
echo "Soy el guión:"$0" voy a devolverte "$1" + "$2"
```

```
echo "El resultado es: "`expr $1 + $2`
```

Este guión se lanzaría así: sh sumador 1 2

Ejercicios

1. Realiza una calculadora que reciba tres argumentos, num1, num2, operación. De tal forma que los posibles valores de operación pueden ser: 1 (suma), 2 (resta), 3 (multiplicación *) ó 4 (división). De tal forma que: sh calculadora 3 1 1, debe devolver 4.
2. Haz un guión que reciba un numero de linea X y un fichero y nos devuelva la linea X pedida. Ampliarlo haciendo que también reciba un numero de columnas C y nos devuelva la linea X y columna C pedida.
3. Haz un guión que reciba tres argumentos y nos diga uno por uno si es par o impar.
4. Haz un guión que reciba cuatro argumentos, siendo cada uno de ellos: IP, Máscara, puerta de enlace y dns. De tal forma que tenemos que configurar la red de manera automática. Nota para añadir el dns tenemos que mirar como funciona la redirección de salida con concatenación, que es algo así: **echo "Hola">>fichero**

Filtros

Grep, grep es un filtro fundamental entra las herramientas de Linux, el uso que le vamos a dar será muy básico pero no obstante nos va a servir para muchas cosas.

Averigua que hacen las siguientes líneas

- grep palabra_buscada fichero
- grep -c palabra_buscada fichero
- grep -i palabra_buscada fichero

Ejercicios

1. Busca tu nombre de usuario en el fichero donde se almacenan los usuarios. Busca ahora también el nombre de tu grupo, haciendo uso primero de tu IDGRUPO y luego haciendo uso de tu nombre de usuario.
2. Busca la dirección ip de tu computador, luego busca en la lista de procesos un proceso llamado Xorg.
3. Usando man grep. Busca cuatro parámetros que consideres importantes de grep y trata de ponerles un ejemplo.

Otras utilidades

Números aleatorios

Si necesitamos una semilla o genera un número aleatorio en nuestros scripts con Bash podemos hacer uso de la variable RANDOM.

Ejemplo: `echo $RANDOM`

Te devolverá un string numérico entre 0 y 32768, cada vez que la ejecutes. El caso es por ejemplo si necesitas generar un número aleatorio dentro de un rango, por ejemplo entre 0 y 20, entonces ejecutarías:

- `echo $(($RANDOM%20))`
- `VALOR=`echo $(($RANDOM%20))``

Ejercicios

- Haz un guión que saque una línea aleatoria de un fichero. Deberá recibir como argumento el nombre de un fichero
- Muestra una columna aleatoria de la segunda línea del fichero `/etc/resolv.conf`.

Variables del sistema

- HOME, camino absoluto directorio de presentación
- PATH, donde busca el Shell ejecutables
- CDPATH, directorios donde se puede hacer `cd`
- PS1, PS2, signos de petición orden primaria y secundaria
- LOGNAME, nombre de presentación del usuario
- MAIL, directorio de correo entrante
- SHELL, nombre del Shell en uso

Ejercicios

1. Visualiza el contenido de las variables e intenta comprender su contenido.
2. Modifica Símbolo de petición de orden primaria.