

Administración de Sistemas Operativos - 1ª Evaluación (RA 7 – CE a, c)

Unidad Didáctica 2. Lenguajes de script

1. ¿Qué es el Shebang? Busca información sobre el Shebang en programación en Bash.
2. Crea un script “Hola Mundo”. ¿Qué cambios tengo que hacer en el entorno para poder ejecutar mi script?
3. ¿Qué implica ejecutar un script usando “. script.sh”, “./script.sh” o “source script.sh”?
4. ¿Por qué podemos ejecutar un script, que no tiene permisos de ejecución, con el comando “bash script.sh”?
5. Investiga cómo puedo visualizar las variables de entorno.
6. Busca cómo crear un script que defina una variable y la muestre por pantalla. ¿Qué diferencia existe entre usar la variable entre comillas simples o dobles? Cambia el anterior script para que salude al usuario actual y le diga dónde está su directorio casa.
7. ¿Qué ocurre cuando usamos \${a}? ¿Y si la cambiamos por \${!a}? Explica bajo qué circunstancias es necesario utilizar las llaves.
8. Comenta, con tus palabras, la siguiente captura:

```
ricardo@ricardo-VirtualBox:~/Escritorio/leccionScripts$ echo ${PATH}
/home/ricardo/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin
games:/snap/bin
ricardo@ricardo-VirtualBox:~/Escritorio/leccionScripts$ echo ${PATH:0:5}
/home
ricardo@ricardo-VirtualBox:~/Escritorio/leccionScripts$ echo ${PATH:6:7}
ricardo
```

9. A partir de la siguiente imagen, busca una manera alternativa de realizar lo expuesto.

```
ricardo@ricardo-VirtualBox:~/Escritorio/leccionScripts$ cat $(ls .) | head
#!/bin/bash

echo "Hola mundo"
#!/bin/bash

prueba="hola"

echo '$prueba=$prueba'
#!/bin/bash

ricardo@ricardo-VirtualBox:~/Escritorio/leccionScripts$ echo "$(date +"%x %r %Z")"
12/10/22 08:18:15 CEST
```

10. Observa las dos capturas inferiores, ¿qué crees que hace el guion que se encuentra delante de `_EOF_`? ¿Para qué sirven los `<<`?

```
#!/bin/bash

#Script utilizado para descargar ficheros vía FTP

#Definimos las variables
ftp_servidor="ftp.nl.debian.org"
ftp_path="/debian/dists/stretch/main/installer-amd64/current/images/cdrom"
ftp_file="debian-cd_info.tar.gz"

#<< le indica al comando el final del texto
ftp -n << _EOF_
open ${ftp_servidor}
user anonymous me@linuxbox
cd ${ftp_path}
hash
get ${ftp_file}
bye
_EOF_

ls -lart ${ftp_file}
```

```
#!/bin/bash

#Script utilizado para descargar ficheros vía FTP

#Definimos las variables
ftp_servidor="ftp.nl.debian.org"
ftp_path="/debian/dists/stretch/main/installer-amd64/current/images/cdrom"
ftp_file="debian-cd_info.tar.gz"

#<< le indica al comando el final del texto
ftp -n <<- _EOF_
    open ${ftp_servidor}
    user anonymous me@linuxbox
    cd ${ftp_path}
    hash
    get ${ftp_file}
    bye
    _EOF_

ls -lart ${ftp_file}
```

11. Implementa la última captura del ejercicio 10. Además, realiza cambios para probar otros comandos de GNU/Linux.

12. Utilizando el anterior método, para mostrar texto por pantalla, desarrolla un script donde, además de saludar al usuario, utilice el comando `df` y `du` para darle información del uso del disco. La salida generada deberá estar codificada en HTML para que pueda ser redireccionada a un fichero y visualizada desde el navegador.

13. Investiga cómo leer datos que vengan de la entrada estándar. Escribe un script donde se utilicen las opciones `-t`, `-s` y `-p`; y otro donde se utilicen las opciones `-i` y `-e`.

14. Partiendo de la base del ejercicio 13, desarrolla uno nuevo donde el script reciba 5 palabras, separadas por espacios, y las muestre por pantalla. ¿Cómo puedo cambiar el script para utilizar como separador el carácter ':'? Para lo último busca información sobre la variable `IFS`.

15. Implementa un script que reciba por la entrada estándar un usuario y muestre sus datos por pantalla. Obtendrá los datos del fichero `/etc/passwd` y mostrará su contenido separado por campos.

16. Investiga sobre el uso de parámetros en Bash. Cambia el anterior ejercicio para que reciba por la línea de comandos, como parámetro, el nombre del usuario.

17. Busca información sobre la estructura selectiva `If-Else` en Bash. Crea un script que reciba como parámetro de entrada la ruta de un fichero o un directorio. Deberá comprobar la existencia del mismo y, si existe, tendrá que indicar si se trata de un fichero o directorio y si tiene permiso de lectura, escritura y ejecución.

18. De igual forma que en el ejercicio 17, escribe un script que le pida al usuario dos cadenas de texto. Con las cadenas en variables, comprueba si tienen ambas contenido, si son iguales o distintas y si a nivel de orden la primera está por detrás de la segunda o viceversa.

19. Cambia el ejercicio 18 para utilizar la función test.

20. Crea un script que reciba, como parámetros de entrada, dos números y los compare para saber cuál es mayor. También, apoyándote en la expansión de variables, comprueba si el primer número es par.

21. Implementa un script que reciba, como parámetro de entrada, un número; y combine, dentro de las condiciones del If, dos expresiones. Deberá chequear que el número sea mayor que 0 y menor que 11.

22. Cambia el anterior ejercicio para negar la condición.

23. Busca información sobre la estructura selectiva Case y desarrolla un script con un menú de 4 opciones. El usuario deberá introducir un dígito (control del número de dígitos), correspondiente a una opción, mostrando por pantalla un mensaje indicando la opción seleccionada. Es importante configurar la lectura de la opción para que solamente acepte un dígito.

24. Reescribe el anterior ejercicio, aunque esta vez, para las opciones, utiliza clases de caracteres POSIX.

25. Crea un script donde, mediante un menú, el usuario pueda seleccionar opciones que realicen determinadas acciones dentro del sistema. La elección de dichas acciones se dejará al desarrollador.

26. A partir de la captura inferior, reescribe el script 23 para que esté en ejecución hasta que el usuario seleccione la opción de salida. ¿Qué ocurre si cambiamos la condición del bucle por el booleano true? Busca información sobre Break y Continue en bucles.

```
#!/bin/bash

contador=0

while [ $contador -lt 5 ]; do
    echo "$contador"
    contador=$((contador + 1))
done
```

27. ¿Qué ocurriría si en vez de utilizar un bucle While, en el ejercicio anterior, utilizamos un bucle Until?

28. Exponiendo en la parte inferior cómo recorrer un fichero con While, escribe un script que recorra el fichero /etc/passwd y compruebe la existencia de los directorios casa de todos los “usuarios regulares”.

```
#!/bin/bash

fichero="/etc/passwd"

while read linea; do
    echo "$linea"
done < $fichero
```

29. Transforma el ejercicio 28 para que trabaje con un array de campos en vez de con campos sueltos.

30. Vuelve a transformar el ejercicio anterior para que funcione utilizando un bucle For. Como pista, recuerda que podrías utilizar la sustitución de comandos para la generación del conjunto.

```
#!/bin/bash

for i in A B C D; do
    echo $i
done
```