

ESCUELA UNIVERSITARIA POLITÉCNICA DE TERUEL

GRADO EN INGENIERÍA INFORMÁTICA

ADMINISTRACIÓN DE SISTEMAS

PRÁCTICA 2

SHELL SCRIPT

Sergio Albiol-Pérez
Curso 2019/20

1. Objetivos de la práctica

El objetivo básico de la presente práctica es que el alumno sea capaz de desarrollar una serie de scripts en “**Scripting**” de forma adquiriera las habilidades correspondientes en la Administración de Sistemas Operativos y Redes de Computadores.

2. Actividades de laboratorio.

Ejercicio 1. Analiza las siguientes órdenes y describe su uso, al igual que todos aquellos argumentos que consideres interesantes.

- sort
- split
- kill
- ps
- pstree
- renice
- uptime
- hostname
- shutdown
- uname

Ejercicio 2. Escribir un script donde el usuario pasará un conjunto de valores por consola y automáticamente devolverá todos los valores sin ordenar por consola y posteriormente ordenados. Su sintaxis será:

`./ordenar 13 5 6 7 0 -6`

El script deberá contemplar todos los casos de error existentes, como podría ser el paso de algún parámetro de tipo no numérico. ¿qué sucedería si le pasamos al script más de 10 parámetros? Implementar las oportunas modificaciones para que lo soporte.

Ejercicio 3. Desarrollar un script llamado “configuración.sh” que nos muestre diferentes configuraciones del sistema:

1. Usuarios conectados y su logname.
2. El shell actual y los shells disponibles.
3. El directorio home y el PATH actual.
4. Número y nombre de usuarios que no están conectados, pero pertenecen al sistema.
5. Versión del Sistema operativo.
6. Configuración del ratón instalado.
7. Información de la CPU.
8. Información de la memoria.
9. Información de los diferentes discos duros del sistema.
10. Sistema de ficheros montados.

Ejercicio 4. Desarrollar un script que nos muestre por consola información relevante del identificador de proceso que el usuario inserte por consola.

Sintaxis: sh proceso.sh

El script deberá mostrar la siguiente información del proceso: identificador UID, propietario PID, PPID, Prioridad absoluta "C" , Prioridad Relativa "PRI" y la Dirección en memoria o en disco del proceso "ADDR".

Ejercicio 5. Desarrollar un script para que busque a aquellos procesos con una prioridad relativa superior a 10 y les incremente su prioridad en 5 unidades; posteriormente deberá mostrar que procesos han sido susceptibles de modificación junto con la nueva prioridad asignada.

Sintaxis: sh modprioridad.sh

Ejercicio 6. Realizar un shell script que genere cuentas de usuario.

Funcionamiento: ./genera_usuario login nombre_real grupo

- Login: el login del usuario a crear.
- Nombre_real: datos del usuario (nombre y apellidos).
- Grupo: únicamente podrá ser: contabilidad, finanzas, estadística.

Requisitos:

- El número de parámetros tiene que ser tres, en caso contrario tendrá que mostrar "Número de parámetros incorrecto" y deberá finalizar la ejecución.
- El primer parámetro deberá tener entre 3 y 8 caracteres, estando todos ellos en minúsculas, en caso contrario deberá mostrar "Login incorrecto: el login "login" deberá tener entre 3 y 8 caracteres" o bien "Login incorrecto: el login "login" deberá estar en minúsculas" y deberá finalizar la ejecución.
- Si el segundo parámetro contiene el carácter ":" deberá mostrar "el nombre real no puede contener el carácter :" y deberá finalizar la ejecución.
- Si el tercer parámetro no es ni contabilidad, ni finanzas, ni estadística, entonces deberá mostrar "grupo incorrecto: el grupo tiene que ser contabilidad, finanzas, estadística" y deberá finalizar la ejecución.
- Si el grupo seleccionado no se ha creado todavía, crear el primer GID que esté libre en el sistema [100 – 999], y si no existe ninguno libre, entonces deberá mostrar "No hay GID's libres" y deberá finalizar la ejecución.
- El directorio de inicio del usuario estará en /HOME y deberá llamarse igual que el login del usuario, si a la hora de crearse se produce un error, entonces deberá mostrar por consola el error pertinente.
- El usuario usará /bin/bash como shell de inicio.
- El script preguntará al usuario la contraseña asociada, para ello usará la orden passwd. En el caso de que el usuario se equivoque al insertar la contraseña tres veces consecutivas, entonces el script deberá bloquear la cuenta de usuario.
- Los ficheros de personalización se tomarán de /etc/skel

Finalmente deberás comprobar que la cuenta ha sido creada correctamente.

Ejercicio 7. Generar un script que nos permita generar ficheros tar, inicialmente el script nos mostrará un menú con las siguientes opciones:

- a. Generacion fichero tar
- b. Extracción fichero tar
- c. Visualización de la información del fichero tar
- d. Listado de todos los archivos incluidos en el fichero tar

Sintaxis: sh generatar.sh

Ejercicio 8. Realizar un Shell Script denominado nchmod que nos visualice por el código en decimal asociado a los permisos que le queramos dar a un fichero.

Ejemplo:

nchmod r-sr-xr-x el resultado que devolverá será 4555

El shell script tendrá que controlar el número de caracteres insertados por el usuario, mostrando un mensaje de error donde especifiquemos el uso

Uso: \$0 permisos-fichero (ejemplo r-srwxr-- tiene que tener 9 caracteres)"

Ejercicio 9. Realizar un script que nos genere password cifrados de 10 caracteres aleatorios, siendo dichos caracteres los que a continuación se describen dentro de un vector:

VECTOR="0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"

Posteriormente realizar la compresión de un fichero de texto:
casa bbb zzzzz ff → casa 3b 5z ff (2f)

Y a continuación realizar un cifrado simple de cada uno de los caracteres:

Carácter cifrado = (Código ASCII(carácter) + número constante) Módulo 256

¿Cómo se podría descryptar?

Nota: se puede usar el valor devuelto por la variable RANDOM