

U4 ADMINISTRACIÓN Y ASEGURAMIENTO DE LA INFORMACIÓN

ACTIVIDAD PRÁCTICA 14: Automatización (SCRIPTS)



Contenido

[Ejercicio 26 2](#_Toc135069910)

[Ejercicio 27 2](#_Toc135069911)

[Ejercicio 28 3](#_Toc135069912)

[Ejercicio 29 4](#_Toc135069913)

[Ejercicio 30 4](#_Toc135069914)

[Ejercicio 31 5](#_Toc135069915)

[Ejercicio 32 5](#_Toc135069916)

[Ejercicio 33 5](#_Toc135069917)

[Ejercicio 34 5](#_Toc135069918)

[Ejercicio 35 6](#_Toc135069919)

[Ejercicio 36 6](#_Toc135069920)

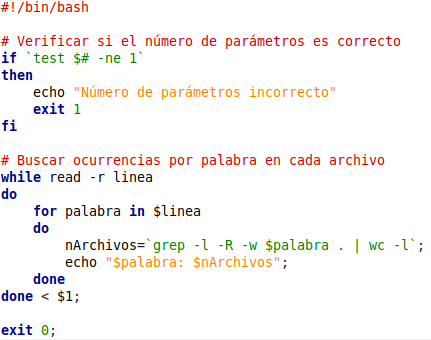
[Ejercicio 37 6](#_Toc135069921)

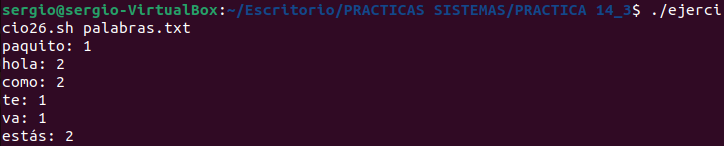
[Ejercicio 38 7](#_Toc135069922)

# Ejercicio 26

|  |  |
| --- | --- |
| 26 | Escriba un shellscript que, para todas las palabras de un archivo hipotético, palabras.txt, busque si existe algún archivo en su directorio actual (y su árbol de directorios completo) que contenga cualquiera de las palabras del archivo indicado. En concreto, para cada palabra, indique cuántos archivos la contienen.  Pista: es necesario combinar el comando find con el comando grep. |

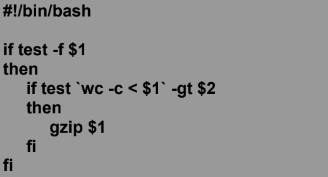






# Ejercicio 27

|  |  |
| --- | --- |
| 27 | Indique justificadamente cuál es la función del siguiente shellscript, indicando el significado más probable de sus argumentos. |

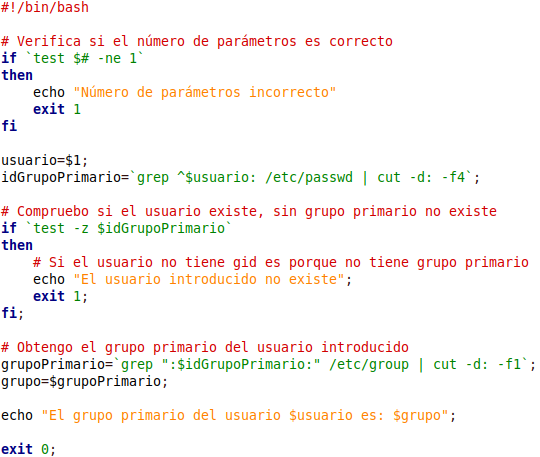


El siguiente shellscript comprueba que el primer parámetro sea un archivo. Seguidamente si lo es, cuenta si el tamaño del primer parámetro es mayor que el del segundo parámetro, si lo es lo comprime en gzip.

# Ejercicio 28

|  |  |
| --- | --- |
| 28 | Escriba un shellscript que, dado un nombre de usuario muestre por pantalla a qué grupos pertenece, ya sea ​​de forma primaria o secundaria, es decir, escriba un comando que haga lo mismo que el comando groups, para cualquier usuario. Debe resolver el ejercicio manipulando los archivos /etc/passwd y /etc/group y no puede utilizar los comandos groups e id. |

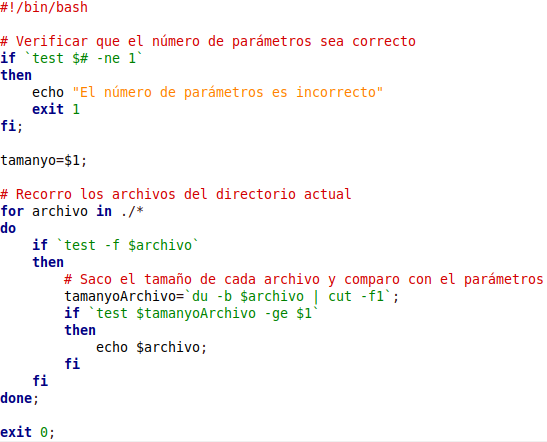
Solo he sabido sacar el grupo primario.

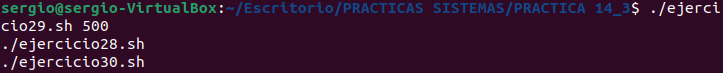




# Ejercicio 29

|  |  |
| --- | --- |
| 29 | Escriba un shellscript que muestre los nombres de los archivos ordinarios (no otros directorios) del directorio actual con tamaño igual o superior al tamaño especificado por el usuario como parámetro.  Recomendación: escriba un bucle que itere mostrando los nombres de todos los archivos del directorio actual. A continuación, haga que en cada iteración se tome el tamaño del archivo. Por último, si este tamaño es superior al parámetro del shellscript, muestre por pantalla el nombre del archivo. |

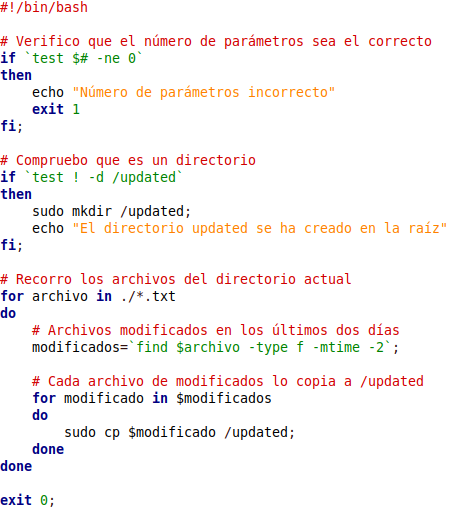




# Ejercicio 30

|  |  |
| --- | --- |
| 30 | Escriba un shellscript que copie en el directorio updated todos los archivos ordinarios del directorio actual y sus subdirectorios que hayan sido modificados a lo largo de las últimas 48 horas (para realizar pruebas puede resultarle útil el comando touch que permite modificar las fechas asociadas a un archivo). Asegúrese de que el directorio updated existe, y si no, créalo. |

No he conseguido hacerlo de forma recursiva.



# Ejercicio 31

|  |  |
| --- | --- |
| 31 | Haga un shellscript que cada N segundos compruebe si se ha montado un nuevo volumen en el sistema. En caso que el shellscript detecte que se ha montado un nuevo sistema de archivos, mostrará por pantalla un listado del directorio base donde se haya montado (archivos y directorios). N será un parámetro del shellscript.  Pista: le puede ser útil monitorear el resultado del comando df, por lo que, si de una iteración en la siguiente hay más líneas, utilice las líneas añadidas (tail), combinándolo con el comando cut con el fin de listar el contenido del directorio base en el que se haya montado.  Se imagina que este shellscript, en vez de listar, ¿borrara recursivamente todo el contenido del volumen montado? |

# Ejercicio 32

|  |  |
| --- | --- |
| 32 | Escriba un shellscript que cada N según compruebe si, en el último minuto, la parte entera de la carga media del sistema es superior a M. En caso afirmativo, debería imprimir un mensaje indicándolo. N y M serán argumentos del shellscript (para hacer pruebas, incremente la carga del sistema con un shellscript que tenga un bucle infinito con una llamada a date, por ejemplo). |

# Ejercicio 33

|  |  |
| --- | --- |
| 33 | Escriba un shellscript que indique cuántas particiones primarias, cuántas extendidas y cuántas lógicas  hay en su sistema. Tenga en cuenta el tipo de usuario necesario para ejecutar este shellscript. |

# Ejercicio 34

|  |  |
| --- | --- |
| 34 | Escriba un shellscript que, a partir de un nombre de archivo, un identificador de usuario (por ejemplo,  "buddy") y un permiso (por ejemplo, r); indique si el usuario tiene el permiso indicado sobre el archivo. Para comprobarlo, tendrá que estudiar tanto los permisos del usuario como los del grupo o grupos a los que pertenezca. |

# Ejercicio 35

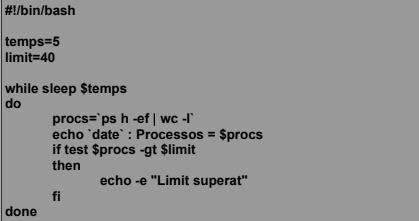
|  |  |
| --- | --- |
| 35 | Escriba un shellscript que cada N según compruebe si el archivo o directorio M tiene más de P links. En este caso, deberá mostrar un mensaje indicando que se ha superado el máximo de links. N, M y son parámetros de este shellscript. |

# Ejercicio 36

|  |  |
| --- | --- |
| 36 | Escriba un shellscript que hará de frontendo de los compresores/descompresores típicos de Linux (gzip y bzip2). El shellscript recibirá un único parámetro y se comportará de la siguiente forma:   1. Si el parámetro pasado es un archivo tipo gz o bz2, lo descomprimirá utilizando el descompresor adecuado. 2. Si el parámetro pasado es un archivo tipo tar.gz o tar.bz2, hará la descompresión y descompactación pertinentes (en un único paso). 3. Si el parámetro pasado es un archivo de cualquier otro tipo o un directorio, el sistema mostrará por pantalla el siguiente menú:    1. Gzip    2. Bzip2   De modo que se comprimirá en formato gz o bz2 en el caso de un archivo; o en formato tar.gz o tar.bz2 en el caso de un directorio. El archivo generado deberá tener el mismo nombre que el original.  Todas las comprobaciones anteriores se pueden realizar en función de la extensión del archivo. Supondremos que, si los archivos tienen la extensión correcta, son del formato que indica su extensión. |

# Ejercicio 37

|  |  |
| --- | --- |
| 37 | Indique justificadamente cuál es la función del siguiente shellscript: |



Tiene un bucle while que se ejecuta después de un tiempo. Dentro de él se almacena en una variable el número de procesos. Seguidamente, muestra la fecha y el número de procesos Después hay un condicional que habilita la interpretación de barra invertida si el número de procesos es mayor que el límite asignado.

# Ejercicio 38

|  |  |
| --- | --- |
| 38 | Escriba un shellscript que, cada N segundos, compruebe cuál es el número total de procesos en ejecución en la máquina. En caso de que supere el valor M, el shellscript deberá enviar la señal SIGTERM a todos los procesos que estén ejecutando el programa P y añadir al archivo F los pids de los procesos a los que ha enviado el signal. N, M, P y F serán parámatras del shellscript (y en éste orden); por ejemplo, 60 100 firefox /tmp/log. |

NO HEMOS DADO SEÑALES