

Ejercicio propuesto de Bash. Curso 25-26.

Objetivos

Es hora de aprender a manejar los conceptos vistos en semanas anteriores para crear scripts por nosotros mismos que automaticen las tareas más comunes.

Evaluación

La evaluación de este trabajo se realizará con una prueba presencial práctica de tipo individual, donde tendrás que resolver problemas similares a los presentados aquí. En esa prueba **no dispondrás del código desarrollado para resolver los ejercicios que aquí se proponen**. Se realizará en un ordenador sin conexión a internet y dispondrás de los apuntes sobre bash del aula virtual, así como de las páginas de la herramienta man.

Introducción: comando lsof

El comando **lsof** se utiliza para listar todos los archivos abiertos en el sistema. El comando es complejo y admite diversas opciones. Una de las más interesantes es **-u**, que permite conocer todos los archivos abiertos por un usuario concreto:

```
$ lsof -u root
```

Tareas

El script a realizar se llamará `open_files`, y admite las siguientes opciones:

```
open_files [-h] [-help] [-f regexp] [-o] [--offline] [-u user1 user2 ...] [-s] [-sort]
```

En la notación anterior las letras seguidas de un guión como `-h`, o de dos guiones como `--offline` son opciones. Las que van después de un guión se llaman opciones en formato corto (una letra), mientras que las que van después de dos guiones se llaman opciones en formato largo. A veces una misma funcionalidad está expresada por la opción en formato corto y también en formato largo, por ejemplo `-h` y `--help`, ya que ambas indican que se desea mostrar la ayuda del comando.

Observa que las opciones pueden combinarse e introducirse en cualquier orden.

El script implementará las siguientes funcionalidades:

1. Funcionamiento básico y listado de usuarios con archivos abiertos

Usando lo aprendido hasta el momento en clase, **desarrolla el script 'open_files' que, si no se especifica opción, envía a la salida estándar una lista ordenada alfabéticamente por el nombre de usuario y sin duplicados de todos los usuarios conectados en el**

sistema y la cuenta del número de ficheros regulares abiertos que tiene actualmente cada usuario. Los archivos regulares se especifican por lsof con las letras REG en la columna Type.

En este comportamiento por defecto, este listado estará formado por los usuarios que se incluyen en el resultado del comando who, y no usuarios como demonios o utilizados por servicios que no han hecho login. También se debe incluir el UID del usuario, y el PID de su proceso más antiguo.

1. Recuerda facilitar la lectura de tu código:

1. Usa comentarios en los elementos más complejos del programa para que te sea más sencillo entenderlo cuando haya pasado un tiempo y no te acuerdes
2. Usa funciones para compartmentar el programa, así como variables y constantes (variables en mayúsculas) que hagan tu código más legible.

2. Incluye código para procesar la línea de comandos.

1. Se debe mostrar ayuda sobre el uso del comando si el usuario emplea la opción -h o --help.
2. Se debe indicar el error y mostrar ayuda sobre el uso si el usuario emplea una opción no soportada

3. Ojo con la sustitución de variables y las comillas. En caso de problemas piensa en cómo quedarían las sentencias si las variables no valieran nada ¿tendría sentido para BASH el comando a ejecutar?

4. Maneja adecuadamente los errores.

1. En caso de error muestra un mensaje y sal con código de salida distinto de 0. Recuerda la función `error_exit` explicada en los apuntes y, si lo crees conveniente, reutilízala o haz la tuya propia.
2. Trata como un error que el usuario emplee opciones no soportadas
3. Detecta si `lsof` está instalado antes de usarlo. Si no lo está, indica el error.
4. Haz lo mismo con las otras posibles condiciones de error que se te ocurran ¿has probado a invocar tu programa opciones absurdas a ver si lo haces fallar?

2. Variable de entorno OPEN_FILES_FOLDER

En el caso de que la variable de entorno OPEN_FILES_FOLDER esté asignada a una ruta en el sistema de archivos a un directorio, se entenderá que la cuenta de archivos solo incluirá aquellos archivos dentro de ese directorio.

Comprueba primero si la variable no es nula y en el caso de que no lo sea, que la ruta lleve a un directorio existente. Si la variable no es nula, pero lleva a un directorio no existente hay que terminar el script con un estado de error.

El comando lsof puede combinar los criterios de elección de archivos abiertos, por ejemplo los pertenecientes a un usuario o los existentes en un directorio. Por defecto utiliza un or lógico, pero en este caso puede que te interese más un and lógico, en cuyo caso en la línea de comando de lsof debes separar los criterios con la opción -a.

3. Busca archivos según un patrón

La última columna de **Isof** muestra la ruta del archivo abierto.

Añade la opción '-f filtro' de tal forma que filtro pueda ser una expresión regular. Esta expresión se usaría para filtrar la salida de Isof en base a la última columna. Es decir que al usar tu script así:

```
open_files -f '.*sh'
```

En lugar de contar todos los ficheros abiertos, se contarán solo aquellos en los que la ruta encaje con esta expresión regular.

Para refinar los resultados, recuerda que un \$ en una expresión regular en grep que indica un match con el final de la línea. Quizás sea buena idea añadir este carácter al filtro indicado por el usuario.

4. Ficheros de Usuarios no conectados

Si se incluye el parámetro -o o --offline en su versión larga se debe incluir el listado anterior pero únicamente para los usuarios que no estén conectados al sistema, es decir **aquellos que no salgan en el resultado de who. Si el usuario no tiene un proceso más antiguo se mostrará la cadena NA en el campo que le corresponde en la tabla.**

5. Filtra por usuario

Si el script se invoca así:

```
open_files -u jmtorres jttoledo dabreu
```

donde la opción -u va acompañada de una lista de nombres de usuario de tamaño arbitrario y separados por espacios, sólo se muestre la información de Isof para aquellos archivos abiertos por los usuario especificados en la opción -u. Esta opción debe poderse combinar con el resto de opciones, incluida la opción -f. Esto implica que si se utiliza sin la opción -o solo se mostrarán en la tabla los usuarios conectados (incluidos por who) mientras que con la opción -o solo se mostrarán en la tabla los usuarios no conectados (los que no se obtengan con who)

6. Ordenamiento

Si se utiliza la opción -s, la lista de usuarios debe mostrarse ordenada por el número de archivos abiertos.