

Tema 7 – Ejercicio 7-1

NOTAS:

- Todos los scripts de esta relación deben estar preparados para ser lanzados en un sub-shell de bash
 - La primera acción que deben realizar todos los scripts generados es comprobar que han recibido el número correcto de argumentos. En caso de no ser así, mostrarán un mensaje tipo "USO: <nombre de script> ...", y terminarán devolviendo un 1
 - Crear los ficheros y directorios que sean necesarios para probar los diferentes scripts
- 1) Usar un cliente ssh para conectar con el sistema GNU/Linux instalado en el equipo "Profesor". Todos los alumnos trabajaran simultáneamente conectados a la misma máquina
 - 2) Crear un script de nombre *permisosEnDirectorio.sh* que reciba como argumento el nombre de un directorio. El script debe realizar las siguientes acciones:
 - Comprobará si el argumento recibido es realmente un directorio. En caso contrario mostrará un mensaje de error por pantalla y terminará devolviendo un 1
 - Informará de los permisos que el usuario que ejecuta el script tiene sobre el directorio recibido, y terminará devolviendo un 0. Por ejemplo, si recibe un directorio con la configuración drwxr-x---, y el usuario que ejecuta el script es miembro del grupo propietario del directorio, el script mostrará por pantalla:

Tienes permiso de:
Lectura
Ejecución
 - 3) Crear un script de nombre *editar.sh* que reciba como único argumento el nombre de un fichero. El script servirá para editar ficheros empleando un editor parametrizado mediante una variable, aunque previamente realizará una serie de comprobaciones:
 - Si el argumento recibido es el nombre de un fichero regular que existe previamente, el script deberá comprobar si el usuario tiene permiso de lectura y de escritura sobre dicho fichero. De ser así, el script abrirá el editor parametrizado pasándole como argumento el nombre del fichero. En caso contrario mostrará el mensaje "ERROR: el usuario no tiene permisos de lectura y/o escritura sobre <nombre del fichero>" y finalizará devolviendo un 1
 - Si el argumento recibido no es el nombre de un fichero regular preexistente, el script abrirá el editor parametrizado pasándole como argumento el nombre del fichero

- 4) Crear un script de nombre *numeroVecesComando.sh*, que recibirá como único argumento el nombre de un comando. El script servirá para calcular el número de veces que el usuario que ejecuta el script ha ejecutado el comando pasado como argumento (por ejemplo, una salida posible del script podría ser "El usuario dvegash ha ejecutado 8 veces el comando who")
- 5) Crear un script de nombre *lecturaYescritura.sh* que no recibirá argumentos. El script servirá para listar en formato largo los nombres de los ficheros regulares del directorio actual sobre los que el usuario que ejecuta el script tiene permiso de escritura y de lectura
- 6) Crear un script de nombre *contenidoDelPATH.sh* que no recibirá argumentos. El script servirá para listar en formato largo los nombres de los ficheros incluidos en cada una de las rutas que forman parte de la variable del shell PATH
- 7) Crear un script de nombre *pingMultiple.sh* que recibirá como argumento una o más direcciones IP. El script comprobará, para cada una de las direcciones IP recibidas, si obtiene o no respuesta de ellas al enviarles un comando *ping*¹:
 - Para cada dirección IP de la que obtenga respuesta, el script mostrará el mensaje: "La dirección <direccion IP> responde correctamente"
 - Para cada dirección IP de la que no obtenga respuesta, el script mostrará el mensaje: "La dirección <direccion IP> no responde"
- 8) Crear un script de nombre *comprobarUsrConectados.sh* que recibirá como único argumento el nombre de un fichero, que contendrá uno o más nombres de login (cada uno en una línea distinta).
 - El script comprobará si el fichero recibido como argumento existe, y si el usuario que ejecuta el script tiene permiso de lectura sobre él. De no ser así mostrará el mensaje "ERROR: el fichero <fichero> no es accesible", y terminará devolviendo un 1
 - Si el fichero recibido como argumento es accesible, el script comprobará para cada uno de los nombres de login incluidos en él, si el usuario correspondiente está conectado o no al sistema:
 - Si el usuario está conectado, el script obtendrá además el número de sesiones que tiene establecidas el usuario, y mostrará el mensaje "El usuario <nombre de login> está actualmente conectado (<N> sesiones)"
 - Si el usuario no está conectado, el script mostrará el mensaje "El usuario <nombre de login> no está actualmente conectado"

¹ Emplear la opción -c para limitar el número de intentos de obtener respuesta, y evitar así que el programa se quede indefinidamente sondeando la IP