

Práctica1 - Scripting

Sistemas Operativos - Curso 2º Grado en Ingeniería de Computadores

Curso 2023-24



1. Descripción

En esta práctica se abordará la utilización del **intérprete de mandatos de Linux**, utilizándolo como herramienta de programación para crear nuevos mandatos sin tener que usar las llamadas al sistema operativo. Es importante hacer notar que este intérprete **nos va a permitir automatizar** muchas de las **tareas** diarias que son utilizadas **en los sistemas** basados en Linux.

Para poder gestionar nuestro sistema operativo regularmente hay que prestar especial atención a los usuarios registrados en el mismo ya que son siempre el eslabón más débil de la seguridad del mismo. Por ello, el administrador debería saber de qué manera interactúa con el sistema operativo cada uno de ellos y cuáles son los privilegios asignados al mismo. Esto nos puede ayudar a detectar asignaciones de privilegios no deseadas o actividad sospechosa que deba ser tenida en cuenta.

El enfoque lo podemos hacer desde los siguientes puntos de vista:

- Inicios de sesión correctos y fallidos realizados por el usuario tanto locales como remotos.
- Ficheros propiedad del usuario
- Asignación de pertenencia a grupos

Los inicios de sesión se pueden observar en los ficheros de registro del sistema. En concreto en el fichero **/var/log/auth.log**. Este fichero proporciona un registro de todas las actividades que implican un proceso de autenticación. Por ejemplo, registra los usuarios autenticados y activos en sistema operativo. Registra el día, hora, usuario y órdenes que se han ejecutado con el comando sudo, los *cronjobs* que se han ejecutado, los intentos fallidos de autenticación, etc.

En el fichero **/etc/group** encontramos las asignaciones de usuarios a grupos del sistema. Especial cuidado debemos tener con los usuarios agregados al grupo **sudo** ya que tendrán permiso de ejecución como administrador en nuestro sistema.

En esta primera práctica, se pide desarrollar un *script* denominado **info_user.sh** que admita uno o varios argumentos. En concreto, la llamada al *script* tendrá la siguiente forma:

```
info_user.sh [-u usuario][[-g grupo]][[--login]][[--help]]
```

El **objetivo** del *script* será buscar información referente a un usuario. Además, se han de tener en cuenta los siguientes requisitos:

Atendiendo al prototipo de llamada, el *script* podrá recibir como **argumentos**:

- **Dos argumentos:** En este caso discriminaremos dos casos diferentes en función del valor del primero de ellos.
 - Si en el primer parámetro encontramos un valor “-u”, el script debe interpretar el segundo valor como el nombre de un usuario y mostrar por pantalla la siguiente información:
 - Indicar si el usuario está conectado actualmente al sistema
 - Fecha y hora de las 5 últimas conexiones del mismo junto con la IP y la forma de acceso (LOCAL o REMOTA) obtenidas del fichero auth.log
 - Lista de grupos a los que pertenece
 - Espacio ocupado por su carpeta de usuario y cantidad total de ficheros en su carpeta de usuario que ocupen más de 1MB
 - Si en el primer parámetro encontramos un valor “-g”, el script debe interpretar que el segundo parámetro es el nombre de un grupo y debe indicar los siguientes datos:
 - Mostrar los usuarios asociados a dicho grupo por pantalla
- **Un argumento:** En este otro caso discriminaremos entre dos posibilidades de nuevo en función del valor del argumento.
 - Si en el argumento encontramos el texto “--login”, debe indicar por pantalla los 5 últimos usuarios que han accedido al sistema junto con su fecha y hora de acceso y si lo han hecho de manera local o remota indicando en el segundo caso la IP utilizada para ello.
 - Si en el argumento está el texto “--help” mostraremos un texto con la ayuda de nuestro script indicando las diferentes posibilidades de ejecución.
- Sin argumentos: Debe ejecutar lo mismo que con “--help”.

El *script* producirá, por la salida estándar, los mensajes de error y de aviso que se consideren necesarios para la explicación de cada problema y generará un error de salida diferente para cada caso.

Para poder probar adecuadamente el script se debe añadir al sistema varios usuarios e iniciar sesión con algunos de ellos en diferentes momentos para generar registro de acceso local. Opcionalmente, si queréis generar registros remotos, podéis instalar el paquete ssh y conectarnos desde una terminal externa. Como ejemplo de prueba, se adjunta en el aula virtual un fichero de registro de accesos (**auth.log**) que podeis copiar en /var/log para las pruebas, pero tenéis que recordar que este fichero se sobrescribe cada vez que se inicia el sistema operativo.

Nota: tenga en cuenta que se deberá comprobar el valor devuelto por cada mandato utilizado y acabar la ejecución del *script* notificando el error en el caso de que el mandato haya producido un error irrecuperable.

2. Ejemplos de funcionamiento

```
carlos@UB-CARLOS:~$ sudo ./info_user.sh -e
Error: Opción no válida. Use --help para obtener ayuda.
```

```
carlos@UB-CARLOS:~$ ./info_user.sh --help
info_user.sh [-u usuario][[-g grupo]][[--login]][[--help]]
  -u usuario      Mostrar información sobre el usuario especificado
  -g grupo        Mostrar usuarios asociados al grupo especificado
  --login         Mostrar los 5 últimos usuarios que han accedido al sistema
  --help          Mostrar ayuda
```

```
carlos@UB-CARLOS:~$ ./info_user.sh --login
```

```
las últimas 5 conexiones REMOTAS son:
```

```
-----
Feb 14 20:03:26 carlos 192.168.1.111
Feb 14 20:04:02 maria 192.168.1.111
Feb 14 20:04:22 carlos 192.168.1.111
Feb 14 20:05:12 maria 192.168.1.111
Feb 14 20:05:46 maria 192.168.1.111
```

```
las últimas 5 conexiones LOCALES son:
```

```
-----
Feb 14 19:57:06 maria(uid=1003)
Feb 14 20:02:37 carlos(uid=1000)
Feb 14 20:04:02 maria(uid=1003)
Feb 14 20:05:13 maria(uid=1003)
Feb 14 20:20:55 carlos(uid=1000)
```

```
carlos@UB-CARLOS:~$ sudo ./info_user.sh -u
Error: Se debe proporcionar el nombre de usuario con el parámetro -u
```

```
carlos@UB-CARLOS:~$ sudo ./info_user.sh -u pepe
El usuario pepe no existe en el sistema
```

```
carlos@UB-CARLOS:~$ sudo ./info_user.sh -g
Error: Se debe proporcionar el nombre de grupo con el parámetro -g
```

```
carlos@UB-CARLOS:~$ ./info_user.sh -g ventas
El grupo ventas no existe en el sistema
```

```
carlos@UB-CARLOS:~$ ./info_user.sh -g proyectos
Usuarios en el grupo proyectos:
maria
luis
```

```
carlos@UB-CARLOS:~$ sudo ./info_user.sh -u maria
el usuario maria está conectado actualmente
```

```
las últimas 5 conexiones REMOTAS son:
```

```
-----
Feb 12 13:55:40 172.16.124.155
Feb 14 13:28:34 172.16.124.155
Feb 14 20:04:02 192.168.1.111
Feb 14 20:05:12 192.168.1.111
Feb 14 20:05:46 192.168.1.111
```

```
las últimas 5 conexiones LOCALES son:
```

```
-----
Feb 14 13:32:54
Feb 14 13:34:21
Feb 14 19:57:06
Feb 14 20:04:02
Feb 14 20:05:13
```

```
maria pertenece a los siguientes grupos: maria,proyectos,empresa,empleados
```

```
Espacio ocupado por la carpeta de maria (/home/maria): 12M
```

```
Contiene 63 ficheros mayores de 1MB en la carpeta /home/maria
```

```
carlos@UB-CARLOS:~$ ./info_user.sh --login
```

```
las últimas 5 conexiones REMOTAS son:
```

```
-----
```

```
Feb 14 20:03:26 carlos 192.168.1.111
```

```
Feb 14 20:04:02 maria 192.168.1.111
```

```
Feb 14 20:04:22 carlos 192.168.1.111
```

```
Feb 14 20:05:12 maria 192.168.1.111
```

```
Feb 14 20:05:46 maria 192.168.1.111
```

```
las últimas 5 conexiones LOCALES son:
```

```
-----
```

```
Feb 14 19:57:06 maria(uid=1003)
```

```
Feb 14 20:02:37 carlos(uid=1000)
```

```
Feb 14 20:04:02 maria(uid=1003)
```

```
Feb 14 20:05:13 maria(uid=1003)
```

```
Feb 14 20:20:55 carlos(uid=1000)
```

3. Objetivos parciales

- Comprobación de argumentos. (1 punto)
 - Se debe **comprobar que el número de argumentos recibidos es el correcto**. Si alguno de los datos pasados como argumentos no es congruente, el programa **no continuará su ejecución** y se mostrará un mensaje de error indicándolo por la salida de error estándar.
- Mensajes de salida. (2 puntos)
 - Antes de realizar procesamiento alguno, el *script* mostrará por la salida estándar: el **nombre** del **usuario** que ejecuta el *script*, la **fecha y hora** de su ejecución y la **versión de *bash*** utilizada. Se valorará la gestión de los mensajes de salida en cada una de las funcionalidades de la práctica.
- Gestión de errores. (1 puntos)
 - Se debe **comprobar el valor devuelto** por cada mandato utilizado y acabar la ejecución del *script* notificando el error, en caso de que el mandato haya acabado con un error.
- Salida a fichero de log (1 puntos)
 - Se debe programar el script para que los resultados además de ser mostrados por pantalla también se guarden en un fichero de log en: **/var/log/info_user.log**
- Ejecución correcta de la funcionalidad del *script*. (5 puntos)

Nota: las puntuaciones para cada objetivo parcial son las puntuaciones máximas que se pueden obtener si se cumplen esos objetivos.

Nota: no se debe hacer un programa separado para cada objetivo, sino un único programa genérico que cumpla con todos los objetivos simultáneamente.

4. Entrega de prácticas

La entrega de prácticas se hará **a través del Campus Virtual** en las fechas anunciadas en el mismo. Se debe entregar **un único archivo resultado de la práctica denominado info_user.sh**. Dicho fichero debe estar debidamente comentado. Además, se debe entregar una memoria de la práctica en formato **pdf**. La memoria debe realizarse partiendo de la plantilla WORD aportada en el aula virtual y debe incluir:

- Índice de contenidos
- Autores
- Descripción del *script*: incluyendo descripción de las principales funciones implementadas.
- Comentarios personales: incluyendo problemas encontrados, críticas constructivas, propuesta de mejoras y evaluación del tiempo dedicado.
- No incluir código fuente
- Respetar el número máximo de hojas especificadas por apartado.
- Eliminar los textos que aparecen en la plantilla con fondo verde y amarillo (son explicaciones, pero no deben ser parte de la memoria entregada).
- **NO DESCUIDE LA CALIDAD DE LA MEMORIA DE SU PRACTICA.** Aprobar la memoria es tan imprescindible para aprobar la práctica, como el correcto funcionamiento de esta. Si al evaluar la memoria se considera que no alcanza el mínimo admisible, la práctica se considerará SUSPENSA.

IMPORTANTE: Todos los archivos solicitados (fuente y memoria) deberán empaquetarse en un único archivo comprimido en formato ZIP llamado practica1.zip y entregarse a través del Campus Virtual pinchando en el enlace correspondiente.

5. Evaluación de la práctica

La práctica se evaluará comprobando el **correcto funcionamiento** de los distintos objetivos, y valorando la **simplicidad del código**, los **comentarios**, la **gestión de errores** y la **calidad de la memoria**. El profesor pedirá una defensa oral de la misma.

6. Autoría de la práctica

La práctica se debe realizar en **grupos de 2 personas como máximo**. El hecho de detectar **copia** en las prácticas expone a los alumnos a la **posibilidad de una apertura de expediente disciplinario y expulsión**. Una práctica será considerada copia en caso de que

contenga la totalidad o una parte de la práctica de otro alumno. **Se considerará copia en caso de:**

- Archivos que contengan la totalidad o fragmentos de código de otro alumno o generados por IA.
- Memorias con la totalidad o fragmentos de frases e imágenes de otro alumno.