

Escuela de Ingeniería Informática

Administración de Sistemas Operativos, Curso 2019-20

Informe trabajo del tema 6

Trabajo final

Galindo Molina, Luis Enrique	Y3328300F
Lárez Romero, Jesús Alberto A.	Y5685670S
Marín Reyes, Ignacio José	54123926G

Grupo 2/44
Grupo 2/44
Grupo 2/44

Estrategia de implementación.....	2
Opciones adicionales	2
Prueba del correcto funcionamiento	3

Estrategia de implementación

La manera en que se decidió atacar el problema fue mediante el uso de un archivo secundario para poder guardar toda la información que se pide en la práctica, para su posterior organización y representación en pantalla, eliminando finalmente dicho archivo.

El archivo es creado bajo el directorio **/var**, para seguir las reglas de la estructura de archivos de Linux, dado que es un archivo que varía con el tiempo, cabe destacar que, antes de crear el archivo se revisa si existe y se le cambia el nombre en caso de haber alguna coincidencia para evitar sobrescribir archivos ajenos o problemas en la ejecución del script.

Tras haber creado el archivo que servirá de “buffer” para guardar los datos necesarios, se realiza una búsqueda con el comando **FIND** dentro de la carpeta **/PROC** para revisar todos los procesos que se están ejecutando en ese momento en el sistema. Luego buscamos dentro de cada una de las carpetas de los procesos encontrados si están haciendo uso de algún fichero regular, esto se puede comprobar examinando cada carpeta de **/PROC/PROCESO/FD**, dado que en **/FD** se crean hipervínculos que señalan a los ficheros que estén utilizando dicho proceso. Si al comprobar cada fichero dentro del **/FD** correspondiente, se encuentra que es un fichero regular, se guarda su ruta para su posterior uso y se guarda su descriptor en variables, la segunda variable (el descriptor) es guardada para posteriormente ser cotejada contra los descriptors que se encuentran dentro de la carpeta **/FDINFO** del mismo proceso, si se encuentran dos descriptors iguales significa que es el descriptor que apunta al fichero que estaba siendo usado antes, y tras hacer esto se revisa el modo en el que está abierto dicho archivo. Este último paso se hace simplemente revisando los últimos dígitos al final del **flags**, un 0 implica lectura, un 1 implica escritura y un 2 implica lectura y escritura.

Una vez generado el archivo secundario con una estructura que nos muestra por cada proceso, las rutas de ficheros en uso, el modo de apertura, y el propietario del fichero. Podemos apoyarnos en esta información para a través de un sort y un uniq -c, obtener la información solicitada, es decir, por cada fichero mostrar el número de procesos que lo tienen abierto para lectura y para escritura, con el nombre del propietario del fichero.

El programa además realiza un control de errores, comprueba que el número de argumentos sea correcto (dos o ningún argumento) y que al especificar la opción -u, la lista de usuarios pasados por parámetro corresponda con usuarios del sistema, verificados con el fichero **ETC/PASSWD**.

Opciones adicionales

La opción sin argumentos sería el resultado del apartado anterior, en cambio, con la opción -u, al realizar el sort y el uniq -c vamos comprobando que el propietario del fichero pertenezca a la lista de usuarios pasados por parámetro.

Prueba del correcto funcionamiento

Probamos sin argumentos.

```
[root@localhost Escritorio]# ./Script
```

Read	Owner = root	Count = 27	/root/.xsession-errors
Write	Owner = root	Count = 27	/root/.xsession-errors
Write	Owner = root	Count = 1	/tmp/gconfd-root/saved_state_0b6872a1860413f1dc8c37bb0000001c
Write	Owner = root	Count = 1	/var/lib/dhclient/dhclient-95e4a9da-8751-495e-92e6-9d889ffe388c-eth5.lease
Read	Owner = root	Count = 1	/var/lib/PackageKit/transactions.db
Write	Owner = root	Count = 1	/var/lib/PackageKit/transactions.db
Read	Owner = postfix	Count = 1	/var/lib/postfix/master.lock
Write	Owner = postfix	Count = 1	/var/lib/postfix/master.lock
Write	Owner = root	Count = 1	/var/log/audit/audit.log
Write	Owner = root	Count = 1	/var/log/ConsoleKit/history-20200105
Write	Owner = root	Count = 1	/var/log/cron
Write	Owner = root	Count = 1	/var/log/gdm/:0.log
Write	Owner = root	Count = 4	/var/log/gdm/:0-slave.log
Write	Owner = root	Count = 1	/var/log/messages
Write	Owner = root	Count = 1	/var/log/wpa_supplicant.log
Write	Owner = root	Count = 1	/var/log/Xorg.0.log
Read	Owner = root	Count = 1	/var/run/crond.pid
Write	Owner = root	Count = 1	/var/run/crond.pid
Read	Owner = gdm	Count = 1	/var/run/gdm/auth-for-gdm-4GKCsv/database
Write	Owner = gdm	Count = 1	/var/run/gdm/auth-for-gdm-4GKCsv/database
Read	Owner = root	Count = 1	/var/run/gdm/auth-for-root-HD4DRE/database

Ejecutamos con la opción -u.

```
[root@localhost Escritorio]# ./Script -u gdm,postfix
```

Read	Owner = postfix	Count = 1	/var/lib/postfix/master.lock
Write	Owner = postfix	Count = 1	/var/lib/postfix/master.lock
Read	Owner = gdm	Count = 1	/var/run/gdm/auth-for-gdm-4GKCsv/database
Write	Owner = gdm	Count = 1	/var/run/gdm/auth-for-gdm-4GKCsv/database

Comprobamos el control de errores.

```
[root@localhost Escritorio]# ./Script 2
```

Número de argumentos incorrecto

El formato debería ser: ./Script [-u lista usuarios]

```
[root@localhost Escritorio]# ./Script -u root,jesus,luis,ignacio
```

jesus no es un usuario

El formato debería ser: ./Script [-u lista_usuarios]

Mostramos un fragmento para comprobar la estructura del archivo secundario y temporal.

```
[root@localhost var]# cat /var/Temp-Script
```

```
/root/.xsession-errors 1 root
/root/.xsession-errors 0 root
/root/.xsession-errors 1 root
/root/.local/share/gvfs-metadata/home 0 root
/root/.local/share/gvfs-metadata/home-ec446f08.log 0 root
/root/.local/share/gvfs-metadata/root 0 root
/root/.local/share/gvfs-metadata/root-049a36af.log 0 root
/proc/mdstat 0 root
/proc/2020/mountinfo 0 root
/root/.xsession-errors 0 root
/root/.xsession-errors 1 root
/root/.xsession-errors 0 root
```