SESSIÓN TP3

Sistemas Operativos 1
Marzo 2025

1 INTRODUCCIÓN

En esta sesión teórico-práctica se presentan tres ejercicios relacionados con el segundo tutorial y segunda práctica de la asignatura. A continuación, se propone un ejercicio que se deberá entregar a través de la tarea correspondiente del campus virtual, y que será evaluado por vuestros compañeros y compañeras.

2 EJERCICIOS A REALIZAR DURANTE LA SESIÓN

Ejercicio 1

El comando `find` es uno de los comandos más potentes en el entorno Linux para buscar archivo o directorios. A continuación, se muestran varios ejemplos de su uso. Analiza la información que imprime por pantalla y las opciones de ejecución usadas. ¿Qué hacen cada uno de estas comandas?

```
find gutenberg/ -type d

find gutenberg/ -type f

find gutenberg/ -name "*.txt"

find gutenberg/ -name "*.txt" -size +512k -size -1024k

find gutenberg/ -name "*.txt" -size +512k -size -1024k > fitxer.txt
wc -l fitxer.txt

find gutenberg/ -name "*.txt" -size +512k -size -1024k | wc -l

find gutenberg/ -name "*.txt" -size +512k -size -1024k -exec wc -l {}
\;

find gutenberg/ -name "*.txt" -type f -size +512k -exec ls -l {} \;
```

Ejercicio 2

En la práctica 1 se ha utilizado la comanda `awk`. El comando `awk` incluye todo un lenguaje de programación propio que permite realizar un gran número de tareas. Como por ejemplo los comandos BEGIN y END. Éstas nos permiten incluir acciones antes y después de la acción a realizar.

Analice las siguientes comandas (la segunda comanda corresponde a dos líneas que son un único comando). ¿Qué hacen estas comandas?

```
ls gutenberg/etext00/ | grep -F '.txt'

ls -l gutenberg/etext00/ | grep -F '.txt' | awk 'BEGIN {print "Fitxers
.txt:"} {sum+=$5; print $9} END {print "Total bytes: " sum}'
```

Ejercicio 3

El comando `ps aux` nos devuelve toda la información sobre los procesos que se están ejecutando en el sistema. Se pide leer y revisar el manual de este comando (mediante comando `man`), y responder a las siguientes preguntas:

1. ¿Qué diferencia existe entre los comandos `ps` y `ps aux`?
2. Muestra todos los procesos ejecutados únicamente por un usuario dado (p.ej. root). Se requiere utilizar una tubería, evitando el uso de bucles.
3. Muestra todos los procesos ejecutados por vuestro usuario. Puede ayudarse del comando `whoami` que devuelve el nombre de vuestro usuario. Se requiere utilizar una tubería, evitando el uso de bucles.

3 EJERCICIO A ENTREGAR

Se propone a continuación un ejercicio que forma parte de los problemas evaluables (PA) de la asignatura. Los detalles de la evaluación se indican al final.

El problema consiste en realizar un script para ayudar a inspeccionar los *logs* de una aplicación web. Estos logs se encuentran el directorio *tp3-logs* (campus virtual). Están separados en ficheros diferentes dependiendo del componentes de la aplicación *api, static* y *video,* que los ha generado

Cada fichero de logs tiene el siguiente formato (separados por 1 espacio):

<timestamp> <método http> <código respuesta http> <ruta> <ip cliente>

e.g:

2025-02-26T17:01:49Z GET 200 /api/user/stats/ 79.146.59.82
2025-02-27T15:06:09Z GET 200 /static/manifest/site.webmanifest 83.44.11.165
2025-02-27T15:10:41Z GET 404 /videos/vod/chnnel2/chunk5.ts 203.0.111.46

El objetivo del script es poder inspeccionar y filtrar los logs de la aplicación por componente y por el código de respuesta. El resultado será mostrar por pantalla las líneas de logs que cumplan las 2 condiciones.

El script debe aceptar los siguientes parámetros de entrada:

./tp3.sh <directorio_logs> <nombre_aplicación> <código respuesta http>

- <directorio_logs>: ruta al directorio donde están los logs
- <nombre_aplicación>: el nombre de la aplicación que queremos inspeccionar (api, static o video)
- <código respuesta aplicación>: el código http por el que queremos filtrar (200, 301, 404)

Ejemplos de uso:

$ ./tp3.sh tp3-files api 200

2025-02-26T17:01:49Z OPTIONS 200 /api/user/stats/ 79.146.59.82
2025-02-26T17:01:49Z GET 200 /api/user/stats/ 83.56.121.35
2025-02-26T17:01:49Z GET 200 /api/billing/items/ 83.59.44.99
...

$ ./tp3.sh tp3-files api 404

2025-02-26T17:01:50Z GET 404 /api/billing/items/purposes/es/ 83.59.44.99
2025-02-26T17:01:51Z GET 404 /api/user/stats/ 176.80.96.235
2025-02-26T17:01:52Z GET 404 /api/user/stats/ 31.221.254.199
...

Y así con todas las combinaciones posibles de componente y código de respuesta.

Para implementar este ejercicio **solo se permiten usar las funciones descritas en el tutorial 1 y 2**, con las opciones disponibles de las comandas, aunque no estén descritas en el tutorial.

El ejercicio debe realizar las siguientes comprobaciones:

- Comprobar que el número de parámetros sea correcto. Si no son correctos, el programa deberá salir con un código de error.
- Comprobar que el primer parámetro pasado como argumento existe y es un directorio. En caso de que no exista o no sea un directorio, el programa deberá salir con un código de error.
- Comprobar que el segundo parámetro és uno de los componentes de la aplicación (api, static o video). En caso contrario salir con un código de error.

Si la ejecución se ha realizado de forma correcta, el script debe salir con un código de ejecución correcta (típicamente es un 0).

4 PROCESO DE EVALUACIÓN

Se pide entregar mediante la tarea del campus virtual el ejercicio propuesto en la sección 3. Se realizará una única entrega por cada grupo de laboratorio. Cada pareja tiene un número asignado, que puede consultar en el campus virtual. El nombre de archivo deberá indicar sólo la pareja que entrega: así, por ejemplo, **el grupo 31** entregará un único archivo llamado **grupo31.sh**.

La fecha límite de entrega es mañana jueves hasta las 15h. A las 18h se harán disponibles todas las entregas en el campus virtual. Es muy importante que haga las entregas con el nombre de archivo correcto para evitar retrasos para hacer disponibles las entregas.

A continuación, **cada grupo evaluará dos ejercicios entregados por l@s compañer@s** que tengan un número de grupo inmediatamente inferior al que tengan asignados ell@s. Es decir, el grupo 31 tendrá que evaluar pues los grupos 30 y 29. Si el 30 no ha entregado, el grupo 31 evaluará el 29 y 28. El grupo 02 evaluará el grupo 01 y el que tenga el número más grande (es decir, la numeración es circular).

La evaluación se utilizará utilizando la **rúbrica** disponible en el campus. Utiliza los listados de archivos proporcionados para asegurar el buen funcionamiento del ejercicio entregado. Evaluar cada ejercicio no debería llevar más de 15 minutos. **Cada grupo entregará sus dos rúbricas** vía campus virtual. Para facilitar la revisión de las rúbricas se pide que cada rubrica tenga el nombre **rubrica_grupoXX_grupoYY.xlsx**, donde XX es el grupo que ha evaluado y YY corresponde a su grupo.

La fecha plazo para entregar las dos rúbricas asignadas es el viernes (23:59). La evaluación asignada ayudará a decidir qué soluciones interesantes se muestran en la próxima sesión de problemas.

La evaluación de este ejercicio sólo tendrá los valores de 10, 5 y 0. El ejercicio debe intentar resolver el problema propuesto utilizando sólo las instrucciones presentadas en el primer tutorial (a menos que se indique lo contrario). A la hora de evaluar no se penalizará entregar un ejercicio que no funcione. En cambio, en caso de que el código de un grupo no funcione y la rúbrica correspondiente de quien lo ha evaluado no lo refleje, este hecho quedará reflejado en la nota. Por último, **en caso de que no se entregue el ejercicio o la rúbrica, o la solución entregada**

utilice instrucciones no permitidas o la solución no tenga nada que ver con el ejercicio propuesto la evaluación será de cero.