

[MPWAR] - Principios y herramientas de desarrollo

Entregable 4 – Linux & Git

Parte I: Ejercicios Linux

1. Utiliza el siguiente comando para descargar el fichero que vamos a necesitar para realizar el ejercicio:

```
curl -o nginx_logs_examples.log  
https://raw.githubusercontent.com/elastic/examples/master/Common%20Data%20Formats/nginx\_logs/nginx\_logs
```

Este fichero contiene ejemplos de logs de acceso a NGINX. Con la ayuda de los comandos vistos en clase, se requiere:

- Ordenar las IPs (en orden decreciente)
- Buscar el número total ocurrencias por cada una de las IPs

El output tras la ejecución del comando debería ser el siguiente:

```
80.91.33.133 -> 1202  
93.180.71.3 -> 30  
...  
...
```

Tras la ejecución del comando, guardar la salida en un fichero nginx_requests_total.txt

Comentarios:

- **Se debería conseguir el resultado esperado con la ejecución de una única sentencia. Recordad, podéis utilizar command 1 | command 2 | command 3**
- **Podéis utilizar comandos que no hayamos visto en clase**

2. Crea un script que cumpla con los siguientes requisitos:

Comprueba la existencia del siguiente directorio:

```
/backup/<student_name>/<year>/<month>/<day>
```

Si no existe el directorio, se deberá crear. Este directorio contendrá una copia del fichero de logs del ejercicio anterior siguiendo el siguiente formato:

nginx_log_requests_<date>.log donde la fecha vendrá definida como YYYYMMDD (año-mes-día).

Por ejemplo:

nginx_log_requests_20210125.log

Si la fecha de ejecución del script se realiza el último día de la semana, además del cumplir el requisito anterior, se deberá guardar en un fichero comprimido tar.gz el resultado de las ejecuciones de todos los días de la semana (solo la semana vigente).

Ejemplo:

nginx_logs_20210131.tar.gz

- nginx_log_requests_20210125.log
- nginx_log_requests_20210126.log
- nginx_log_requests_20210127.log
- nginx_log_requests_20210128.log
- nginx_log_requests_20210129.log
- nginx_log_requests_20210130.log
- nginx_log_requests_20210131.log

3. Utilizando el script del ejercicio anterior, crea un crontab con los siguientes requerimientos:

- El script se deberá ejecutar exclusivamente de lunes a domingo
- La hora de ejecución será 23.59 hrs
- Todos los meses del año

Parte II: Ejercicios Git

1. Crea un nuevo proyecto e inicializa git
2. Añade una regla para ignorar los ficheros de tipo .sh
3. Crea dos ficheros vacíos: demo1.txt y demo2.txt. Añade estos ficheros al stage area.
4. Añade contenido al fichero demo1.txt. A continuación, realiza un commit con el mensaje "Modified demo1.txt"
5. Añade contenido al fichero demo2. A continuación, realiza un commit con el mensaje "Modified demo3.txt"
6. Rectifica el commit anterior con el mensaje "Modified demo2.txt"
7. Crea una nueva rama "develop" y añade un fichero script.sh que imprima por pantalla "Git 101"
8. Da permisos de ejecución al script
9. Realiza un merge (develop -> master) para integrar los cambios
10. Sube todos los cambios a tu repositorio remoto

Ejecutar el siguiente comando e inclúyelo en tu repositorio para ver que todos los cambios solicitados se han realizado:

```
git log --oneline > verification.log
```

Indica todos los comandos que has utilizado para llevar a cabo cada uno de los puntos.

ENTREGA: Crear un proyecto en vuestro repositorio de código (Github, Gilab, ...) llamado **linux-git-exercises**. Dentro de ese proyecto, crear una carpeta para la cuarta entrega, por ejemplo **hw-04**.

Dentro de esta carpeta, además de los ficheros utilizados, deberá existir un fichero para la primera parte y otro para la segunda.

En el enlace de entrega del campus, entregar un nota (notes_hw_4.txt) con el enlace del repositorio y el código del commit.