

Simulacion Examen: Permisos y Comandos de Linux

1. Estructura de Archivos y Directorios

carpeta raíz llamada `examFinalAvanzado` con la siguiente jerarquía (nomenclatura CamelCase)

```
examFinalAvanzado
├── privateFiles
│   ├── credentials.txt
│   └── configMain.cfg
├── publicZone
│   ├── readMeInfo.md
│   └── runPublic.sh
├── sharedGroup
│   ├── groupDoc.txt
│   └── sharedScript.sh
├── logFiles
│   ├── mainLog.log
│   └── dailyLog.log
└── scriptCore
    ├── cleanSystem.sh
    └── findData.sh
```

1. `privateFiles`: Contiene archivos confidenciales.
2. `publicZone`: Carpetas y archivos de acceso público.
3. `sharedGroup`: Directorio donde varios usuarios con un mismo grupo comparten archivos.
4. `logFiles`: Destinado al guardado de logs.
5. `scriptCore`: Contiene scripts de administración.

El archivo `examFinalAvanzado.tar.gz` lo descomprimen con:

```
tar -xzvf examFinalAvanzado.tar.gz
```

2. Instrucciones Generales

- **Sin `sudo`**: Resolver todo con el usuario normal.
- **Documentar** todo en `solutionFinal.txt` dentro de la carpeta `examFinalAvanzado`.
- **Cada archivo debe contener datos** (texto, información de sistema, resultados de comandos, etc.).
- **Crear directorios y navegar** entre ellos según se solicite.
- **Al finalizar**, empaquetar todo nuevamente con `tar -czvf`.

3. Tareas y Requisitos

Tarea 1: Restaurar Permisos Iniciales y Comandos de Navegación

1. Al descomprimir, verifica que no tienes ningún permiso (`ls -l`).
2. Usa **comandos de navegación** (`pwd`, `cd`, `ls`) para situarte en la carpeta `examFinalAvanzado`.
3. Asigna **permisos básicos** para que puedas acceder y **crear o editar** cada archivo y directorio.
 - Registra en `solutionFinal.txt` qué comandos ejecutaste.
4. **Agrega contenido** en cada archivo utilizando los comandos que prefieras:
 - Puedes guardar la salida de `pwd` o `ls` en algún archivo, por ejemplo:

```
pwd > credentials.txt
ls -la >> configMain.cfg
```

- O escribir líneas manualmente, por ejemplo:

```
echo "Información confidencial" >> credentials.txt
```

Tarea 2: Permisos y Contenido en `privateFiles`

- **Carpeta:** `privateFiles`
 - **Archivos:** `credentials.txt`, `configMain.cfg`
- **Requisitos:**
 1. **Solo** el dueño (estudiante) puede leer y escribir.
 2. Nadie más puede ni siquiera leer el contenido.
 3. **Rellenar:**
 - `credentials.txt`: Almacenar **al menos** dos líneas con salida de un comando, por ejemplo `whoami` o `uname -r`.
 - `configMain.cfg`: Añadir información de configuración (ficticia o real). ejm:

```
# Configuración Principal
PORT=8080
MODE=production
```

- **Documenta** en `solutionFinal.txt`:
 - Los permisos octales elegidos (ej. `chmod 600` o `chmod 700`).
 - Por qué son apropiados.

Tarea 3: Directorio Público

- **Carpeta:** `publicZone`
 - **Archivos:** `readMeInfo.md`, `runPublic.sh`
- **Requisitos:**
 1. Cualquier usuario puede crear, leer y escribir archivos dentro de `publicZone`.

2. El script `runPublic.sh` debe ser **ejecutable por todos**.

- **Rellenar:**

- `readMeInfo.md`: Al menos 2 líneas explicando su uso.
- `runPublic.sh`: Pequeño script de ejemplo, p. ej.:

```
#!/bin/bash
echo "Hola, este script es público, mi nombre es
$nombreEstudiante ."
```

- **Entrega:**

- Explica los permisos octales y el uso del sticky bit en `solutionFinal`.
-

Tarea 4: Directorio de Grupo

- **Carpeta:** `sharedGroup`

- **Archivos:** `groupDoc.txt`, `sharedScript.sh`

- **Requisitos:**

1. Asigna un grupo (por ejemplo, `teamWork`) a esta carpeta y sus archivos, para que los miembros de ese grupo tengan lectura y escritura.
2. El propietario (tú) puede leer, escribir y ejecutar `sharedScript.sh`.
3. Otros miembros del grupo pueden leer y escribir `sharedScript.sh`, pero no ejecutarlo.

- **Rellenar:**

- `groupDoc.txt`: al menos 2 líneas describiendo el trabajo en grupo.
- `sharedScript.sh`: un script sencillo:

```
#!/bin/bash
echo "Script compartido para el grupo teamWork"
```

- **Entrega:**

- Explica los permisos octales en `solutionFinal`.
-

Tarea 5: Administración de Logs

- **Carpeta:** `logFiles`

- **Archivos:** `mainLog.log`, `dailyLog.log`

- **Requisitos:**

1. Solo el propietario puede leer y escribir estos archivos
2. Nadie más puede acceder.

- **Rellenar:**

- mainLog.log: al menos 2 líneas con, por ejemplo, la salida de date o eventos ficticios:

```
[INFO] Sistema iniciado  
[WARN] Lectura de sensor tardía
```

- dailyLog.log: 1 línea con la salida de date.

- **Entrega:**

- Describe en solutionFinal.txt las decisiones de permisos.
-

Tarea 6: Scripts de Administración

- **Carpeta:** scriptCore

- **Archivos:** cleanSystem.sh, findData.sh

- **Requisitos:**

1. cleanSystem.sh:
 - Ejecutable solo por el propietario.
 - Nadie más puede leer ni escribir en él.
2. findData.sh:
 - Ejecutable por el propietario y el grupo.
 - Otros usuarios pueden leer el archivo, pero no ejecutarlo.

- **Rellenar:**

- cleanSystem.sh: un script que muestre un mensaje, p. ej.:

```
#!/bin/bash  
echo "Limpiando /tmp y procesos inactivos"
```

- findData.sh: un script que busque ficheros o muestre algo, p. ej.:

```
#!/bin/bash  
echo "Buscando data en examFinalAvanzado..."  
ls examFinalAvanzado
```

- **Entrega:**

- Explica en solutionFinal.txt las combinaciones octales aplicadas.

4. Entrega Final

1. **Documenta** todos los pasos en `solutionFinal.txt` dentro de la carpeta `examFinalAvanzado`.
2. **Empaqueta** la carpeta con todos los cambios y tu archivo `solutionFinal.txt`:

```
tar -czvf nombreApellido.tar.gz examFinalAvanzado/
```