

## LPIC-2 / Examen 206 - Mantenimiento del Sistema - Ejercicios

*Nota: Estos ejercicios implican crear archivos y manipular datos. Realízalos en una VM de prueba con datos no críticos. Necesitarás privilegios de superusuario (sudo) para algunos pasos.*

### Ejercicio 6.1.1: Creando un Backup Completo con tar y Compresión

- **Objetivo:** Usar `tar` para empaquetar y comprimir un directorio.
- **Requisitos:** Acceso a la línea de comandos. Un directorio con archivos para respaldar (ej: tu directorio personal `~` o un directorio de prueba). Espacio libre para el archivo de backup.
- **Desarrollo Paso a Paso:**
  1. Abre una terminal.
  2. **Crea un directorio de prueba y algunos archivos:** Ejecuta `mkdir -p ~/test_backup/data` && `echo "Archivo 1" > ~/test_backup/data/file1.txt` && `echo "Archivo 2" > ~/test_backup/data/file2.txt`.
  3. **Crea un archivo .tar.gz (comprimido con gzip) del directorio de prueba:** Ejecuta `tar -czvf ~/test_backup/backup_data.tar.gz ~/test_backup/data`.
    - `-c`: Crear archivo.
    - `-z`: Comprimir con gzip.
    - `-v`: Mostrar archivos procesados.
    - `-f ~/test_backup/backup_data.tar.gz`: Archivo de salida.
    - `~/test_backup/data`: Directorio a respaldar.
  4. **Verifica que el archivo de backup fue creado:** Ejecuta `ls -lh ~/test_backup/`.

### Ejercicio 6.1.2: Listando y Restaurando desde un Archivo tar

- **Objetivo:** Ver el contenido de un archivo `tar` y extraer archivos de él.
- **Requisitos:** El archivo de backup creado en Ej. 6.1.1.
- **Desarrollo Paso a Paso:**
  1. Abre una terminal.
  2. **Lista el contenido del archivo de backup:** Ejecuta `tar -tzvf ~/test_backup/backup_data.tar.gz`. La `t` lista en lugar de extraer.
  3. **Crea un directorio para la restauración:** Ejecuta `mkdir ~/restore_test`.
  4. **Restaura el contenido a un directorio específico:** Ejecuta `tar -xzf ~/test_backup/backup_data.tar.gz -C ~/restore_test`.
    - `-x`: Extraer.
    - `-z`: Descomprimir con gzip.
    - `-v`: Mostrar archivos procesados.
    - `-f ~/test_backup/backup_data.tar.gz`: Archivo de entrada.
    - `-C ~/restore_test`: Extraer en este directorio.

5. **Verifica que los archivos fueron restaurados:** Ejecuta `ls ~/restore_test/test_backup/data/`. Visualiza su contenido con `cat`.
6. **Limpia:** Ejecuta `rm -rf ~/test_backup/ ~/restore_test/`.

### Ejercicio 6.1.3: (Conceptual) Copiando con `dd`

- **Objetivo:** Entender cómo se usa `dd` para copias a nivel de bloque.
- **Requisitos:** Acceso a la línea de comandos. **VM de prueba. Nunca uses `dd` en discos con datos importantes sin estar absolutamente seguro.**
- **Desarrollo Paso a Paso:**
  1. Abre una terminal.
  2. **Comprende el comando para copiar un archivo a otro archivo:** `dd if=origen of=destino [bs=tamaño_bloque] [count=numero_bloques]`. Esto es similar a `cp` pero a nivel de bloque.
  3. **Comprende el comando para crear una imagen de una partición (¡PELIGROSO!):** `sudo dd if=/dev/sda1 of=~/backup_sda1.img bs=4M`. Esto copiaría la partición `/dev/sda1` a un archivo.
  4. **Comprende el comando para restaurar una imagen a una partición (¡MUY PELIGROSO! Sobrescribe el destino):** `sudo dd if=~/backup_sda1.img of=/dev/sda1 bs=4M`. ¡Cuidado! El destino debe ser del mismo tamaño o mayor.
  5. **Comprende copiar un disco a otro disco (¡EXTREMADAMENTE PELIGROSO!):** `sudo dd if=/dev/sda of=/dev/sdb bs=4M`.  
**¡SOBRESCRITOS LOS DATOS EN /dev/sdb!**

### Ejercicio 6.1.4: Realizando un Backup Incremental Local con `rsync`

- **Objetivo:** Usar `rsync` para sincronizar un directorio y ver cómo maneja los cambios.
- **Requisitos:** Acceso a la línea de comandos. El paquete `rsync` instalado (`sudo apt install rsync` o `sudo dnf install rsync`).
- **Desarrollo Paso a Paso:**
  1. Abre una terminal.
  2. **Crea un directorio de origen y uno de destino vacío:** Ejecuta `mkdir -p ~/rsync_src && mkdir -p ~/rsync_dest`.
  3. **Crea algunos archivos iniciales en el origen:** Ejecuta `echo "Inicial 1" > ~/rsync_src/file1.txt && echo "Inicial 2" > ~/rsync_src/file2.txt`.
  4. **Realiza el primer backup (copia completa):** Ejecuta `rsync -avh ~/rsync_src/ ~/rsync_dest/`. Observa que copia ambos archivos. La barra inclinada `/` al final del origen es importante (`~/rsync_src/` significa copiar el contenido del directorio, `~/rsync_src` sin `/` significaría copiar el propio directorio `rsync_src`).

5. **Modifica un archivo en el origen y añade uno nuevo:** Ejecuta `echo "Modificado" > ~/rsync_src/file1.txt && echo "Nuevo archivo" > ~/rsync_src/file3.txt`.
6. **Realiza el segundo backup (incremental):** Ejecuta `rsync -avh ~/rsync_src/ ~/rsync_dest/`. Observa que solo transfiere `file1.txt` (porque cambió) y `file3.txt` (porque es nuevo). No transfiere `file2.txt`.
7. **Elimina un archivo en el origen y vuelve a respaldar (usando `--delete`):** Ejecuta `rm ~/rsync_src/file2.txt`. Ejecuta `rsync -avh --delete ~/rsync_src/ ~/rsync_dest/`. Observa que transfiere `file1.txt`, `file3.txt` y elimina `file2.txt` en el destino.
8. **Limpia:** Ejecuta `rm -rf ~/rsync_src/ ~/rsync_dest/`.

### Ejercicio 6.1.5: (Conceptual) Programando Backups con Cron o Systemd Timers

- **Objetivo:** Entender cómo programar la ejecución de un script de backup.
- **Requisitos:** Acceso a la línea de comandos. Privilegios de superusuario (`sudo`) para cron de sistema.
- **Desarrollo Paso a Paso:**

1. Abre una terminal.

2. **Comprende la programación con Cron:**

- Crea un script de backup (ej: `~/backup_script.sh`) que contenga los comandos `tar`, `rsync`, etc. Asegúrate de que sea ejecutable (`chmod +x ~/backup_script.sh`).
- Para un usuario: Ejecuta `crontab -e` y añade una línea como `0 2 * * * /home/tu_usuario/backup_script.sh`. Esto ejecutará el script a las 2:00 AM todos los días.
- Para el sistema (como root): Coloca el script en `/etc/cron.daily/` (se ejecuta diariamente) o crea un archivo en `/etc/cron.d/` (ej: `sudo vi /etc/cron.d/my_backup`) con contenido como:

```
0 3 * * * root /opt/scripts/backup_system.sh
```

Esto ejecuta `/opt/scripts/backup_system.sh` como usuario `root` a las 3:00 AM todos los días.

3. **Comprende la programación con Systemd Timers:**

- Necesitas crear dos archivos de unidad: un archivo `.service` que define el script de backup a ejecutar y un archivo `.timer` que define cuándo ejecutar el servicio.
- Archivo `.service` (ej: `/etc/systemd/system/my-backup.service`):

```
[Unit]
Description=My Daily Backup
```

```
[Service]
Type=oneshot
ExecStart=/opt/scripts/backup_system.sh
```

- Archivo `.timer` (ej: `/etc/systemd/system/my-backup.timer`):

```
[Unit]
Description=Runs my daily backup script

[Timer]
OnCalendar=*-*-* 3:00:00 # Ejecutar a las 3:00 AM todos los días
# Unit=my-backup.service # Opcional, por defecto ejecuta un servicio con el mismo nombre

[Install]
WantedBy=timers.target
```

- Después de crear los archivos, habilita y arranca el timer: `sudo systemctl enable my-backup.timer && sudo systemctl start my-backup.timer`. Puedes ver el estado con `systemctl list-timers`.