

## LPIC-2 / Examen 204 - Administración Avanzada de Dispositivos de Almacenamiento - Ejercicios

*Nota: Estos ejercicios implican verificar y modificar configuraciones de dispositivos. **Realiza cambios solo en dispositivos de prueba (loopback o discos virtuales dedicados) en una VM.** Necesitarás privilegios de superusuario (sudo).*

### Ejercicio 4.2.1: Identificando Planificadores de E/S Disponibles y Activos

- **Objetivo:** Determinar qué planificadores están disponibles en tu kernel y cuál se usa para un disco específico.
- **Requisitos:** Acceso a la línea de comandos. Un dispositivo de bloque para verificar (ej: /dev/sda, o tu dispositivo loop de prueba).
- **Desarrollo Paso a Paso:**
  1. Abre una terminal.
  2. **Identifica un dispositivo de bloque:** Ejecuta `lsblk`. Anota el nombre de tu disco principal (ej: sda) o de un disco de prueba (ej: sdb).
  3. **Verifica el planificador activo y los disponibles:** Ejecuta `cat /sys/block/<nombre_dispositivo>/queue/scheduler` (ej: `cat /sys/block/sda/queue/scheduler`). La salida mostrará los nombres de los planificadores disponibles, con el activo entre corchetes [ ].

### Ejercicio 4.2.2: Cambiando el Planificador de E/S Temporalmente

- **Objetivo:** Modificar el planificador de E/S para un dispositivo hasta el próximo reinicio.
- **Requisitos:** Privilegios de superusuario (sudo). Un dispositivo de prueba (loopback o disco virtual dedicado, **NO tu disco principal**). Identifica los planificadores disponibles (Ej. 4.2.1).
- **Desarrollo Paso a Paso:**
  1. Abre una terminal.
  2. **Identifica tu dispositivo de prueba (desmontado):** Ejecuta `lsblk`. Anota el nombre (ej: /dev/loop0 o /dev/sdb). Asegúrate de que no está montado.
  3. **Identifica un planificador disponible diferente al actual:** Consulta la salida del Ej. 4.2.1.
  4. **Cambia el planificador:** Ejecuta `sudo echo <nuevo_planificador> > /sys/block/<nombre_dispositivo_prueba>/queue/scheduler` (ej: `sudo echo noop > /sys/block/loop0/queue/scheduler`).
  5. **Verifica el cambio:** Ejecuta `cat /sys/block/<nombre_dispositivo_prueba>/queue/scheduler`. El nuevo planificador debería estar entre corchetes.
  6. **(Opcional) Monta y genera algo de carga de E/S en el dispositivo de prueba y observa el rendimiento (conceptual):** Puedes montar el dispositivo, generar E/S (ej: dd) y monitorizar con `iostat -x` o `iotop`. Luego cambia a otro planificador y

repite para comparar (esto requiere tiempo y una metodología de prueba controlada para ser significativo).

7. **(El cambio es temporal):** Este cambio se perderá al reiniciar.

#### Ejercicio 4.2.3: Verificando Opciones de Montaje Relacionadas con Rendimiento

- **Objetivo:** Identificar opciones de montaje que impactan el rendimiento.
- **Requisitos:** Acceso a la línea de comandos.
- **Desarrollo Paso a Paso:**
  1. Abre una terminal.
  2. **Muestra todos los montajes y sus opciones:** Ejecuta `findmnt`. Observa la columna `OPTIONS`.
  3. **Busca opciones relevantes para rendimiento/seguridad:**
    - `noatime`, `relatime`, `atime`: Relacionadas con la actualización del tiempo de acceso. `noatime` ofrece el mejor rendimiento, `relatime` es un buen compromiso.
    - `data=ordered`, `data=journal`, `data=writeback`: Para sistemas `ext4`. `writeback` es el más rápido pero menos seguro en caso de fallo.
    - `barrier=1`, `barrier=0`: Para `ext4/XFS`. `0` puede ser más rápido pero riesgoso.
  4. **Visualiza `/etc/fstab`:** Ejecuta `cat /etc/fstab`. Verifica las opciones definidas aquí, ya que son persistentes.

#### Ejercicio 4.2.4: Verificando Tiempos de Acceso a Archivos (`stat`)

- **Objetivo:** Ver cómo las opciones `atime/noatime/relatime` impactan en los metadatos del archivo.
- **Requisitos:** Acceso a la línea de comandos. Un archivo de prueba.
- **Desarrollo Paso a Paso:**
  1. Abre una terminal.
  2. **Crea un archivo de prueba:** Ejecuta `touch testfile.txt`.
  3. **Muestra la información del archivo, incluyendo tiempos:** Ejecuta `stat testfile.txt`. Observa las líneas "Access", "Modify", "Change". Access es `atime`, Modify es `mtime`, Change es `ctime`.
  4. **Espera unos segundos (para sistemas con poca precisión en `atime`).**
  5. **Lee el archivo de prueba:** Ejecuta `cat testfile.txt`.
  6. **Muestra la información del archivo de nuevo:** Ejecuta `stat testfile.txt`.
  7. **Compara los tiempos de acceso:** Si el sistema de archivos donde se encuentra el archivo está montado con `atime` completo, el tiempo de acceso debería haberse actualizado a la hora de `cat`. Si está montado con `relatime`, podría no haberse actualizado si `mtime` no cambió. Si está montado con `noatime`, el tiempo de acceso no debería haberse actualizado.

#### Ejercicio 4.2.5: (Conceptual) Configuración Persistente del Planificador de E/S con Udev

- **Objetivo:** Entender cómo usar reglas de udev para establecer el planificador de E/S para un dispositivo específico de forma permanente.
- **Requisitos:** Privilegios de superusuario (sudo). Un dispositivo para configurar (ej: un disco virtual secundario /dev/sdb). Conocer los atributos del dispositivo (Ej. 2.4.1).
- **Desarrollo Paso a Paso:**
  1. Abre una terminal.
  2. **Comprende la sintaxis de la regla:** Necesitarás claves de coincidencia para identificar el dispositivo (ej: `KERNEL=="sdb"`, o una combinación de `SUBSYSTEM` y `ATTRS` usando el ID del fabricante/modelo para mayor precisión). La clave de asignación es `ATTR{queue/scheduler}=`.
  3. **Crea un archivo de reglas en /etc/udev/rules.d/:** Dale un nombre apropiado (ej: `60-schedulers.rules`). Ejecuta `sudo vi /etc/udev/rules.d/60-schedulers.rules`.
  4. **Añade la regla (ejemplo para /dev/sdb):**

```
# Establecer planificador deadline para /dev/sdb
KERNEL=="sdb", ATTR{queue/scheduler}="deadline"
```

    - O usando atributos más persistentes:

```
# Establecer planificador noop para mi SSD Crucial (ejemplo de atributos)
SUBSYSTEM=="block", ATTRS{idVendor}=="Crucial",
ATTRS{idModel}=="CT500MX500SSD1", ATTR{queue/scheduler}="noop"
```
    - Reemplaza los atributos con los de tu dispositivo si no es sdb.
  5. **Guarda y sal.**
  6. **Recarga las reglas de udev:** Ejecuta `sudo udevadm control --reload-rules`.
  7. **Dispara eventos para el subsistema block:** Ejecuta `sudo udevadm trigger --subsystem-match=block`.
  8. **Verifica el cambio:** Ejecuta `cat /sys/block/<nombre_dispositivo>/queue/scheduler`. El planificador debería haberse actualizado. Este cambio será persistente después de reiniciar.
  9. **(Limpieza en VM):** Elimina o comenta la regla si no quieres que sea permanente.

#### Ejercicio 4.2.6: (Conceptual) Configuración Persistente del Planificador vía Parámetro del Kernel

- **Objetivo:** Entender cómo usar el parámetro `levator=` en GRUB.
- **Requisitos:** Privilegios de superusuario (sudo). **VM de prueba.**
- **Desarrollo Paso a Paso:**
  1. Abre una terminal.

2. **Edita el archivo de configuración global de GRUB:** Ejecuta `sudo vi /etc/default/grub`.
3. **Busca la línea `GRUB_CMDLINE_LINUX_DEFAULT` o `GRUB_CMDLINE_LINUX`.**
4. **Añade `elevator=<nombre_planificador>` dentro de las comillas:** Por ejemplo: `GRUB_CMDLINE_LINUX_DEFAULT="quiet splash elevator=deadline"`.
5. **Guarda y sal.**
6. **Regenera el archivo `grub.cfg`:** Ejecuta el comando apropiado para tu distribución (Ej. 2.3.4): `sudo update-grub` (Debian/Ubuntu) o `sudo grub2-mkconfig -o <ruta_a_grub.cfg>` (Red Hat/Fedora).
7. **Reinicia la VM.**
8. **Después de reiniciar, verifica el planificador para tus discos:** Ejecuta `cat /sys/block/<nombre_disco>/queue/scheduler`. El planificador por defecto debería ser el que especificaste en el parámetro `elevator=`.
9. **(Limpieza en VM):** Edita `/etc/default/grub` de nuevo, elimina el parámetro `elevator=`, regenera `grub.cfg` y reinicia.