Ingeniería de Servidores

grado en ingeniería informática

Guión de prácticas resueltas

Autor
Carlos Sánchez Páez

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

Curso 2019-2020

Índice

Índice de figuras

1. Práctica 1: Virtualización e instalación de Sistemas Operativos

1.1. Sesión 1

1.1.1. Tipos de arquitecturas

Un servidor es una máquina que se dedica a resolver peticiones. Hay varios tipos:

- **Hosting dedicado**. Alguien monta su propio servidor y únicamente lo utiliza él.
- **VPS (*Virtual Private Server*)**. Se utiliza la virtualización para proporcionar recursos dedicados (privados) al cliente a partir de un servidor con múltiples usuarios.
- ***Serverless***. Se necesita un proveedor cloud (*AWS, Azure, Google Cloud...*), que gestiona dinámicamente los recursos. Las aplicaciones se ejecutan al detectarse determinados eventos. Se cobra por ancho de banda utilizado, capacidad de disco duro, etc. La principal ventaja de esta arquitectura es la escalabilidad.

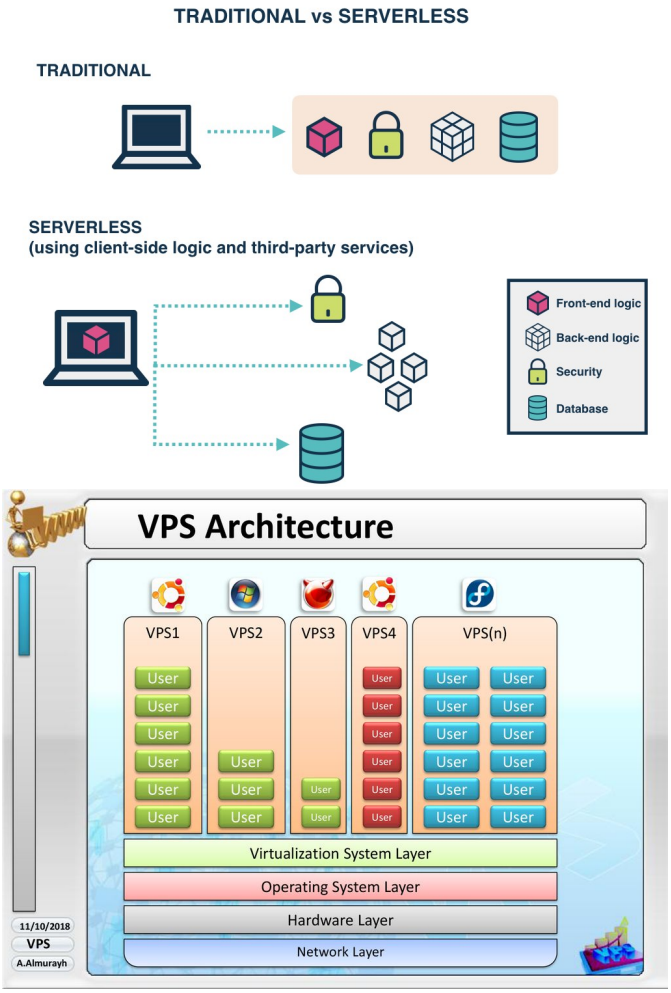


Figura 1: Tipos de arquitecturas de servidor

Una **máquina virtual** es aquella en la que todo su *hardware* está virtualizado, es decir, compartido con el host. Las principales ventajas de las MV son precio, encapsulamiento y flexibilidad. Realmente no son más que archivos.

Un **contenedor** empaqueta una aplicación con sus correspondientes dependencias, consiguiendo así la máxima *portabilidad*. Lo único que se necesita tener instalado en una máquina para ejecutar la aplicación es necesario contar con el hipervisor adecuado (por ejemplo, *Docker*).

Un **hipervisor** es un motor que se encarga de traducir las instrucciones de una máquina virtual (o contenedor) a llamadas al sistema. Algunos ejemplos de hipervisor son *VirtualBox* o *VMWare*.

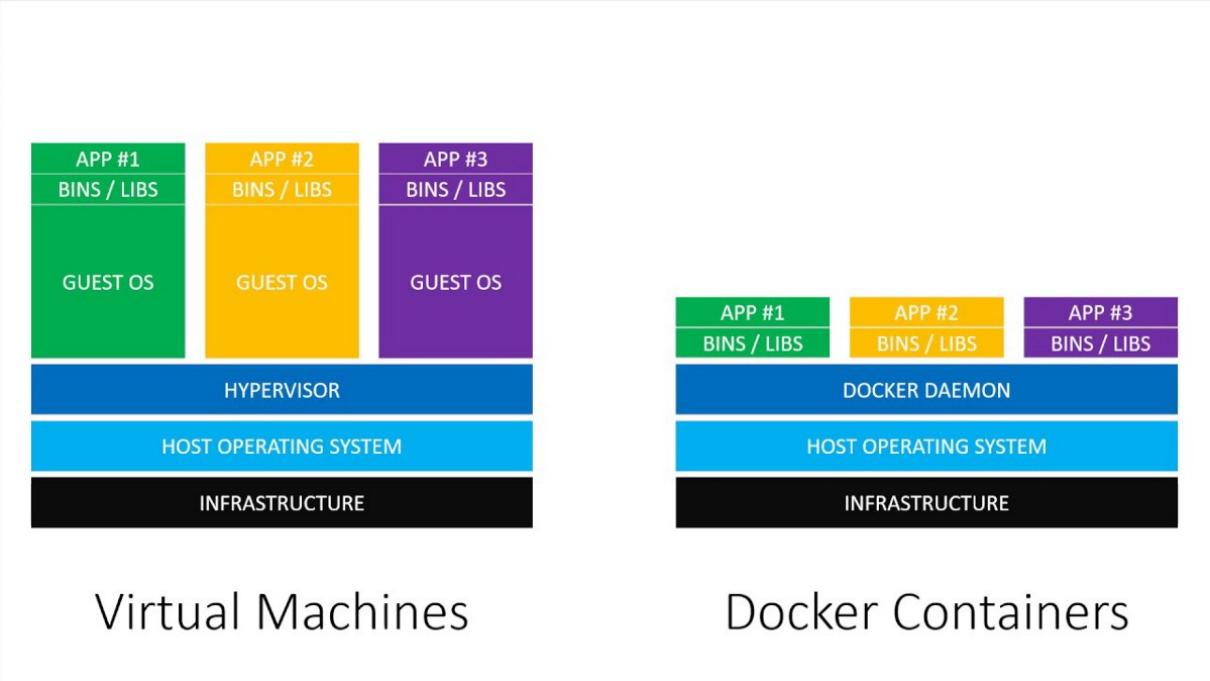


Figura 2: Diferencias entre contenedor y máquina virtual

Podríamos realizar una analogía diciendo que las máquinas virtuales son procesos (encapsulados, no pueden acceder entre ellos) y los contenedores son hebras (sí que pueden comunicarse).

1.1.2. RAID

RAID (***R**edundant **A**rray of **I**ndependent/**I**nexpensive* **D**isks) es una tecnología que utiliza varias unidades de almacenamiento entre las que se replican los datos. Existen varios tipos de RAID:

- **RAID0**. En este modelo no hay réplica. Los datos se distribuyen equitativamente entre ambos volúmenes.
- **RAID1**. Los datos se copian en el otro disco (espejo).
- **RAID2,3,4** no se usan en la actualidad.

- **RAID5**. Implementa bloques de paridad como medida de redundancia. Puede fallar un disco como máximo.
- **RAID6**. Implementa doble paridad. Pueden fallar dos discos como máximo.
- **RAID0+1,RAID1+0**. Se anidan ambos tipos de RAID.

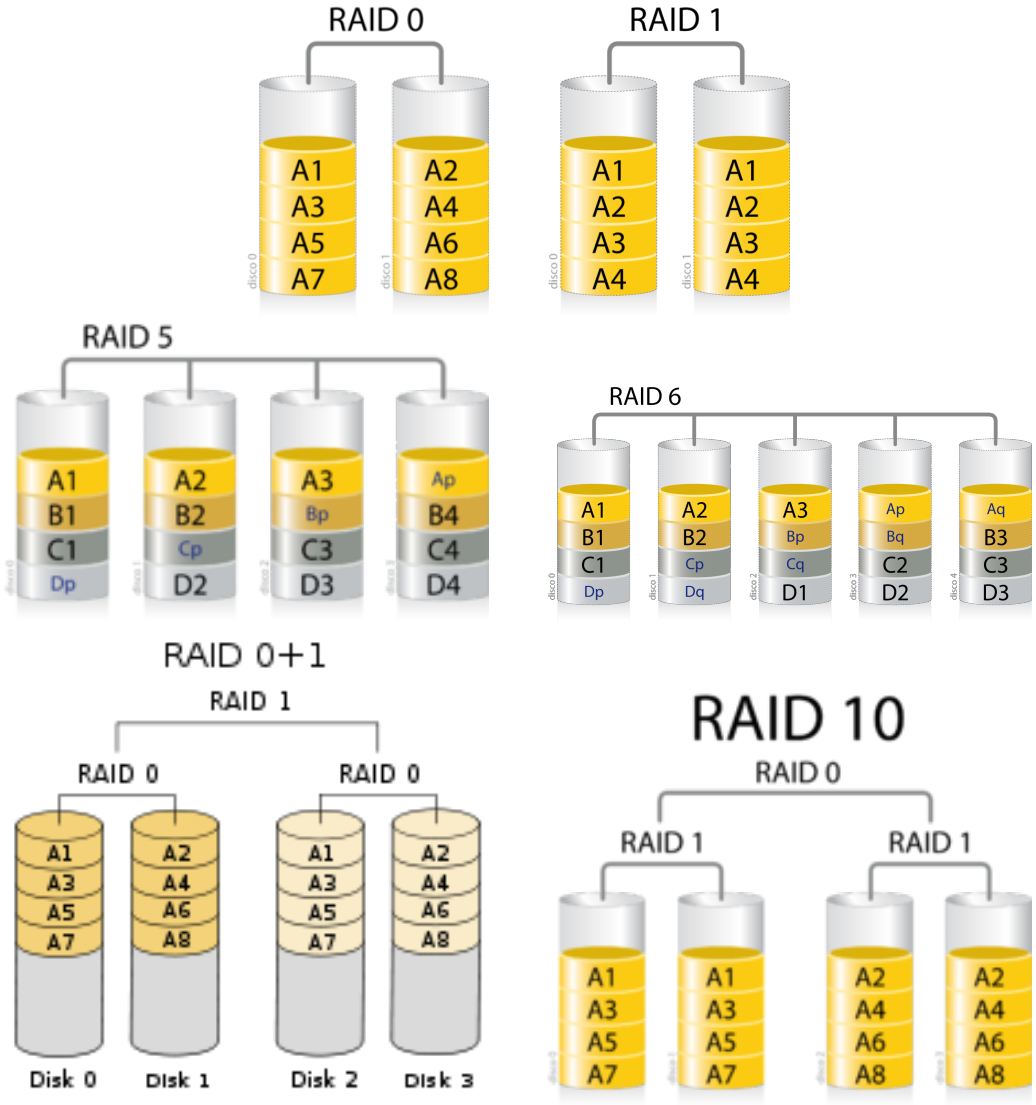


Figura 3: Tipos de RAID

1.1.3. LVM

LVM (***L**ogical **V**olume **M**anager*) provee abstracción sobre el almacenamiento físico y el sistema de ficheros. Sus mayor ventaja es la flexibilidad: podemos incorporar nuevas unidades de almacenamiento *en caliente* (sin parar el sistema) de forma sencilla. LVM está compuesto por:

- Volumen físico (*Physical Volume*, **PV**). Es un dispositivo de almacenamiento (HDD, partición, RAID...).
- Grupo de volúmenes (*Volume Group*, **VG**). Es el centro de LVM. Está formado por uno o más PV. Para aumentar su espacio sólo hay que añadir más volúmenes físicos, siendo esto transparente para los sistemas de archivos, procesos o usuarios.
- Volumen lógico (*Logical Volume*, **LV**). Es el "producto final", es decir, dispositivos que usaremos para crear sistemas de ficheros. Podríamos decir que son las particiones.

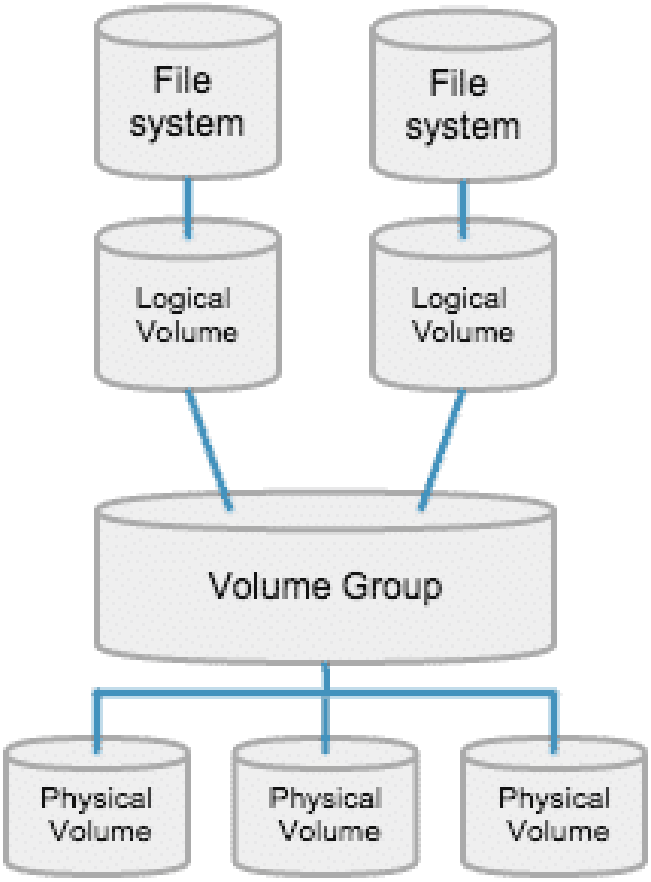


Figura 4: Arquitectura LVM

1.1.4. Instalación de Ubuntu Server con RAID1

1. Descargamos la ISO desde aquí.
2. Creamos una máquina virtual con 1024MB RAM y dos discos duros VDI reservados dinámicamente de 10GB cada uno. Por último, montamos la ISO en el controlador IDE.
3. Arrancamos la máquina, marcamos idioma español y procedemos a la instalación.
4. Elegimos España y la distribución de teclado Spanish/Spanish.
5. Dejamos el nombre de la máquina en übuntuz establecemos el nombre de usuario con nuestras iniciales. La clave será *practicas,ISE*.
6. No ciframos la carpeta personal, ya que usaremos FDE (*Full Disk Encryption*).
7. Aceptamos la zona horaria propuesta.
8. Elegimos particionado manual.
9. Damos Enter en ambos discos para crear las tablas de particiones.
10. Comenzamos configurando RAID (*Configurar RAID por software*).
11. Creamos dispositivo MD (*Multiple Devices*), elegimos RAID1, establecemos 2 discos en uso y 0 vacíos, los seleccionamos y damos a terminar.
12. Pasamos a configurar LVM (*Configurar el Gestor de Volúmenes Lógicos (LVM)*).
13. Creamos el grupo de volúmenes, con nombre *Servidor* y elegimos */dev/md0*.
14. Creamos ahora los volúmenes lógicos sobre Servidor (*swap* (1024MB),*arranque* (200MB),*hogar* (800MB) y *raíz* (resto)). Por último, damos a terminar.
15. Pasamos a la configuración del cifrado (*Configurar los volúmenes cifrados*).
16. Damos a *Create encrypted volumes* y seleccionamos todos menos *arranque* (si lo encriptamos no podremos arrancar el sistema).
17. Mantenemos los parámetros predeterminados, por lo que elegimos *Se ha terminado de definir la partición* en todos los casos.
18. Damos *Sí* a mantener la distribución existente, *Finish* y establecemos la clave *practicas,ISE* en todos los volúmenes.
19. Por último, vamos a formatear los volúmenes y asignar los puntos de montaje. Para ello, elegimos las particiones (marcadas con el símbolo #) *arranque*, *hogar* y *raíz*. Seleccionamos *Utilizar como: sistema de ficheros ext4 transaccional* y asignamos los puntos de montaje */boot*, */home* y */* respectivamente.
20. Para la partición *swap*, elegimos *Utilizar como: área de intercambio*.
21. Finalizamos el particionado y esperamos.

22. Dejamos en blanco el campo de proxy y elegimos la opción *Sin actualizaciones automáticas*, ya que queremos mantener el control total sobre el sistema.
23. Dejamos seleccionado *Standard system utilities* y pulsamos Enter.
24. Instalamos en cargador de arranque en */dev/sda*.
25. Una vez que el sistema esté instalado, iniciamos sesión y desbloqueamos los volúmenes cifrados.
26. Pasamos ahora a instalar *GRUB* en el otro disco. Para ello ejecutamos:

```
$> sudo grub install /dev/sdb
```

1.1.5. Configuración de red

Necesitamos configurar la red de forma que las máquinas puedan comunicarse entre sí,con el *host* y con el exterior. Para ello, seguiremos los siguientes pasos:

1. Crear una red sólo-anfitrión.
   a) En VirtualBox (¡no en la máquina!) damos a Archivo-Administrador de red anfitrión.
   b) Seleccionamos Crear y comprobamos que la IPv4a sea 192.168.56.1. En caso contrario, la modificamos.
   c) En la configuración de la máquina virtual Ubuntu, habilitamos el adaptador 2 (conectado a adaptador sólo-anfitrión)
2. Añadir la interfaz y activarla.
   a) Arrancamos la máquina.
   b) Editamos el archivo de interfaces mediante *sudo nano /etc/network/interfaces*.
   c) Añadimos el siguiente código:

```
# Host-only interface
auto enp0s8
iface enp0s8 inet static
address 192.168.56.105
```

   d) Salimos y guardamos.
   e) Activamos la interfaz con *sudo ifup enp0s8*.
   f) Ejecutamos *ip addr* y comprobamos que la interfaz tiene asignada la IP que configuramos (192.168.56.105).
3. Comprobar que funciona.
   a) Desde nuestro host, abrimos una terminal y ejecutamos *ping 192.168.56.105 -c 5*. Deberíamos recibir las 5 respuestas.
   b) Desde la máquina virtual ejecutamos *ping 192.168.56.1 -c 5*. Deberíamos recibir las 5 respuestas.

1.2. Sesión 2

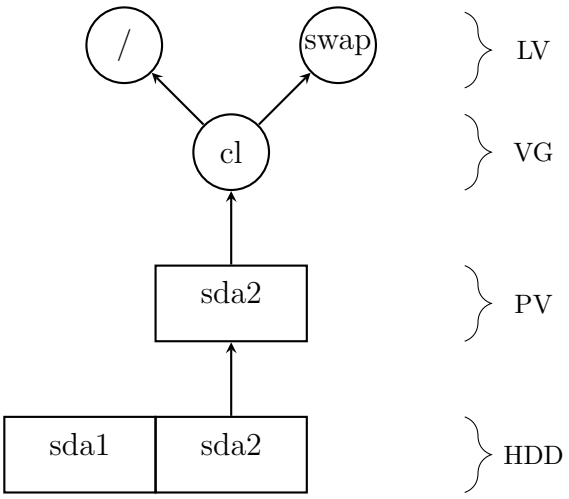
1.2.1. Instalación de CentOS

1. Descargamos la ISO desde este enlace.
2. Creamos una máquina virtual con 1024MB de RAM y un disco duro de 10GB (elegimos la opción *Fedora 64bits*). Por último, montamos la ISO.
3. Seleccionamos *Install CentOS Linux*.
4. Elegimos Español (España).
5. En *Destino de la instalación* elegimos el único disco disponible y damos a *Listo* y a *Empezar instalación*.
6. Creamos el usuario (iniciales como usuario y *practicas,ISE* como clave). Establecemos también la contraseña del *root* (*practicas,ISE*).

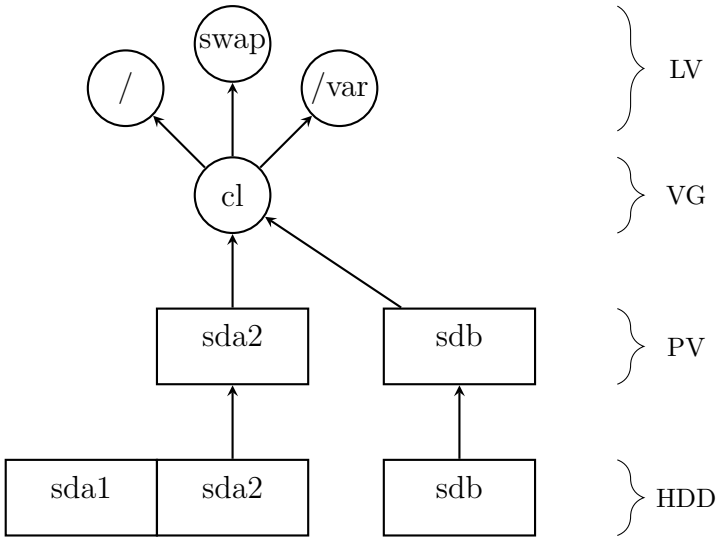
La temática de esta sesión es la siguiente: necesitamos espacio en */var* pero no tenemos suficiente. Para dar solución a ello, tendremos que seguir estos pasos:

1. Añadir el disco a la máquina.
2. Configurar el disco mediante *LVM* y darle formato.
3. Copiar los datos de */var* al nuevo volumen.
4. Indicar al SO que monte el nuevo volumen en */var*.
5. Borrar los datos antiguos de */var* (en la vida real ésto se haría un tiempo prudencial después para poder disponer de un backup.)

Básicamente, tenemos que pasar del esquema actual:



Al siguiente:



Añadir el disco a la máquina Mediante el administrador de VirtualBox añadimos un nuevo disco (VDI, 10GB, reserva dinámica) a nuestra instancia.

Configurar el disco mediante *LVM* y darle formato

1. Arrancamos la máquina.

2. Comprobamos que el disco efectivamente ha sido detectado mediante *lsblk*(**LiS**t **BL**oc**K** devices). Deberíamos ver el nuevo dispositivo */dev/sdb*. Podemos apreciar también aquí que *CentOS* usa *LVM* de forma nativa, bajo un grupo de volúmenes llamado *cl*. También podemos ver como */boot* está sobredimensionado. Esto se hace para poder tener un backup del kernel antiguo en caso de que deba actualizarse.

3. Como vamos a modificar elementos del sistema, nos logueamos como superusuario:

```
$> su
```

4. Podemos ver los detalles de *LVM* mediante los comandos *lvdisplay* (***L**ogical **V**olume Display*), *vgdisplay* (***V**olume **G**roup Display*) o *pvdisplay* (***P**hysical **V**olume Display*).

5. Creamos el volumen físico:

```
#> pvcreate /dev/sdb
```

   Podemos leer en el *man* que podemos ejecutar el comando tanto en discos duros como en particiones. Comprobamos que se ha creado mediante *pvdisplay*.

6. Ahora debemos añadir el volumen físico al grupo de volúmenes existente (*cl*). Para ello, ejecutamos

```
#> vgextend cl /dev/sdb
```

Comprobamos que ha ido bien mediante *vgdisplay*.

7. Por último, debemos crear el volumen lógico. Para ello, ejecutamos el siguiente comando:

```
#> lvcreate -L 5G -n newvar cl
```

Donde:

- -L indica el tamaño del volumen.
- -n indica el nombre del volumen.

Para finalizar, comprobamos que ha ido bien mediante *lvdisplay*.

En este momento *LVM* ha sido configurado completamente.

Copiar los datos de */var* al nuevo volumen

1. Debemos comenzar creando un sistema de archivos en nuestro volumen lógico. Pero primero debemos ver cuál es el punto de montaje del mismo mediante *lvdisplay* (campo *LV Path*).

2. Ahora ejecutamos el comando para crear el sistema de archivos:

```
#> mkfs -t ext4 /dev/cl/newvar
```

Copiar los datos de */var* al nuevo volumen

1. Lo primero que debemos hacer es montar el volumen. Para ello ejecutamos

```
#> mkdir /mnt/newvar
#> mount /dev/cl/newvar /mnt/newvar
```

Comprobamos con *mount*.

2. Ahora debemos plantearnos lo siguiente: la copia debe realizarse de forma *atómica*, es decir, nadie puede modificar ningún archivo, ya que esto generaría inconsistencias. Para ello, accedemos al modo de mantenimiento:

```
#> systemctl isolate runlevel1.target
```

Introducimos de nuevo la clave del *root*.

3. Realizamos la copia de los ficheros:

```
#> cp -a /var/. /mnt/newvar
```

Donde:

- -a indica que se deben copiar los archivos recursivamente, manteniendo enlaces y preservando el contexto. El contexto está compuesto por políticas de seguridad de *SELinux* (***S**ecurity **E**nhanced Linux*) que controlan el acceso a recursos de los procesos.
- /var/. indica todos los archivos (incluidos los ocultos).

Comprobamos mediante *ls -ahZ* sobre */var* y *mnt/newvar*

Indicarle al SO que monte el nuevo volumen en */var*

1. Abrimos el editor de texto sobre */etc/fstab*

```
#> vi /etc/fstab
```

2. Pulsamos *i* para acceder al modo de inserción y añadimos la siguiente línea:

```
/dev/mapper/cl-newvar /var ext4 defaults 0 0
```

3. Salimos de vi (ESC, escribimos *:wq* y Enter).
4. Desmontamos el nuevo volumen:

```
#> umount /mnt/newvar
```

5. Montamos los sistemas de archivos de */etc/fstab*:

```
#> mount -a
```

Comprobamos ejecutando *mount*.

Borrar los datos antiguos de */var* Actualmente no tenemos acceso al antiguo */var* (el que tenemos montado actualmente es el nuevo).

1. Comenzamos desmontando */var*:

```
#> umount /dev/mapper/cl-newvar
```

2. Cambiamos el nombre de */var*:

```
#> mv /var /var_old
```

3. Creamos */var* para que se pueda montar:

```
#> mkdir /var
```

4. Restauramos el contexto de */var*:

```
#> restorecon /var
```

5. Montamos la nueva partición */var*:

```
#> mount -a
```

6. Salimos del modo de seguridad:

```
#> systemctl isolate default
```

7. Eliminamos los archivos antiguos:

```
#> rm -rf /var_old
```

2. Preguntas de examen

2.1. Práctica 1

1. Si monto un RAID0 y un RAID1 con 2 HDD de 10GB, ¿cuánto espacio tengo disponible?

   Solución

   - RAID0: 20GB, ya que no hay redundancia.
   - RAID1: 10GB, ya que los datos se replican.

   ¿Y si uso un HDD de 10GB y otro de 50GB?

   Solución

   - RAID0: 60GB.
   - RAID1: 10GB. Si tomáramos más espacio no podría replicarse.

2. Disponemos de un sistema con particiones *home*, *swap*, *boot* y *root*. Sólo podemos encriptar una de ellas. ¿Cuál elegirías?

   Solución

   SWAP, ya que contiene datos de la RAM, como claves, archivos de las aplicaciones, etc.

3. ¿Qué diferencias hay entre *EXT2* y *EXT4*?

   Solución

   EXT4, a diferencia de *EXT2*, admite *journaling*, es decir, un diario que permite restablecer los datos anteriores a una transacción en caso de que ésta falle. Esta técnica permite al sistema de archivos volver a un estado coherente si se produce un corte de electricidad o cualquier otro fallo durante una transacción. Otra característica muy interesante de *EXT4* es el asignamiento multi-bloque, que permite asignar varios bloques en una misma llamada.

4. ¿Qué diferencias hay entre:

   - *cp -a /var /newvar*
   - *cp -a /var/ /newvar*
   - *cp -a /var/* /newvar*
   - *cp -a /var/. /newvar*

   ?

   Solución

   - *cp -a /var /newvar* copia la carpeta, no el contenido. Es decir, quedaría */newvar/var/archivos*
   - *cp -a /var/* /newvar* no copia los archivos ocultos.
   - *cp -a /var/. /newvar* copia todos los archivos (visibles y ocultos) y además mantiene el contexto.