

Cambios en el documento

- 20 de octubre

Lo que era la práctica 1.7 pasa a ser la 1.8 y viceversa. Si ya habías troceado el fichero ova, para hacer la práctica de usuarios y grupos simplemente crea una nueva.

Formato de la memoria de prácticas

La memoria de estas prácticas debes escribirla en un fichero en formato de texto plano. El fichero estará en tu cuenta del laboratorio. El día del examen, recogeremos automáticamente tus prácticas.

1. Es imprescindible que respetes al pie de la letra los nombres de los ficheros especificados en el guión. Una letra mal puesta equivale a una práctica no presentada (o un examen no presentado). Si bien dispondrás de un script que verificará que has usado los nombres correctos
2. Redacta la memoria describiendo escuetamente todo lo que haces. Esta memoria te será muy útil para preparar el examen práctico (recuerda que podrás llevarla al examen).

No merece la pena que te preocupes de tener una redacción muy cuidada: copia y pega órdenes y resultados, describe telegráficamente lo que haces. Vete preparando la memoria a la vez que trabajas, no lo dejes para el final.

- a) Cuando un paso del guión te pida una orden de shell o similar, es recomendable que primero ejecutes la orden y luego copies y pegues en la memoria. Si resulta algún texto, pégalo también.
- b) Cuando el guión pida que escribas o edites un fichero, basta con que lo modifiques en su sitio. No es necesario que copies y pegues en la memoria.

Práctica 1.1. Directorios de las prácticas

Ahora prepararás el directorio de la práctica 1 y el fichero de texto para la memoria de la esta práctica.

1. Entra por ssh en tu cuenta del laboratorio Linux de la ETSIT. Crea el directorio `~/lagrs`. Aquí guardarás la mayoría de tu trabajo de prácticas en la asignatura
2. Ponle permisos `rwX-----`
Debes mantenerlo así todo el curso. Recuerda que debes ser autor del 100% de tus prácticas y no permitir que ningún compañero tuyo tenga acceso a ningún fragmento.
3. Crea el directorio
`~/lagrs/practica01`
4. Crea el fichero
`~/lagrs/practica01.txt`
Observa que el nombre de este fichero **no** es
`~/lagrs/practica01/practica01.txt` # ¡Este no es el nombre correcto!
Por el momento déjalo vacío.
5. Seguiremos este convenio en todas las prácticas, un directorio para cada práctica y un fichero de texto plano para cada práctica, fichero que cuelga de `~/lagrs`

Práctica 1.2. Uso básico de vi

El objetivo de esta práctica es que sepas usar al menos las órdenes elementales de vi. Si no te gusta este editor, podrás usar algún otro editor sencillo en modo texto. Pero para poder instalar un editor nuevo, necesitas usar vi.

1. Usando vi, crea el fichero `~/lagrs/practica01/ejemplo.txt` y escribe en él 4 titulares de cualquier periódico de hoy. Escribe 3 o 4 faltas de ortografía, intencionadamente. Guárdalo.
2. Vuelva a abrirlo y corrige las faltas.

Práctica 1.3. Uso de un editor sin gráficos

Ahora empezarás a usar el fichero de la memoria de prácticas. Escribe la memoria de este apartado y de todos los apartados siguientes de esta práctica en el fichero que creaste en el apartado 1.1. `~/lagrs/practica01.txt`

Es necesario que manejes con soltura funciones al menos intermedias de algún editor de texto sin gráficos. La recomendación es vi/vim, pero también puedes elegir uno más sencillo.

1. Lee las transparencias sobre <http://gsyc.urjc.es/mortuno/lagrs/editores.pdf>. Arranca al menos una vez los editores en modo texto presentados (vim, mcedit, joe y nano). Elige uno, para usar en esta asignatura. ¿Cuál has elegido? ¿por qué?. Puedes cambiar de opinión mas adelante, pero en tal caso, indícalo (recuerda que estas prácticas las recogeremos el día del examen).
2. Es muy conveniente conocer atajos de teclado. Es mucho más eficaz memorizar, por ejemplo, 2 al día durante 5 días que intentar retener 10 atajos 1 día. Aprende unos cuantos de esta forma, al menos media docena, e indícalo aquí. Ejemplo:

2015.09.24

Elijo vi porque es el más potente. Practico el copy-paste con yy p

2015.09.28

Me coloco en una palabra y pulso asterisco para buscar esa misma palabra en el texto. n minúscula para repetir la búsqueda hacia adelante y N mayúscula para buscar hacia atrás.

2015.09.30

Practico ma, mb, mc para poner las marcas a, b y c en un texto. Vuelvo a las marcas con 'a 'b 'c

Práctica 1.4. Gestión de contraseñas

Guarda dos o tres contraseñas, de prueba o reales, usando

1. gpg
2. LibreOffice
3. KeePassx. O alguna herramienta similar como Bitwarden.

Usa recordatorios de contraseña en todos los casos, en un fichero de texto aparte. Guarda los ficheros donde quieras, describe brevemente en la memoria los pasos que has seguido

Práctica 1.5. Secret Sharing

Usa ssss para descomponer una contraseña en 6 trozos, de forma que baste con 4 para restaurarlos. Reconponla.

Práctica 1.6. Vagrant

En este ejercicio probarás Vagrant, lanzando una máquina virtual de VirtualBox. Siempre en tu ordenador de casa, tendrás que instalar Vagrant y VirtualBox si no lo tenías ya.

1. Crea un *project directory* de Vagrant en el directorio `~/lagers/vbox01` de tu pc de casa.
2. Prepara una máquina Ubuntu 20.04 LTS Focal Fossa.
3. Lanza la máquina y entra por ssh, usando vagrant.
4. Comprueba que el *project directory* del host está montado dentro de la máquina virtual en `/vagrant`
Para ello, edita un fichero en este directorio desde el *guest* o el *host*, guárdalo y vuelve a editarlo desde la otra máquina (el *host* si antes usaste el *guest* y viceversa).
5. Apaga la máquina desde vagrant, sin destruirla.
6. Copia el directorio `~/lagers/vbox01` de tu pc de casa al directorio `~/lagers` de tu cuenta del laboratorio. De esta forma, el directorio `~/lagers/vbox01` de ambas máquinas será idéntico.

Práctica 1.7. Usuarios y grupos

En este ejercicio practicarás con los usuarios y grupos de linux.

1. Vuelve a entrar en la máquina virtual del apartado anterior.
2. Crea un nuevo usuario, con el nombre que quieras.
3. Abre una sesión de este usuario (con la orden `su`)
4. Este usuario no tendrá privilegios para ejecutar `sudo`, pero intenta lanzar alguna orden y observa qué pasa.
5. Haz que este usuario sí pueda ejecutar `sudo`. Pruébalo.
6. Con este usuario, crea dos nuevos grupos con el nombre que quieras. Mete al usuario en estos grupos. Comprueba que están incluidos.
7. Prueba la orden `newgrp`, observa su efecto.

Práctica 1.8. Split

Ahora practicarás con el troceado de ficheros. Supongamos que quieres mover la máquina que acabas de crear a un host distinto. No es un fichero excesivamente grande pero sí tiene cierto tamaño, puede ser conveniente trocearlo. Es lo que practicarás aquí.

1. Desde el interfaz gráfico de VirtualBox, exporta la máquina virtual que acabas de crear. Usa el formato `ova`. Llévalo al directorio auxiliar que prefieras.
2. En ese directorio auxiliar, crea un fichero `README.txt`. En este fichero escribe el hash MD5 del fichero `.ova`.
3. Comprime este directorio con el `.ova` y el `.txt` en un fichero `.tgz`. Ponle el nombre que te parezca adecuado.
4. Divídelo en trozos, del tamaño que te parezca adecuado.
5. Destruye la máquina virtual en VirtualBox (como prefieras, desde Vagrant o desde el GUI)
6. Borra el `.ova`, conservando el fichero de texto con el hash.

7. Borra el tgz. (En otras palabras, borra todo menos los trozos y el hash)
8. Reconstruye el tgz a partir de los trozos.
9. Extrae los ficheros, comprueba que el hash es correcto.
10. Importa el .ova desde el GUI de VirtualBox, comprueba que funciona correctamente.

Práctica 1.9. Instalación de Docker

En este apartado instalarás Docker en tu ordenador

1. Instala docker en tu ordenador de casa.
2. Comprueba que la instalación de docker es correcta, lanzando un contenedor de tipo *holamundo* basada en debian.
3. Haz una prueba de tipo *holamundo* basada en ubuntu.

Práctica 1.10. Uso básico de imágenes

En este apartado empezará a usar los contenedores. Mientras el enunciado no especifique lo contrario, usa imágenes basadas en Ubuntu.

1. Crea un contenedor interactivo. No le pongas nombre. Ejecuta alguna orden básica de la shell. Indica alguna orden básica que esté disponible y alguna otra que no.
2. No detengas el contenedor. Crea otro contenedor interactivo, poniéndole el nombre xxxxc01, donde xxxx son las primeras 4 letras de tu nombre de usuario en el laboratorio. Para jperez, sería jperc01.
3. En un nuevo terminal , usa las ordenes de docker listar contenedores e imágenes. Explica lo que estás viendo.
4. Termina la ejecución de los contenedores y observa el estado de las imágenes y de los contenedores detenidos.
5. Comprueba que el sistema de ficheros dentro del contenedor no es persistente.

Práctica 1.11. Creación de una imagen de un contenedor

1. Crea una cuenta en docker hub.
2. Prepara la imagen **test/banner** descrita en las transparencias, pero llámala **tulogin/banner** siendo *tulogin* tu nombre de usuario en docker hub.
3. Lanza un contenedor con esa imagen.
4. Modifica la imagen para que una vez lanzada, no solo muestre el banner sino que abra una shell
5. Lanza un contenedor con la imagen para comprobar que puedes ejecutar la shell.
6. Sube la imagen a docker hub.

Práctica 1.12. Creación de una imagen personalizada

En este apartado prepararás una imagen sencilla de un contenedor. Se llamará $iTULOGIN_i/cal$, y lo único que hará será invocar a la orden de shell `cal` para mostrar el calendario del mes actual y concluir. Vamos a establecer los siguientes convenios, que mantendremos el resto de las prácticas:

- Usaremos `ubuntu:20.04` como distribución base
- Si la imagen se llama `cal`, y tu login en el laboratorio es `jperez`, los distintos contenedores que ejecutarás a partir de ella se llamarán `jpercal01`, `jpercal02`, etc
Esto es: las primeras 4 letras de tu nombre de usuario, el nombre de la imagen y un número de dos dígitos.
- Todo lo necesario para construir y lanzar esta imagen estará en `~/lagrs/cal`
- El fichero `~/lagrs/cal/construye.sh` será un script para construir la imagen `<TULOGIN>/cal`
- El fichero `~/lagrs/cal/lanza_jpercal01.sh` será un script para lanzar la imagen `jpercal01`.
El fichero `~/lagrs/cal/lanza_jpercal02.sh` será un script para lanzar la imagen `jpercal02`, y así sucesivamente (si quisiéramos lanzar más contenedores)
Observa que los scripts `lanza_jpercal01.sh` `lanza_jpercal02.sh` incluye en su nombre los nombres de los contenedores, porque varios contenedores podrán compartir la misma imagen inicial.
Observa que el script `construye.sh` no incluye el nombre de la imagen, porque dentro del directorio `~/lagrs/cal/` solo habrá ficheros relativos a la imagen `<TULOGIN>/cal`
- El directorio `~/lagrs/cal/context` contendrá el contexto para esta imagen. Por tanto, los ficheros `Dockerfile` y `entrypoint.sh` estarán, respectivamente, `~/lagrs/cal/context/Dockerfile` y `~/lagrs/cal/context/entrypoint.sh`

Atendiendo a estos convenios,

1. Prepara la imagen del contenedor solicitado. La utilidad `cal` está en el paquete `bsdmainutils`.
2. Prepara el script `lanza_jpercal01.sh` y lánzalo.
3. Prepara el script `lanza_jpercal02.sh` y lánzalo.
4. Ejecuta `docker ps -a` y `docker images`, y observa que, naturalmente, se puede comprobar que ambos contenedores están basados en la misma imagen.

Práctica 1.13. FreeFileSync

Ahora practicarás el uso de FreeFileSync para simular que sincronizas tu pc de casa con el laboratorio. Empieza creando en tu pc las carpetas `~/simula_labo/lagrs` y `~/simula_casa/lagrs`. Vamos a imaginar que

1. La carpeta `~/simula_casa/lagrs` es la carpeta `lagrs` de tu ordenador de casa.
2. La carpeta `~/simula_labo/lagrs` es la carpeta `lagrs` de tu cuenta en el laboratorio.

Una vez que lo tengas todo preparado, ya puedes hacer esta práctica:

1. Prepara un fichero `~/simula_labo/lagrs/pueba.txt`. Escribe cualquier texto.
2. Simula que vas a casa y allí añades una línea más. (por ejemplo *ahora estoy en casa*). Simula que vuelves al laboratorio y que vuelves de nuevo a casa. Recuerda que cada vez que estás *en casa*, tienes que sincronizar al principio y al final de la sesión.

Práctica 1.14. Conflictos con FreeFileSync

1. Operando de forma análoga al ejercicio anterior, y con los mismos ficheros, simula un conflicto de sincronización. Observa los mensajes de error de FreeFileSync.
2. Resuelve el conflicto. Comprueba que FreeFileSync ya no muestra errores.

Práctica 1.15. ssh sin contraseñas

Configura tu ordenador para entrar en tu cuenta del laboratorio sin escribir contraseña.

Práctica 1.16. ssh sin contraseñas, desde el laboratorio

(Suprimida)

Práctica 1.17. Reutilización de claves

(Suprimida)

Práctica 1.18. Montaje bind

Ahora prepararás una imagen que cree un contenedor, con un usuario, que haga un montaje de tipo bind. En este ejercicio, usarás un único contenedor. Lo lanzarás en tu ordenador de casa.

Siguiendo el criterio establecido en el apartado anterior, y suponiendo que tu usuario sea `jperez`

- El directorio del contenedor será `~/lagrs/bind/`
- El directorio contexto, `~/lagrs/bind/context`
- El contenido del directorio contexto, `~/lagrs/bind/context/Dockerfile` y `~/lagrs/bind/context/entrypoint.sh`
- El script de creación de la imagen, `~/lagrs/bind/construye.sh`
- El script de lanzamiento del contenedor, `~/lagrs/bind/lanza_jperbind01.sh`

Funcionamiento del contenedor:

- El contenedor tendrá un usuario, con un nombre igual a tu nombre de usuario en el laboratorio.
- El contenedor ejecutará una shell.
- El contenedor montará tu directorio *home* (del servidor de contenedores, esto es, tu pc de casa) en el directorio `/home/jperez` del contenedor

Prueba del contenedor:

1. Lanza el contenedor y escribe en el directorio montado un fichero vacío con nombre `hola_jperez`
2. Comprueba que es persistente. Esto es, sal del contenedor, vuelve a lanzarlo y observa que el fichero `hola_jperez` sigue en su sitio.

Práctica 1.19. sshfs

Ahora practicarás con sshfs. Escribe un script llamado `~/lagrs/practica01/monta_tmp` que monte los directorios `/tmp` de tres máquinas cualquiera de las que estén disponibles en el laboratorio (consulta el *parte de guerra*). Por ejemplo `f-12108-pc05`, `f-12108-pc07`, `f-12108-pc08` y en `~/lagrs/practica01/tmp01`, `~/lagrs/practica01/tmp02`, `~/lagrs/practica01/tmp03` respectivamente.

Comprueba que el resultado es el esperado: entra por ssh en esas máquinas, edita algún fichero en el directorio `/tmp/`, comprueba que puedes editar el mismo fichero en el directorio montado en tu máquina local.

Práctica 1.20. Contenedor con fichero hosts

En este apartado prepararás un contenedor preparado para hacer ping y ssh a las máquinas del laboratorio, usando solo el nombre de host, no el FQDN (Fully Qualified Domain Name).

Esta imagen se llamará `<TULOGIN>/caa` (contenedor aa), estará basada en Ubuntu 20.04 y configurado en español.

Siguiendo el convenio descrito en el apartado anterior:

- Los contenedores lanzados a partir de esta imagen se llamarán `jpercaa01`, `jpercaa02`, etc
- Todos los ficheros necesarios estarán en `~/lagrs/caa`

- Los scripts se llamarán

`~/lagrs/caa/construye.sh`

`~/lagrs/caa/lanza_jpercaa01.sh`

`~/lagrs/caa/lanza_jpercaa02.sh`

(donde `jper` representa las primeras 4 letras de tu login)

- El directorio contexto será

`~/lagrs/caa/context`

La imagen se creará con los ficheros

`~/lagrs/caa/context/Dockerfile`

`~/lagrs/caa/context/entrypoint.sh`

Para conseguir que los contenedores conozcan las direcciones IP de las máquinas del laboratorio, tendrás que añadir al fichero `/etc/hosts` de cada imagen las entradas correspondientes al laboratorio, que encontrarás en el fichero `/etc/hosts` de cualquier puesto.

Para ello

- Prepara, en el directorio contexto, un fichero `delta_hosts` que contenga las entradas necesarias.
- Haz que este fichero aparezca en el directorio `/tmp/` de la imagen.
- Haz que cada vez que se inicie la imagen, se añadan estas entradas al `/etc/hosts` del contenedor.
(En docker no es posible borrar ni reemplazar el fichero `/etc/hosts` de una imagen, pero sí puedes modificarlo añadiendo entradas)
- Haz que desde los contenedores se pueda hacer `ifconfig`, ping y ssh. Para ello necesitarás los paquetes `ubuntu` adecuados, búscalos en internet.

Puedes instalar algún paquete adicional si quieres, con tal de que no resulte una imagen mucho más pesada.

Comprueba que una vez lanzado el contenedor, puedes hacer ping o entrar por ssh a los puestos del laboratorio que lo soporten, sin necesidad de escribir el FQDN, esto es, sin añadir al nombre el dominio `aulas.gsync.urjc.es`.

Ten en cuenta que:

- Solo algunas máquinas del laboratorio acepta conexión por ssh.
- El cortafuegos del laboratorio no permite el tráfico de paquetes ICMP (lo que incluye el ping) desde fuera de su subred. Por tanto, podrás ver que se ejecuta la orden ping y que se envía la petición a la dirección IP adecuada, pero no verás respuesta.

Práctica 1.21. Conectividad entre contenedores

En este apartado instalarás sshd en un contenedor y probarás la red bridge de Docker.

1. Los contenedores estarán basados en Ubuntu 20.04.
2. El nombre de la imagen que crearás será `<TULOGIN>/cab`
Como en la práctica 1.20, al nombre de cada contenedor le añadirás como prefijo las primeras 4 letras de tu usuario, y como sufijo, un número de dos dígitos.
3. Al igual que en la práctica 1.20, este contenedor estará configurado en español, desde él debe ser posible acceder por ssh a los puestos del laboratorio sin usar su FQDN. También lanzar peticiones ping e usar ifconfig.
4. Lanzarás dos contenedores con esta imagen.
Por tanto, los nombres de los ficheros necesarios serán:
`~/lagrs/cab/context/Dockerfile`
`~/lagrs/cab/context/entrypoint.sh`
`~/lagrs/cab/construye.sh`
`~/lagrs/cab/lanza_jpercab01.sh`
`~/lagrs/cab/lanza_jpercab02.sh`
5. Prepara la imagen de forma que el contenedor que la ejecute tenga lanzado el demonio servidor de ssh.
6. Averigua la dirección IP de cada contenedor
7. Haz ping entre ambos
8. Añade un usuario a `jpercab01`
Hazlo de forma interactiva, esto es, desde la shell. Elige el nombre y contraseña que quieras.
9. Abre una sesión desde `jpercab02` hasta `jpercab01`

Práctica 1.22. Benchmark de cpu

Observa los BogoMIPS de tu ordenador. Compáralos con los de los puestos del laboratorio.

Práctica 1.23. Benchmark de red

Escribe un pequeño informe basado en iperf sobre la calidad de alguna pequeña red de datos. Puede ser la de tu casa o la de alguna otra red a la que tengas acceso. Puedes comparar la conexión entre diferentes puntos o a diferentes horas del día. Puedes observar el efecto de elementos como el microwaves o la lavadora.

Revisión de los nombres de los ficheros

Ejecuta `~/mortuno/revisa practicas lagrs` para comprobar que los nombres de los programas son los correctos.