Laboratorio de Administración y Gestión de Redes y Sistemas 2022-2023
Grado en Ingeniería Telemática
Escuela Técnica Superior de Ingeniería de Telecomunicación
Universidad Rey Juan Carlos

Observaciones

- Crea el fichero `~/lagrs/practica03.txt`, que contendrá la memoria que escribas sobre este bloque de prácticas. En la primera línea, indica tu nombre y login.

Práctica 3.1. FHS

Busca en los exámenes de teoría resueltos de cursos anteriores todas las preguntas sobre FHS, *Filesystem Hierarchy Standard*, indica en qué convocatorias han aparecido y resume su contenido.

Práctica 3.2. Recode

Crea el directorio `~/lagrs/practica03/recode` y escribe dentro los ficheros que solicite este ejercicio

1. Genera 2 o 3 ficheros de texto plano, copiando y pegando desde cualquier página web. Ponles extensión .txt
2. Comprueba la codificación empleada en estos fichero
3. Comprueba la codificación empleada en tu máquina
4. Haz una copia de cada fichero en una codificación distinta. (latin1, alguna codificación unicode distinta a la codificación por omisión...)
   Ponle nombres añadiendo un sufijo que indique la codificación, entre el nombre de fichero y la extensión `.txt`. Ejemplo:

   ```
   quijote.latin1.txt   hans_reiser.utf16.txt
   ```

Práctica 3.3. netstat

Abre una sesión ssh entre tu ordenador y un puesto del laboratorio. Lanza netstat en tu ordenador y analiza el resultado. Lánzalo en el puesto y analiza el resultado

Práctica 3.4. tmux

El objetivo de esta práctica es que uses *tmux* para mantener demonios corriendo sin terminales abiertos, así como multiplexar sesiones.
Prepararás una sesión de tmux, con dos ventanas:

- Una con dos paneles, uno para el script *tictac* y otro para la orden *vmstat*.
- Otra con una única panel, para la orden top.

Hazlo según los siguientes pasos:

1. Empieza familiarizándote con tmux, probando los ejemplos de las transparencias. No hace falta que documentes esto en la memoria.
2. Crea el fichero `~/lagrs/practica03/tictacTULOGIN` y pega en él este script. (TULOGIN serán las primeras 4 letras tu nombre de usuario en el laboratorio).

```
#!/bin/bash
fichero_salida=/tmp/log.$USER.txt
while true
do
    sleep 1
    echo -n "tic"  >> $fichero_salida
    sleep 1
    echo " tac"  >> $fichero_salida
done
```

Pruébalo usando `tail -f` (que escribe en la salida estándar las últimas líneas de un fichero, y queda a la espera de que haya cambios en el fichero, para escribirlos también).

3. Busca una máquina linux del laboratorio a la que puedas entrar por ssh.
4. Copia `tictac` al directorio `/tmp/` de la máquina remota.
5. Entra por ssh en la máquina remota.
6. Haz lo necesario para invocar a *tictacTULOGIN* y que quede funcionando en una sesión de tmux.
7. Deja *tictacTULOGIN* corriendo y sal de la máquina remota.
8. Vuelve a entrar por ssh y recupera el control de la sesión de *tmux* con *tictacTULOGIN*.
9. Crea dentro de la ventana donde está *tmux* otro panel. Ejecuta en él `vmstat 2`.
10. Muévete entre un panel y otro.
11. Deja todo corriendo. Crea otra ventana y ejecuta en ella la orden `top`.
12. Sal de la máquina remota, dejando todo en marcha (tictacTULOGIN, vmstat y top).
13. Vuelva a entrar y comprueba que sabes recuperar todos los programas (los dos paneles de la primera ventana y el panel único de la segunda ventana)
14. Mata con `Crl C` los procesos de las tres ventanas. Cierra todas las ventanas. Comprueba (con ps) que no queda ninguna sesión de tmux. Cierra la sesión de ssh.

Práctica 3.5. Túnel ssh inverso

En este ejercicio probarás el uso de un túnel inverso de ssh, que te permitirá acceder desde cualquier lugar de internet a un servicio en una dirección privada, detrás de un NAT, sin necesidad de *port forwarding* (*abrir puertos*). Usarás tu host del laboratorio como proxy, que como sabes tiene una dirección IP pública (aunque para este ejercicio bastaría cualquier IP accesible para el cliente, podría ser por ejemplo una IP privada del laboratorio). También puedes hacer este ejercicio en casa, usando tu ordenador como proxy.

Usaremos un servicio de prueba, similar a un *holamundo*. Son un par de scripts en python, cliente y servidor. El servidor recibe un número arábigo y devuelve el número romano correspondiente. Colocarás este servicio detrás del NAT creado por VirtualBox, en una red privada. El proxy, *delante* del NAT permitirá acceder este servicio.

- Observa que si haces este ejercicio en casa, tu host no tiene una dirección IP pública, sino una IP privada creada por el NAT de tu router. Al colocar el servicio detrás del NAT de VirtualBox, estará detrás de dos NATs. Con este ejercicio, permitirás que el servicio esté disponible en la red de tu casa, no en la red creada por VirtualBox. En otras palabras, pasará de estar *tras dos NATs* a estar solo *tras un NAT*.

1. Descarga romanserver.py y romanclient.py en `~/bin`, dale permisos de ejecución y comprueba que el directorio `~/bin` está incluido en tu variable de entorno PATH.

2. Prueba romanserver.py y romanclient.py, en cualquier puerto TCP de tu host del laboratorio o de tu ordenador. Encontrarás sus instrucciones de uso al lanzarlos sin argumentos.

3. Usando vagrant, lanza la máquina virtual de la práctica 1 (vbox01).

4. Instala tmux en la máquina y abre varias ventanas (varios terminales)

5. Comprueba que puedes entrar por ssh desde vbox01 hasta tu ordenador.

   - Si estás en tu casa, necesitaras la dirección privada de tu máquina en la red privada que tu router ha creado en tu casa. Puedes usar *ifconfig* para averiguarla.

6. Para poder usar romanserver.py en vbox01, instala python.

7. Copia romanserver.py en cualquier directorio de vbox01 y lánzalo en el puerto 8000.

8. Usando un túnel inverso de ssh, haz que este servicio esté disponible en el puerto 9000 de tu ordenador.

9. Prueba el túnel usando romanclient.py en tu host.

Práctica 3.6 Cron

En esta práctica probarás el uso básico de cron. Los puestos físicos del laboratorio no tienen cron instalado, ya que son máquinas que no están siempre encendidas. Haz el ejercicio en uno de los puestos virtuales del laboratorio o en tu ordenador de casa.

Describe brevemente las tablas de cron que usas, en el fichero `~/lagrs/practica03.txt`

1. Comprueba que cron funciona: Programa en el host una tarea que ejecute cada minuto *touch* sobre el fichero `/tmp/test_cron_tulogin` (donde *tulogin* es tu login en el laboratorio). Consulta la fecha de este fichero, si todo está en orden, borra esta entrada de la tabla de cron

2. Escribe el script `~/lagrs/practica03/escribe_log` en bash o python que añada la cadena *probando cron*, junto con la fecha y la hora, al fichero `~/lagrs/log.txt` Como en cualquier log, debes añadir al final, sin borrar los mensajes precedentes.

   No le pongas extensión al script, ni `.sh` ni `.py`.

   Puedes usar el mandato de shell `date`

3. Programa una tarea que ejecute el script anterior cada minuto. Cuando veas que funciona, modifica la entrada para que se ejecute a las 9 de la mañana, de lunes a viernes.

Revisión de los nombres de los ficheros

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