

Ejercicios de Shell Script III

1) Tenemos un laboratorio de PCs donde cada ordenador tiene un fichero `/etc/hosts` que indica los nombres y direcciones IP de las demás máquinas. El fichero será similar a este:

```
212.128.4.203 a202e33.escet.urjc.es a202e33
212.128.4.204 a202e34.escet.urjc.es a202e34
212.128.4.209 a202e39.escet.urjc.es a202e39
212.128.4.210 a202e40.escet.urjc.es a202e40
212.128.4.211 gamma01.escet.urjc.es gamma01
212.128.4.212 gamma02.escet.urjc.es gamma02
212.128.4.213 gamma03.escet.urjc.es gamma03
```

Vamos a cambiar de dirección las máquinas gammaNN (donde NN es el número del ordenador). La nueva dirección de cada máquina será

192.168.0.YY

donde YY= NN+40

Haz un script de shell que muestre por salida estándar el nuevo fragmento de `/etc/hosts`. Para el fichero visto anteriormente, la salida sería:

```
192.168.0.41 gamma01.escet.urjc.es gamma01
192.168.0.42 gamma02.escet.urjc.es gamma02
192.168.0.43 gamma03.escet.urjc.es gamma03
```

Nota: No sabemos cuantos PCs hay. No podemos suponer que el fichero esté ordenado. Puede haber huecos en la numeración (p.e. que existan el gamma04 y gamma06 pero falte el gamma05)

2) Tenemos un directorio que contiene, entre otras cosas, fotos: ficheros con extensión `.jpg` o `.JPG`. Las fotos tienen mucha resolución. Queremos reducirlas a 800x600 puntos y publicar la versión reducida en una web. Para reducir el tamaño podemos usar el comando

```
convert -geometry 800x600 origen destino
```

Para publicar en el web, basta copiar al directorio `public_html` del home del usuario. Suponemos este directorio existente y con los permisos adecuados.

Proceden de un sistema contaminado por un virus, así que hay ficheros que a pesar de su extensión, no son imágenes jpeg sino ejecutables. Si son verdaderas imágenes el comando "file" mostrará un mensaje similar a este:

```
imagen01.jpg: JPEG image data, EXIF standard 0.73, 10752 x 2048
```

Haz un script de shell bash que reciba como primer argumento el directorio, que compruebe cada fichero, que lo reduzca y publique si está bien y que lo borre si está contaminado, mostrando un mensaje parecido a este:

```
imagen01.jpg CONTAMINADO. Se borra el fichero
imagen02.jpg ok. Reducida y publicada
imagen03.jpg ok. Reducida y publicada
```

1 ficheros contaminados y borrados

2 ficheros reducidos y publicados

3) En cierta asignatura, los estudiantes deben realizar una página web. Todos los matriculados tienen cuenta en la máquina, aunque algunos nunca han llegado a abrir una sesión. A estos los llamaremos usuarios inactivos. Se desea hacer un script bash que haga una copia de la web de todos los usuarios activos, que borre la cuenta de los usuarios inactivos y que muestre un pequeño informe de lo que está haciendo. Los requisitos se detallan a continuación:

1. Los usuarios matriculados tienen su home en el directorio **al-03-04**. Este es un fragmento del fichero */etc/passwd*:

```
jperez:x:10912:1009:Juan Perez,,,:/home/al-03-04/jperez:/bin/bash
mfernán:x:10913:1009:Manuel Fernandez,,,:/home/al-02-03/mfernán:/bin/bash
mgarcia:x:10914:1009:Maria Garcia ,,:/home/al-03-04/mgarcia:/bin/bash
lruiz:x:10915:1009:Luis Ruiz ,,:/home/al-03-04/lruiz:/bin/bash
```

2. Si un usuario es inactivo, el comando *finger* indicará "Never logged in". Ejemplo:

```
$ finger jperez
```

```
Login: jperez                      Name: Juan Perez
Directory: /home/al-03-04/jperez    Shell: /bin/bash
Never logged in.
No mail.
No Plan.
```

3. Las copias de las páginas web queremos guardarlas en

```
/var/tmp/mgarcia
/var/tmp/lruiz
```

Estos directorios no existen previamente.

4. Para hacer una copia de un web, usamos el comando *wget* del siguiente modo

```
wget -m http://localhost/~mgarcia
```

Esto copia el web de mgarcia en el directorio actual

5. El informe que muestra el programa debe ser parecido a este:

```
jperez. Usuario inactivo, se borra su cuenta
mgarcia. Usuario activo, copiando su web en /var/tmp/mgarcia
lruiz. Usuario activo, copiando su web en /var/tmp/lruiz
```

4) Tenemos en un sistema tipo UNIX a una serie de usuarios ya creados. Hemos visto que algunos de ellos tienen su cuenta invadida por personas sin derecho a usar el equipo. Una cuenta está invadida si en el home del usuario hay un directorio llamado ".rootkit".

Este es un ejemplo del fichero de configuración de usuarios (*/etc/passwd*):

```
jperez:x:10912:1009:Juan Perez,,,:/home/jperez:/bin/bash
mfernán:x:10913:1009:Manuel Fernandez,,,:/home/mfernán:/bin/bash
mgarcia:x:10914:1009:Maria Garcia ,,:/home/mgarcia:/bin/bash
lruiz:x:10915:1009:Luis Ruiz ,,:/home/lruiz:/bin/bash
```

Queremos un script que reciba como primer parámetro el fichero de configuración de usuarios y que para cada cuenta:

1. Si está invadida, imprima un mensaje de alerta, la cancele (usando el comando correspondiente) y mueva el home del usuario al directorio */invadidos/*

Puedes usar el comando "test -e fichero" que devuelve cierto si existe el fichero.

2. Si no está invadida, nos diga desde que máquina se conectó el usuario la última vez. Para esto puede usarse el comando *finger*, que para usuario que han usado alguna vez la cuenta muestra algo como esto:

```
$ finger mgarcia
```

```
Login: mgarcia
```

```
Name:
```

```
Directory: /home/mgarcia
```

```
Shell: /bin/bash
```

```
Last login Sun Jun 13 15:59 (CEST) on pts/14 from sumaquina.com
```

```
No mail.
```

```
No Plan.
```

(Observa que finger coloca un espacio entre la mayoría de las palabras de su salida)

5) Tenemos un directorio que contiene, entre otras cosas, scripts de shell. Se desea modificarlos, insertando entre su primera línea y segunda el copyright del autor y la fecha. Por ejemplo, el programa

```
#!/bin/bash
echo hola mundo
```

Se quedaría así:

```
#!/bin/bash
# (c) Juan Garcia. You can redistribute this program under the terms of the GNU GPL.
# mar abr 12 16:05:20 CEST 2005
echo hola mundo
```

Escribir un script de shell bash que haga esta tarea. Se usará por ejemplo de la siguiente forma:

pon_licencia \$HOME/fuentes \$HOME/licencia.txt

- El primer parámetro indica el directorio donde están los fuentes. Todos llevan extensión .sh (en minúsculas).
- La primera línea de cada script debe respetarse, podría no ser exactamente *#!/bin/bash*. (Sugerencia: siendo *n* el número de líneas del fichero, concatena la primera línea con el texto nuevo con las *n-1* últimas).
- El mensaje de *copyright* es una única línea de texto, que está en el fichero que se pasa como segundo parámetro. La fecha es la que proporciona el comando *date*. Ni el fichero con la licencia incluye la almohadilla '#' ni la salida del comando lo muestra.
- El script debe comprobar que recibe exactamente 2 parámetros, que el primero es un directorio y el segundo un fichero legible. Si alguno de estos requisitos no se cumple, se muestra por *stderr* un mensaje de error especificando el problema. (Recuerda que para mostrar un mensaje por *stderr* basta con redirigir texto a **&2**)