



Introducción a los Sistemas Operativos

Examen de prácticas

30 de octubre de 2015

Grupo y profesor:	DNI:
Nombre:	Apellidos:

Instrucciones (MUY IMPORTANTE):

- Tiene 1.5 horas para hacer el examen. Si decide no hacerlo o termina antes de tiempo, no abandonará el laboratorio hasta que se le indique.
- No puede sacar el examen del laboratorio. Tendrá que entregarlo junto con su nombre, DNI y grupo al profesor al finalizar el mismo.
- Los ordenadores no tienen acceso a Internet. Sólo podrá consultar las páginas de manual y cualquier otra fuente de información que haya disponible en su ordenador. Los PDFs con los boletines se han dejado en el directorio `/home/alumno/ISO_boletines_practicas_laboratorios` para su consulta.
- Tiene que entregar un fichero de texto por cada ejercicio con la solución del mismo.
- Una vez finalizado el examen y creados los ficheros con las soluciones, llame al profesor para que éste los verifique. Él conectará el cable de red, para que usted mismo envíe los ficheros por el Aula Virtual, a través de una tarea creada a tal efecto. Adjunte los ficheros de forma individual, sin crear un archivo comprimido con todos ellos.
- En el caso de que se pida un guión shell, el fichero con la solución debe contener el guión shell en sí. Si desea hacer alguna observación, añádala como comentario dentro del propio guión shell.

Ejercicios:

1. (2 puntos) Se pide encontrar una orden de una sola línea (sólo con tuberías, sin bucles ni condicionales) que nos permita obtener una estadística del número de veces que aparece cada extensión (`.txt`, `.pdf`, etc.) al final del nombre en los ficheros regulares del directorio actual y subdirectorios. Si el fichero no tiene extensión, no deberá tenerse en cuenta. Se muestra a continuación un ejemplo de la salida de la orden. **AYUDA:** La orden `rev` invierte cada línea del fichero de texto que recibe por su entrada estándar (para más información `man rev`).

```
$ linea de codigo con tuberías
3      c
1      pdf
2      tex
```

2. (4 puntos) Implemente un guión shell llamado `cuenpalnom.sh` que reciba como parámetros un fichero de texto y un directorio. Se deberá comprobar que se reciben dos parámetros, que el primer parámetro es un fichero regular y que el segundo es efectivamente un directorio. En caso de que no suceda así se sacará un mensaje por la salida estándar de error indicando su uso correcto y se saldrá del programa devolviendo el valor 1.

El guión deberá sacar una estadística indicando para cada palabra **diferente** del fichero de texto, **en orden alfabético**, el número de ficheros del directorio indicado, incluyendo subdirectorios, que contienen dicha palabra **en su nombre**, sin tener en cuenta la ruta de directorios, y sin importar el número de veces que la contiene. Aquellas palabras que no aparecen, no deben salir en el listado. Podemos suponer que si aparecen números se considerarán también palabras, sin embargo deben eliminarse los signos de puntuación. Suponga que se usan sólo los caracteres del idioma inglés (esto es para evitar problemas con eñes y acentos).

Por ejemplo, si el fichero de texto se llamara `lista.txt` y contuviera:

```
hola, esto      no es
(nada mas que) una
prueba de que no supone nada.
```

Y el directorio `dir` tuviera los siguientes ficheros en él o en sus subdirectorios:

```
estocolmo.jpg
ensenada.jpg
foto.jpg
texto.txt
pelicano.jpg
estoque.jpg
nadadores.doc
nononono.es.txt
```

La salida sería:

```
$ ./cuenpalnom.sh lista.txt dir
es          4
esto        2
nada        2
no          2
que         1
```

Si lo llamamos con errores:

```
$ ./cuentapalnom.sh lista.txt
USO ./cuentapalnom.sh fichero_de_palabras directorio
```

3. (4 puntos) Parece ser que un hacker ha entrado al sistema y nos piden que obtengamos información sobre los ficheros que pudieran haber sido afectados. Para ello nos solicitan que hagamos un guión shell que reciba como parámetros un número de días, dos tiempos en el formato HH:MM y una lista de directorios. Suponga que las horas se proporcionan en formato correcto y siempre con dos dígitos tanto la hora como los minutos, por tanto **no hay que comprobar la corrección de los parámetros**.

Deberá devolver una lista **ordenada** de los usuarios que estén ejecutando algún proceso en el sistema. A continuación de cada nombre de usuario, aparecerá la lista de ficheros regulares que, estando dentro de los directorios proporcionados, pertenezcan a dicho usuario, con la condición de que se hayan modificado dentro del número de días indicado, y cuyo tiempo de modificación sea mayor estrictamente que el primer tiempo y menor estrictamente que el segundo. Deberá mostrar una salida como la del ejemplo, donde tras cada usuario aparezca una lista con el día, mes y hora de modificación y el nombre de cada fichero, **ordenada por la fecha y hora**.

Ayuda: Investiga el argumento "%T" de `-printf` en la orden `find`. Dos horas se pueden comparar lexicográficamente (es decir, como si fueran cadenas) si todos los campos (horas, minutos y segundos) tienen el mismo número dígitos, por ejemplo 08:05:05. En el caso de las fechas pasa lo mismo, siempre que se escriban en el formato año/mes/día, por ejemplo, 2015/08/30.

Ejemplo de ejecución: Si suponemos que los usuarios que están ejecutando algo actualmente son `root`, `jose` y `antonio`, para cada uno de ellos se mostrará la lista de los ficheros que les pertenecen en `dir1`, `dir2` y `dir3` que hayan sido modificados hace menos (estrictamente) de 4 días y entre las 18:00 y las 19:30.

En el momento de la ejecución estamos a día 23 de octubre. Si algún usuario no tiene ningún fichero, aparecerá sólo su nombre.

```
# ./entre2tiempos 4 18:00 19:30 dir1 dir2 dir3
Usuario antonio:
21/10 18:09 dir1/fichero1
21/10 19:16 dir3/fichero3
23/10 18:46 dir3/fichero2
Usuario root:
Usuario jose:
21/10 18:41 dir2/fich3
21/10 18:42 dir1/fich2
23/10 18:43 dir3/fich1
```

Soluciones

1. La línea sería la siguiente:

```
find . -type f -printf "%f\n" | rev | cut -s -f1 -d"." | rev | sort | uniq -c
```

o bien

```
find . -type f -printf "%f\n" | grep '\.' | rev | cut -f1 -d"." | rev | sort | uniq -c
```

o bien

```
find . -type f -name '*..*' | rev | cut -f1 -d"." | rev | sort | uniq -c
```

2. El guión `cuenpalnom.sh` sería el siguiente:

```
#!/bin/bash

if [ $# -ne 2 -o ! -f "$1" -o ! -d "$2" ]
then
    echo "USO: $0 fichero_de_palabras directorio" 1>&2
    exit 1
fi

cat "$1" | tr "[:upper:]" "[:lower:]" | tr -c "[a-z0-9] " " " | tr -s " " "\n" | sort |
uniq | while read linea
do
    numero=$(find "$2" -type f -name "$linea*" | wc -l)
    if [ $numero -ne 0 ]
    then
        echo -e "$linea\t$numero"
    fi
done
```

Script 1: `cuenpalnom.sh`

3. El guión `entre2tiempos.sh` sería el siguiente:

```
#!/bin/bash

dias="$1"
shift
tiempoi="$1"
shift
tiempof="$1"
shift
directorios="$*"

usuarios=$(ps aux | tail -n +2 | cut -f1 -d" " | sort | uniq)
for u in $usuarios
do
    echo "Usuario $u:"
    find $directorios -type f -user $u -mtime -$dias -printf "%Tm %Td %TH:%TM %p\n" |
    sort | while read linea
    do
        hora=$(echo $linea | cut -f3 -d" ")
        mes=$(echo $linea | cut -f1 -d" ")
        dia=$(echo $linea | cut -f2 -d" ")
        nombre=$(echo $linea | cut -f4- -d" ")
        if [ "$hora" \> "$tiempoi" -a "$hora" \< "$tiempof" ]
        then
            echo "$dia/$mes $hora $nombre"
        fi
    done
done
```

Script 2: `entre2tiempos.sh`