

Sistemas operativos I

Practica 3

Xavi Cano
Grup C

Problema 1:

1. Encontrar todos los ficheros dentro del directorio coreutils-8.25 y subdirectorios. El resto de puntos de este ejercicio debe resolverse únicamente con esta búsqueda inicial.

```
find . -name "*.*"
```

2. Utilizando el resultado del paso 1, borrar todos los ficheros con extensión “.o” (i.e. evitar volver a buscar los ficheros “.o”). Se valorará evitar la utilización de bucles.

```
find . -name "*.o" -exec rm {} \;
```

3. Utilizando el resultado del paso 11, cambiar de fecha de los ficheros “.c” al 23 de abril del 2016. Evitar volver a buscar los ficheros “.c”. Se valorará evitar la utilización de bucles.

```
find . -name "*.c" -exec touch --date="2016-04-23 11:00" {} \;
```

4. Utilizando el resultado del paso 1, cambiar la fecha de los ficheros “.h” al 23 de abril del 2016 únicamente si el nombre de fichero comienza con una “s”. Evitar volver a buscar los ficheros “.h”.

```
find . -name "s*.h" -exec touch --date="2016-04-23 11:00" {} \;
```

Problema 2:

1. Buscar todos los ficheros “*.txt” que están incluidos dentro del directorio.

```
find . -name "*.txt"
```

2. Filtrar la búsqueda anterior y quedarse únicamente con los ficheros “*.txt” que se encuentren debajo del subdirectorio “etextXX”, donde “X” es un número del 0 al 9. Los ficheros pueden estar a cualquier profundidad.

```
find ./etext?*/ -name "*.txt"
```

3. Contar, para cada uno de los ficheros encontrados en el paso anterior, cuántas líneas y palabras tiene éste. Esta información se volcará en un único fichero de texto con nombre `table.txt` con las cuatro columnas que `wc` imprime por defecto a pantalla: el número de líneas, el número de palabras, el número de caracteres y el nombre de fichero. El fichero de texto `table.txt` tendrá por lo tanto tantas líneas como ficheros analizados.

```
find ./etext?*/ -name "*.txt" -exec wc {} \; >> table.txt
```

4. Ordenar el fichero obtenido en el paso 3 por el número de palabras (la segunda columna), de mayor a menor, y guardar los resultados en un fichero temporal `tmp.txt`.

```
find ./etext?*/ -name "*.txt" -exec wc -w {} \; | sort -nk1 >> tmp.txt
```

5. Imprimir el mensaje “El numero de ficheros analizados es XXX” en un nuevo fichero denominado `score.txt`, donde XXX es el número de ficheros analizados.

```
find ./etext?*/ -name "*.txt" -exec wc {} \; >> score.txt
num=`wc -l score.txt`
echo "El numero de ficheros analizados es $num" >> score.txt
```

6. Imprimir el mensaje “Los 10 ficheros con mayor numero palabras” seguido de los 10 primeros resultados obtenidos en el punto 4. A continuación imprimir el mensaje “Los 10 ficheros con menos palabras” seguido de los 10 resultados con menos palabras. No es necesario que las columnas estén “perfectamente” alineadas. Mirad el manual de `head` y `tail` para responder el problema.

```
find ./etext?*/ -name "*.txt" -exec wc -w {} \; | sort -nk1 | echo -e "Los 10 ficheros con mayor numero palabras:\n `tail -n 10`"
find ./etext?*/ -name "*.txt" -exec wc -w {} \; | sort -nk1 | echo -e "Los 10 ficheros con menor numero palabras:\n `head -n 10`"
```

Problema 3:

1. Buscar líneas que contengan la cadena “this” y “to” en una misma línea. Tanto “this” como “to” pueden formar parte de una palabra más grande.

```
egrep 'this|to' book.txt
```

2. Buscar líneas que contengan alguna palabra que acabe en “ly”.

```
egrep '*ly' book.txt
```

3. Buscar palabras de 4 letras. Estas palabras pueden contener cualquier letra de la a a la z.

```
grep -Ew '\w{4}' book.txt
```

4. Buscar palabras que comiencen por una vocal (a, e, i, o, u) y acaben con “ing”.

```
grep ' [aeiou]\w*ing ' book.txt
```

5. Realizar una búsqueda y sustitución en book.txt para que añada un “-yeah!” a las palabras encontradas en el punto 4.

```
sed 's/ [aeiou]\w*ing /-yeah!/' book.txt
```