

Universidad de Valladolid – Departamento de Informática

# Fundamentos de Informática II

## Laboratorio - Programación del Shell de UNIX

### Guía de Prácticas

Sesiones 4 y 5

Curso 2009/2010

En todos los casos a continuación si el script shell se realiza sin errores debe finalizar con un código de salida igual a cero. En otro caso debe finalizar con un código de salida distinto de cero. El último comando en ejecutarse debe ser siempre la última línea del script (solo debe existir una sentencia `exit`, localizada en la última línea del script).

Se DEBEN realizar todas las verificaciones necesarias para asegurar que los scripts no se interrumpan por la presencia de errores de entrada de datos.

## 1. Ficheros inscritos

En los siguientes script se debe hacer uso de ficheros inscritos en las situaciones en las que se considere conveniente.

1. Realizar un programa que permita ordenar la siguiente lista de datos:

- England
- Spain
- Netherlands
- Italy
- Austria
- France
- Germany
- Luxembourg
- Greece
- Belgium
- Bulgaria

- Cyprus
- Czech Republic
- Denmark
- Estonia
- Finland
- Slovakia
- Hungary
- Ireland
- Latvia
- Malta
- Netherlands
- Poland
- Italy
- Portugal
- Romania
- Slovenia
- Sweden
- Lithuania
- United Kingdom

El programa debe mostrar:

- La lista ordenada, la cual deberá estar almacenada en una variable.
  - La lista ordenada en forma inversa.
  - El país con el nombre más largo.
  - El país con el nombre más corto.
  - El número total de países
2. Realizar un programa que permita mostrar el equivalente en notación decimal de un número romano que debe ser introducido por el usuario. Se supone que el número introducido no tiene errores. El número romano se dará siempre en mayúsculas. El valor decimal del número romano podrá estar en cualquiera de los siguientes intervalos:
- 1 - 30
  - 50 - 55
  - 100 - 105
  - 500 - 505
  - 1000 - 1005
3. Realizar un script que sugiera cinco claves, que podrán ser cualquier combinación de palabras y cifras. El conjunto de palabras y cifras deben darse como ficheros inscritos. La sugerencia debe ser de la forma:

PalabraCifraPalabraCifra....

En pares determinados por un dato que debe aceptarse como argumento del script. Así para  $n = 1$  y palabras [uno dos tres] y cifras [34 56 01] las siguientes son posibles salidas validas del script.

La palabra clave es:  
 uno34  
 dos34  
 tres01  
 dos56  
 uno56

o

La palabra clave es:  
 uno56  
 tres34  
 tres56  
 dos01  
 uno56

Para  $n=2$ , una salida válida sería:

La palabra clave es:  
 uno56tres01  
 tres34dos34  
 tres56dos01  
 dos01uno56  
 uno56uno34

## 2. Interrupciones

1. Realizar un programa que lea un nombre de usuario y que no permita su interrupción hasta que el nombre introducido sea *mxyzptlk*.
2. Realizar un programa que lea un nombre de usuario y una palabra clave.
  - Los pares válidos de nombres de usuarios y palabras clave deberán figurar en el programa como un fichero inscrito.
  - El programa no debe permitir su interrupción mientras no se introduzca un nombre de usuario y una clave válidas.
  - Sin embargo, si el usuario presiona *Ctrl-D* el programa debe abrir un shell y a la salida de este volver al programa de petición de claves.

3. Realizar un programa que permita copiar los ficheros de un directorio *origen* (que debe existir y no estar vacío) en un directorio *destino* (el cual no debe existir).
  - El programa debe informar en todo momento cual es el fichero que está siendo procesado.
  - En caso de ser interrumpido antes de terminar por el usuario, mediante *Ctrl-C*, el programa debe borrar el directorio *destino* y cualquier fichero que haya sido copia en éste directorio.
  - Al finalizar el programa debe informar del número de ficheros copiados.
  - El programa debe realizar todas las validaciones necesarias sobre los datos introducidos, en caso de detectar algún error debe informar de éste al usuario indicando además la forma de uso correcto del script.
  - En caso de ser interrumpido el programa debe informar de esta incidencia con un mensaje adecuado y finalizar con un código de salida igual a 5.
  - Se recuerda que el script debe tener un único punto de salida.

### 3. El comando exec

1. Comprobar el funcionamiento del comando **exec** para ejecución de scripts desde un terminal cualquiera con los ejemplos proporcionados en los apuntes.
2. Utiliza el comando **exec** para redirigir la E/S estándar dentro de un script. El script debe tener dos argumentos, que serán los nombres de los ficheros entrada y salida hacia los que se deben redireccionar las E/S estándar. En el fichero entrada deben estar los comandos u órdenes a realizar, uno por línea. El fichero salida tendrá los mensajes y la salida que genere el script.
3. Desarrolla un script llamado **exelog.sh** que permita generar un fichero de seguimiento que contenga todas las salidas generadas por un script. Por ejemplo, la siguiente llamada permitirá obtener un fichero de seguimiento llamado **datos.txt** del script **ejemplo.sh** con argumentos **uno**, **dos** y **tres**.

```
exelog.sh datos.txt ejemplo.sh uno dos tres
```

4. Realizar un script que permita visualizar el contenido de dos ficheros de texto, intercalando cada línea del primer fichero con la línea correspondiente del segundo fichero.  
Los nombres de los ficheros se pasarán al script como argumentos. El script debe realizar todas las comprobaciones necesarias. En caso de error deberá mostrar un mensaje de explicación del fallo.

## 4. Funciones

1. Crear una función que acepte como argumento una cadena que puede contener espacios y que devuelva la cadena con los espacios reemplazados por guiones bajos (*underscores*).
2. Crear una función que acepte como argumento una cadena que puede contener guiones bajos y que devuelva una cadena en la que se han reemplazado los guiones bajos por espacios.
3. Crear una función que acepte una cadena y un número  $p$  como parámetros y que devuelva el carácter ubicado en la posición  $p$  de la cadena. Se supone que los caracteres de la cadena se numeran a partir de cero.
4. Crear un script que permita buscar los ficheros que tienen cierto tipo de derechos de protección para el usuario, el grupo o los *otros* en un directorio dado. En el siguiente ejemplo se buscan los ficheros que tienen permiso de ejecución para los *otros*, en el directorio `/usr/luis14/`.

```
$ buscader o x /usr/luis14/
/usr/luis14/
-rwxrwxrwx uno
-rwx---rwx datos.txt
/usr/luis14/temp
-rwx-----x 51.tmp
-rwx-----x aux.tmp
/usr/luis14/ejem
-rwxrw---x ejemplo.sh
$
```

Como se puede apreciar, el script debe buscar los ficheros en el directorio dado y todos sus subdirectorios.

5. Modificar el script anterior para que se admitan búsquedas del siguiente tipo:
  - `buscacod go wx /usr/luis14/` Busca por permisos de escritura y ejecución para el grupo y los *otros*.
  - `buscacod uo rwx /usr/luis14/` Busca por permisos de lectura, escritura y ejecución para el usuario y los *otros*.
  - `buscacod ug rx /usr/luis14/` Busca por permisos de lectura y ejecución para el usuario y el grupo.
6. Crear una nueva versión del script anterior, que en este caso muestre los ficheros que no cumplen con la característica especificada.
  - `buscacodinv go wx /usr/luis14/` Muestra los ficheros que no tienen permisos de escritura ni ejecución para el grupo y los *otros*.
7. Crear un sistema basado en scripts del Shell de UNIX que permita mantener un inventario de CDs con los siguientes datos:

- Nombre del Album
- Artista
- Genero
- Año

El sistema debe permitir la inicialización, adición, modificación y la eliminación de datos de CDs de la “*base de datos*” del sistema. Además debe admitir varios tipos de consultas y búsquedas. La interfaz del sistema debe estar gestionada por menús de opciones.

Se pueden usar todas las funciones y scripts que se consideren necesarios.

Se sugiere que el “*sistema*” se encuentre en un subdirectorio cualquiera.

Explicar cual es la estrategia de almacenamiento de datos elegida, el formato de almacenamiento de datos seleccionado y las posibilidades de ampliación/mejora del sistema. Explicar como se comparten las variables y funciones entre varios scripts.