

Universidad de Valladolid – Departamento de Informática

Fundamentos de Informática II

Laboratorio - Programación del Shell de UNIX

Guía de Prácticas

Sesiones 6 y 7

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.

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. Tres en Raya

En todos los ejercicios para mostrar el estado del juego se debe mostrar una salida del siguiente tipo:

```
0 |   | X
---+---+---
X | X | 
---+---+---
0 |   |
```

A la salida de los scripts el programa debe devolver 0 si el usuario ha ganado o 1 si el sistema es el ganador.

1. Crear un script que permita simular el juego de *3 en Raya*. El script debe estar programado de forma tal que al menos consiga siempre un empate. El usuario inicia el juego.
2. Crear un script que permita simular el juego de *3 en Raya*. El script debe estar programado de forma tal que al menos consiga siempre un empate. El sistema inicia el juego en forma aleatoria.

3. Modifica el script anterior para que todas las jugadas del sistema sean aleatorias.

2. Miscelánea

1. Crear un guión que reciba como parámetro un fichero del directorio actual o el nombre de un directorio y que:
 - Compruebe que el número de parámetros es correcto.
 - Compruebe que el tipo de los parámetros es correcto.
 - Si ocurre alguno de los dos errores anteriores que informe al usuario de la sintaxis correcta y le permita continuar, leyendo el nombre por teclado.
 - Si el parámetro es correcto entonces:
 - Si es un fichero que informe del nombre y tamaño en bytes del fichero.
 - Si es un directorio, que informe del tamaño de todos los archivos del mismo que no sean directorios.
2. Crear un script que sume todos los argumentos recibidos, si es que son numéricos, informando del resultado.
3. Crear un script que sume todos los argumentos recibidos si es que son numéricos, en caso contrario que los vaya concatenando en una cadena alfanumérica.
4. Realizar un script que permita mostrar el número de sesiones abiertas por un usuario en el momento de ser ejecutado. El nombre del usuario debe pasarse al script como parámetro. En caso de que no se trate de un usuario registrado en el sistema el script deberá informar esa incidencia y terminar con código de salida 2.
5. Realizar un script para que un usuario introduzca una clave y que luego ésta sea adivinada por otro. El segundo usuario sólo debe poder salir del programa si adivina la clave o introduce la cadena `^BANDONO`". El programa no debe ser interrumpido con la combinación `Ctrl-C`. ¿Qué ocurre si se intenta salir utilizando `Ctrl-D`?
6. Realiza un guión que compruebe si dos archivos que se pasan como parámetros son idénticos, leyéndolos línea a línea.
7. Usa la orden `trap` para realizar el seguimiento del valor contenido en una variable de un proceso shell. El proceso debe asignar al menos tres valores sucesivos a la variable y al final mostrar el ultimo valor asignado a la variable, pero con ayuda de la orden `trap` el usuario debe ser informado de todos los cambios de valor de la variable. Puedes utilizar el siguiente guión como ejemplo.

```
#!/bin/bash

trap 'echo "TRAZA-VARIABLE> \${variable} = \"\${variable}\""' DEBUG #
Debe mostrar el valor de $variable después de cada comando.

variable=29

echo "Nada mas inicializarla: \"\${variable}\" vale $variable."

let "variable *= 3" echo "Despues de multiplicarla \"\${variable}\" por 3."

exit $?
```

8. Desarrollar un pequeño sistema basado en varios scripts que permita añadir un prefijo a los ficheros que se encuentren en un directorio determinado. Por ejemplo: cambiar el nombre de un fichero *datos.txt* a *marzo-datos.txt*. El sistema debe permitir detectar cuando el proceso no ha sido debidamente terminado o que permita anular los efectos del proceso de renombrado.
- El usuario deberá interactuar con el sistema a través de una pantalla que presente el estado del sistema y un menú de opciones. Por ejemplo, al inicio el sistema mostrará:

SISTEMA DE PREFIJOS

```
Directorio de trabajo:
Cadena prefijo:
Número de ficheros procesados: 0
Estado del último proceso: NO DETERMINADO
Ultimo proceso realizado: NINGUNO

1. Introducir directorio de trabajo
2. Introducir o modificar prefijo
3. Realizar proceso de renombrado
4. Retroceder el último proceso de renombrado
5. Verificar último proceso
6. Salir
```

Introduzca selección:

- El prefijo introducido debe ser una cadena no nula, que no contenga ningún carácter especial (por ejemplo: *, /, |, ? , etc).
- Se debe verificar que el directorio introducido exista y que no esté vacío. Además debe verificarse que el usuario tiene derechos para cambiar el nombre a los ficheros.
- Una vez iniciado un proceso el sistema debe mostrar un mensaje adecuado. Por ejemplo *“En proceso”*.
- Después de realizar un proceso de renombrado con normalidad en un directorio con 27 ficheros, la pantalla del sistema mostrará:

SISTEMA DE PREFIJOS

Directorio de trabajo: /usr/luis16/ejemplos/
Cadena prefijo: marzo-
Número de ficheros procesados: 27
Estado del último proceso: TERMINADO
Ultimo proceso realizado: RENOMBRADO

1. Introducir directorio de trabajo
2. Introducir o modificar prefijo
3. Realizar proceso de renombrado
4. Retroceder el último proceso de renombrado
5. Verificar último proceso
6. Salir

Introduzca selección:

- Si se selecciona anular el anterior proceso se debe mostrar un mensaje adecuado mientras se realiza esa tarea. A su finalización la pantalla deberá ser, por ejemplo:

SISTEMA DE PREFIJOS

Directorio de trabajo: /usr/luis16/ejemplos/
Cadena prefijo: marzo-
Número de ficheros procesados: 27
Estado del último proceso: TERMINADO
Ultimo proceso realizado: ANULAR renombrado

1. Introducir directorio de trabajo
2. Introducir o modificar prefijo
3. Realizar proceso de renombrado
4. Retroceder el último proceso de renombrado
5. Verificar último proceso
6. Salir

Introduzca selección:

- En otra situación, el usuario podría entrar al sistema y desear simplemente saber como se realizó el último proceso. Suponiendo que el último proceso fue interrumpido, la secuencia de acciones podría ser:

Bienvenido al sistema....

SISTEMA DE PREFIJOS

Directorio de trabajo:
Cadena prefijo:
Número de ficheros procesados: 0

Estado del último proceso: NO DETERMINADO
 Ultimo proceso realizado: RENOMBRADO

1. Introducir directorio de trabajo
2. Introducir o modificar prefijo
3. Realizar proceso de renombrado
4. Retroceder el último proceso de renombrado
5. Verificar último proceso
6. Salir

Introduzca selección: 5

SISTEMA DE PREFIJOS

Directorio de trabajo: /usr/luis16/temporal/
 Cadena prefijo: abril-
 Número de ficheros procesados: 13
 Estado del último proceso: INTERRUMPIDO
 Ultimo proceso realizado: RENOMBRADO

1. Introducir directorio de trabajo
2. Introducir o modificar prefijo
3. Realizar proceso de renombrado
4. Retroceder el último proceso de renombrado
5. Verificar último proceso
6. Salir

Introduzca selección:

- El usuario ahora podría desear revertir el proceso. La secuencia de acciones podría ser:

SISTEMA DE PREFIJOS

Directorio de trabajo: /usr/luis16/temporal/
 Cadena prefijo: marzo-
 Número de ficheros procesados: 13
 Estado del último proceso: INTERRUMPIDO
 Ultimo proceso realizado: RENOMBRADO

1. Introducir directorio de trabajo
2. Introducir o modificar prefijo
3. Realizar proceso de renombrado
4. Retroceder el último proceso de renombrado
5. Verificar último proceso
6. Salir

Introduzca selección: 4

En proceso ... ANULANDO renombrado ... Terminado

SISTEMA DE PREFIJOS

Directorio de trabajo: /usr/luis16/temporal/
Cadena prefijo:
Número de ficheros procesados: 13
Estado del último proceso: TERMINADO
Ultimo proceso realizado: ANULAR renombrado

1. Introducir directorio de trabajo
2. Introducir o modificar prefijo
3. Realizar proceso de renombrado
4. Retroceder el último proceso de renombrado
5. Verificar último proceso
6. Salir

Introduzca selección: 6

HASTA PRONTO.

- Se pueden usar los scripts y ficheros temporales que se consideren necesarios.
9. Realizar un script que permita simular el lanzamiento de n dados. El parámetro n no será obligatorio, si no es proporcionado se asumirá un valor por defecto de 1; si el parámetro es proporcionado deberá ser un entero en el intervalo 1 a 100.
 10. Realizar un script que permita jugar a los dados a dos jugadores.