

## Practical 'Shell - entrenamiento - Comando interno basado en script'

Your team identifier is: SO\_SA55

**Practical's name:** Shell - entrenamiento - Comando interno basado en script

A continuación se describen especificaciones a realizar sobre el shell básico entregado.

¡Atención!:

- Las modificaciones implementadas **no debe afectar a las especificaciones básicas del shell**, que deben seguir funcionando correctamente.
- Si una especificación requiere la impresión de mensajes, no se dará por correcta si no se ajustan exactamente al formato especificado.
- Salvo que se diga otra cosa, los mensajes a imprimir deben estar libres de códigos de color u otros de control adicionales para el terminal y se imprimirán por la salida estándar.

En este ejercicio se pide construir un comando interno basado en la invocación de un *script bash*. El ejercicio consta de dos partes, la primera desarrollar el script según especificaciones; la segunda es añadir un comando interno que invoque al script para realizar su cometido.

1. Se trata de elaborar un *shell script* con intérprete *bash* (*shebang*) llamado `cuenta fich.sh` que cuente el número de ficheros (que no directorios) que hay en el directorio de trabajo actual y que empiecen por un prefijo. El script tomará uno o cero argumentos. Si es uno, dicho argumento es el prefijo. Si no hay, entonces contará todos los ficheros (que no directorios) que hay en directorio actual. El script imprimirá este mensaje, donde el símbolo `<N>` indica el número de ficheros:

Número de ficheros encontrados: `<N>`

El script devolverá como valor de retorno 0 si se encontró algún fichero (`N>0`) y 1 en caso contrario (`N=0`).

☞ Para poder monitorizar el script, se efectuará una espera de 2 segundos antes de salir.

No olvidar poner permisos de ejecución a dicho script (`chmod +x cuenta fich.sh`)

2. Una vez elaborado el anterior script y comprobado su correcto funcionamiento, se pide añadir al shell un nuevo comando interno `fico` (*file count*) que invocando al script anterior realice la cuenta de los ficheros tal como se ha descrito.

Debe observarse que:

- El comando interno puede ejecutarse en foreground y background
- En caso de que se ejecute en background o se suspenda su ejecución ha de mostrarse en listado de jobs como cualquier otro comando, pero ojo, su nombre es `fico` (no el nombre del script)

Aquí se muestra unos ejemplos, del comportamiento esperado para esta propuesta:

```
COMMAND-> rm -f aaa*      # Borramos ficheros para empezar el ejemplo
COMMAND-> rm -f bbb*

COMMAND-> touch aaa.bbb # creamos unos ficheros que empiezan por aaa
COMMAND-> touch aaa.abc
COMMAND-> touch aaa.xyz

COMMAND-> fico aaa      # Contamos ficheros que empiezan por aaa
                        # el comando interno debe invocar correctamene al script
Número de ficheros encontrados: 3

COMMAND-> fico aaa.bbb
Número de ficheros encontrados: 1

COMMAND-> fico bbb
Número de ficheros encontrados: 0

COMMAND-> fico          # Sin argumentos cuenta el num. total de ficheros
                        # (no directorios) en el directorio de trabajo
                        # Vamos a suponer que son 23 en este ejemplo
Número de ficheros encontrados: 23
```

Como se ha descrito el comando interno puede estar en bg o suspendido (ya que realmente corresponde a la ejecución de un *script bash*)

```
# Probamos a ejecutar en bg y suspendidos
# Nótese que el retardo permite monitorizar durante unos segundos

COMMAND-> fico aaa    # Suspendemos este antes de que acabe
^Z
COMMAND-> fico        # Suspendemos este también
^Z
COMMAND-> fico a &    # bg

COMMAND-> jobs        # si tecleamos rápido se verán 3 jobs
                     # uno en bg y dos suspendidos

COMMAND->jobs
Contents of joblist:
[1] pid: 744204, command: fico, state: Stopped
[2] pid: 744218, command: fico, state: Stopped
[3] pid: 744221, command: fico, state: Background
```

Para que el ejercicio se considere correcto no pueden quedar hijos *zombies* en ningún caso.  
Y no olvidar que, si se reserva espacio para cadenas dinámicamente (`malloc`, `strdup`, ...), debe liberarse adecuadamente dicho espacio en el momento que ya no se necesite.

### File upload

The project filename must be **examenSO\_SA55.zip**

NOTE: Pack all your project files in the ZIP and upload it

Maximum size for new files: 5MB

You can drag and drop files here to add them.

Accepted file types:

Archive (ZIP) .zip

Maximum size for file examenSO\_SA55.zip: 5MB

«

»