

Implantación de Sistemas Operativos – BATCH

EN LA REALIZACIÓN DE TODOS LOS EJERCICIOS SE DEBE EVITAR EL ECO DE LAS ÓRDENES PROPIAS DE CADA UNO DE LOS PROGRAMAS, DEBEN ESTAR COMENTADOS, SE DEBEN MOSTRAR MENSAJES PARA INTERACTUAR CON EL USUARIO Y SE DEBEN EJECUTAR CORRECTAMENTE DESDE CUALQUIER DIRECTORIO ACTIVO.

Ejercicio 1:

Escribe un fichero .bat que escriba en pantalla un saludo. El .bat debe estar comentado, y no debe Mostrar en pantalla el “eco” de las órdenes del propio fichero. Debe pedir la pulsación de una tecla para salir del programa.

Ejercicio 2:

Muestra en pantalla variables definidas por el sistema, para visualizar el nombre del equipo, el nombre del usuario y el del dominio si existe.

Ejercicio 3:

Escribe un programa que pida el nombre del usuario que lo ejecuta por teclado, luego lo debe mostrar por pantalla y dar un mensaje de agradecimiento.

Ejercicio 4:

Escribe un programa que recibe dos parámetros y que la salida sea una palabra compuesta de las palabras simples que se introducen, hay que sacar dos mensajes uno tal como están pedidos los parámetros y otro al revés.

Ejercicio 5:

Explica la salida en pantalla del siguiente código.

```
REM ESTRUCTURA SECUENCIAL
@ECHO OFF
ECHO Hola
GOTO Ir
ECHO y
ECHO adiós
:IR
ECHO ¿Qué
ECHO tal?
```

Ejercicio 6:

Reescribe el código anterior para que muestre en pantalla la frase “Hola y adiós”.

Ejercicio 7:

Usando el comando GOTO, escribe un programa que compruebe si el fichero bat tiene o no un parámetro de entrada y muestre en pantalla dicho parámetro si existe o un mensaje avisando que no tiene.

Ejercicio 8:

Búsqueda de un fichero llamado pepe.txt. Si existe visualizaremos el contenido del fichero, si no existe visualizaremos un mensaje de error.

Ejercicio 9:

Realizar una búsqueda de cualquier fichero introducido como parámetro mostrando los mensajes adecuados. Usa la orden IF EXIST ...

Ejercicio 10:

Realizar una búsqueda de cualquier fichero introducido como parámetro, pero usando el NOT en el IF.

Ejercicio 11:

Crear un bat que multiplique tres números metidos como parámetros.

Ejercicio 12:

Un programa que te dice si has metido parámetros al ejecutar el batch o no y si los has metido los muestra.

Ejercicio 13:

Bucle infinito y uso de SHIFT

En este primero hemos creado un bucle infinito y solo se puede parar la ejecución del programa con Ctrl + C

```
@ECHO OFF
```

```
REM Bucle infinito
```

```
:INICIO
```

```
ECHO Hola mundo
```

```
ECHO para cancelar Ctrl + C
```

```
GOTO INICIO
```

Este también es un bucle infinito que muestra el parámetro introducido.

```
@ECHO OFF
```

```
REM Bucle infinito
```

```
:INICIO
```

```
ECHO %1
```

```
GOTO INICIO
```

Este es un bucle que muestra los parámetros introducidos, pero se para cuando los ha mostrado todos.

```
@ECHO OFF
```

```
REM Muestra los parámetros introducidos de 1 en 1 y sin límite de número de parámetros
```

```
:INICIO
```

```
IF "%1" == "" GOTO FIN
```

```
REM si no metemos ningún parámetro finaliza el programa
```

```
ECHO %1
```

```
SHIFT
```

```
REM con SHIFT desplazamos el contenedor de variables una posición
```

```
REM solo visualiza los parámetros introducidos de uno en uno y aunque sea más de 9
```

```
GOTO INICIO
```

```
:FIN
```

```
ECHO Proceso terminado
```

Ejercicio 14:

Crear un programa que cree un grupo local y meta en él los usuarios. El grupo y usuarios se meterán como parámetros al invocar el programa. Recuerda los comandos net (net help, net user /? ...)

1. Crear Grupo
2. Crear Usuario
3. Meter al usuario al Grupo

Ejercicio 15:

Uso del comando FOR, muestra el contenido de unos archivos.

```
@ECHO OFF
```

```
REM
```

```
FOR %%I IN (f1.txt f2.txt f3.txt f4.txt) DO TYPE %%I
```

Para tantos elementos como haya entre () ejecutar un comando (en este caso un Type de cada fichero) Va recorriendo los elementos o variables entre paréntesis (cometido del FOR IN) y ejecutando el comando que viene después del DO.

Si se quiere más de una acción necesitamos subprogramas.

Realiza lo mismo que el anterior programa, pero que busque cualquier archivo terminado en .bat

Ejercicio 16:

Escribe un programa principal que realice las acciones contempladas en el programa esclavo o subprograma y regrese al programa principal para seguir ejecutándose en el punto en el que estaba.

Ejercicio 17:

Diseña un fichero batch que acepte parámetros y según el número de estos, visualice solamente uno de los siguientes mensajes.

“Se necesitan dos parámetros.” , si no se recibe ningún parámetro.

“Un parámetro es insuficiente, se necesitan dos.” , si solo se recibe un parámetro.

“Número de parámetros correcto.” , si se reciben dos parámetros.

“Demasiados parámetros.” , si se reciben más de dos.

Ejercicio 18:

Diseña un script que se llame menú.bat y simule el siguiente menú:

(L)impiar pantalla

(C)omprobar disco

(V)ersión del sistema operativo

(S)alir

Se debe ejecutar la utilidad de sistema adecuada a cada opción.