

El bucle FOR en MS-DOS/Windows

Introducción

Ejecuta el **comando** para cada uno de los elementos especificados en la **lista**. Su sintaxis es distinta dependiendo de si se ejecuta en un fichero .bat o desde línea de comandos (el cambio es la forma de llamar a la variable del bucle).

```
FOR %nombre_variable IN (lista) DO comando ← desde línea de comandos
FOR %%nombre_variable IN (lista) DO comando ← dentro de archivos BAT
```

- ✓ **nombre_variable** → Un nombre de variable de un **único carácter**, que se utilizará para recorrer los elementos o archivos de la lista. La **variable diferencia si es MAYÚSCULA o minúscula**.
- ✓ **lista** → Especifica un **conjunto de elementos o archivos separados por coma, punto y coma o espacio en blanco**. Si en la lista aparecen comodines (? o *) el comando se ejecutará para cada uno de los archivos que cumplan la expresión regular.
- ✓ **comando** → Comando MS-DOS a ejecutar con cada elemento de la lista.

Ejemplo 1 - Desde línea de comandos

```
W:\>FOR %X IN (uno dos tres cuatro) DO @ECHO %X
uno
dos
tres
cuatro
```

Ejemplo 2 - Desde un archivo bat llamado repite.bat

```
W:\>MORE > repite.bat
@ECHO OFF
FOR %%X IN (%*) DO ECHO %%X
^Z
W:\>repite ms-dos es "bueno, bonito y barato"
ms-dos
es
"bueno, bonito y barato"
```

Cuando aparece un ***** en la lista, lo interpreta como **comodín** y antes de aplicar el comando **se crea una lista de archivos que cumpla con la expresión regular**. En la elaboración de la lista sólo utiliza por defecto archivos, **NO directorios**. En el último ejemplo se han movido imágenes con formatos gráficos a la papelera del disco C:.

```
C:\pruebas>FOR %F IN (*) DO @ECHO %F
punto.NET.txt
Frases.odt
net.jpg
ejercil.gif
...
```

```
C:\pruebas>FOR %F IN (*.png, *.jpg, *.gif) DO MOVE %F C:\Recycled
```

El siguiente archivo .bat nos **lista todos los archivos ejecutables** (.exe, .com y .bat) del directorio pasado desde línea de comandos. Si no pasamos argumento, mostraría los del raíz.

```
@ECHO OFF
FOR %%X IN(%1\*.exe %1\*.com %1\*.bat) DO ECHO %%X
```

Parámetro del comando FOR

El comando FOR admite una serie de parámetros que amplían su funcionalidad:

- ✓ **Recursividad.**

```
FOR /R [ruta] %%V IN (lista) DO comando
```

- ✓ **Directorios.**

```
FOR /D %%V IN (lista) DO comando
```

- ✓ **Lista con contador.**

```
FOR /L %%V IN (inicio, incremento, fin) DO comando
```

- ✓ **Recorrido de tokens en líneas de texto.**

```
FOR /F ["tokens=... delims=..."] %%V IN (archivo | 'comando' |
"cadena") DO comando
```

Con el **parámetro /R** ejecuta el comando **recursivamente**.

```
C:\>FOR /R %X IN (*) DO @ECHO %X
C:\bat2exe.com
C:\editpad.zip
C:\LANGUAGE.txt
...
C:\WINNT\discover.exe
C:\WINNT\explorer.scf
C:\WINNT\winhlp32.exe
...
C:\WINNT\system32\mscoree.dll
C:\WINNT\system32\msorcl32.dll
C:\WINNT\system32\aaaamon.dll
...
```

Visualizamos todas los **archivos .dll y .exe** a partir de C:\WINNT.

```
C:\>FOR /R C:\WINNT %X IN (*.dll *.exe) DO @ECHO %X
C:\winnt\twain.dll
C:\winnt\twain_32.dll
C:\winnt\vmreg32.dll
C:\winnt\system32\mscoree.dll
C:\winnt\system32\msorcl32.dll
C:\winnt\system32\mmc.exe
C:\winnt\system32\wscript.exe
C:\winnt\system32\javaws.exe
...
```

Con el **parámetro /D** recorre los **directorios**, en lugar de los **ficheros**:

```
C:\>FOR /D %X IN (*) DO @ECHO %X
WINNT
Documents and Settings
Archivos de programa
prog
...
```

Con el **parámetro /L** podemos crear el **típico bucle por contador de programación**:

```
FOR /L %nombre_variable IN (inicio salto final) DO comando
```

Podemos separar **inicio**, **salto** y **final** por **comas (,)** o **punto y coma (;)** también, aparte de por un espacio como en el ejemplo. Con el siguiente archivo batch llamado **crea.bat** creamos archivos .txt vacíos, cuya **cantidad** y **nombre** especificamos desde línea de comandos:

```
@ECHO OFF
FOR /L %%A IN (1 1 %1) DO ECHO 2> %2%%A.txt > NUL
```

```
C:\pruebas>crea 5 churro
C:\pruebas>DIR CH* /B
    churro1.txt
    churro2.txt
    churro3.txt
    churro4.txt
    churro5.txt
```

El siguiente archivo .bat llamado **pingred.bat** hace un **ping a broadcast**, mostrando **sólo las direcciones de aquellos que contestan**:

```
@ECHO OFF
SET RED=192.168.1.
FOR /L %%H IN (1, 1, 255) DO PING -n 1 %RED%%%H | FIND "TTL="
```

```
C:\pruebas>pingred
Respuesta desde 192.168.1.1: bytes=32 tiempo=1ms TTL=128
Respuesta desde 192.168.1.13: bytes=32 tiempo<10ms TTL=64
Respuesta desde 192.168.1.22: bytes=32 tiempo<10ms TTL=64
Respuesta desde 192.168.1.25: bytes=32 tiempo<10ms TTL=64
```

Con el **parámetro /F** podemos recorrer:

- a) Un archivo.
- b) El resultado de la ejecución de un comando.
- c) Una cadena.

El parámetro /F para recorrer archivos

```
X:\>type fich.txt
El corazón y el cerebro, las dos grandes vísceras
que se llevan la primacía de los sentimientos y de
la razón, son las que presiden el arranque de las
coplas de Jorge Manrique a la muerte de su padre.
```

```
X:\>FOR /F %X IN (fich.txt) DO @ECHO %X
El
que
```

la
coplas

El bucle for va recorriendo todas las líneas, y **cada línea se ha dividido en tokens** (un token es un elemento, en este caso **por defecto: token=palabra**). La variable x del bucle almacena el primer token de cada línea.

Podemos **seleccionar varios tokens** mediante la **cláusula tokens=**,. Los distintos tokens se irán guardando en **nombres de variable alfabéticamente consecutivas a partir de la variable utilizada en el for**. En el siguiente ejemplo nos quedamos con los tokens (palabras) 1, 3 y 5 de cada línea:

```
X:\>FOR /F "tokens=1,3,5" %I IN (fich.txt) DO @ECHO %I - %J - %K
El - y - cerebro,
que - llevan - primacía
la - son - que
coplas - Jorge - a
```

Podemos escoger **rangos de tokens**. En el siguiente ejemplo nos quedamos con las 3 primeras palabras (rango de la 1 a la 3 y con la 5).

```
X:\>FOR /F "tokens=1-3,5" %A IN (fich.txt) DO @ECHO %A %B %C - %D
El corazón y - cerebro,
que se llevan - primacía
la razón, son - que
coplas de Jorge - a
```

Y si deseamos la línea completa a partir de la séptima palabra:

```
X:\>FOR /F "tokens=7*" %A IN (fich.txt) DO @ECHO %A %B
dos grandes vísceras
los sentimientos y de
el arranque de las
muerte de su padre.
```

Y si queremos toda la línea, seleccionaremos todos los tokens de cada línea:

```
X:\>FOR /F "tokens=*" %M IN (fich.txt) DO @ECHO [ %M ]
[ El corazón y el cerebro, las dos grandes vísceras ]
[ que se llevan la primacía de los sentimientos y de ]
[ la razón, son las que presiden el arranque de las ]
[ coplas de Jorge Manrique a la muerte de su padre. ]
```

En los ejemplos anteriores **los tokens coincidían con las palabras**, pues **para separarlos se han utilizado el espacio en blanco y la tabulación**. Podemos modificar los delimitadores con la cláusula **delims=**. En el siguiente ejemplo anulamos los delimitadores y obtenemos lo mismo de antes:

```
X:\>FOR /F "delims=" %K IN (fich.txt) DO @ECHO [ %K ]
[ El corazón y el cerebro, las dos grandes vísceras ]
[ que se llevan la primacía de los sentimientos y de ]
[ la razón, son las que presiden el arranque de las ]
[ coplas de Jorge Manrique a la muerte de su padre. ]
```

Utilizamos como delimitadores **los signos de puntuación (punto, coma, punto y coma, y dos puntos)** -no debe haber espacios entre ellos- y nos quedamos con el **primer token**:

```
X:\>FOR /F "delims=.,;:" %M IN (fich.txt) DO @ECHO %M
El corazón y el cerebro
que se llevan la primacía de los sentimientos y de
```

la razón
coplas de Jorge Manrique a la muerte de su padre

Podemos **combinar delimitadores y tokens**:

```
FOR /F "tokens=1,2,3 delims=.,;" %M IN (fich.txt) DO @ECHO %M %N %O
```

Y utilizar **otras cláusulas**, como **skip=número**, que se saltaría las **n (número) primeras líneas del archivo**, empezando a procesar en la **n+1**.

O la cláusula **eol=carácter**, que **interpreta las líneas que comienzan por ese carácter** como líneas de **comentario** y **no serán procesadas**, o **useback** que cambia la semántica de la lista: (``comando``), (`'cadena'`), (`archivo`) (`"archivo1 archivo2 archivo3"`).

El parámetro /F con comandos

El funcionamiento es **similar al de los ficheros**, únicamente que **tendremos que encerrar el comando con comillas simples**:

```
X:\>FOR /F "tokens=15 delims=: " %X IN ('ipconfig ^| find "IP"') DO SET myIP=%X
x:\>set myIP
myIP=192.168.1.33
```

Observar el **carácter de escape** `^` necesario para que el pipe o tubería lo interprete dentro del comando en lugar de ser interpretado para el FOR.

El parámetro /F para recorrer cadenas

Con el siguiente fichero .bat llamado rdate.bat **añadimos la fecha a un nombre de archivo**:

```
FOR /F "tokens=2-4 delims=/ " %%W IN ("%date%") DO REN %1 %%Y%%X%%W-%%1

X:\>rdate backup.bat
```

Su ejecución sería **ren backup.bat 20060219-backup.bat**, o sea, **renombr**a el fichero con la fecha.

Formatos de salida

Podemos especificar un formato para la variable incluida en el bucle for donde **f es la ruta completa** y **s el nombre corto** (no más de 8 caracteres para el nombre y 3 para la extensión).

```
C:\Documents and Settings\Ejemplo>FOR %X IN (*) DO @ECHO %~FSX
C:\DOCUME~1\Ejemplo\mueva.bat
C:\DOCUME~1\Ejemplo\EXCURS~1.TXT
C:\DOCUME~1\Ejemplo\EXCURS~2.TXT
C:\DOCUME~1\Ejemplo\EXCURS~3.TXT
C:\DOCUME~1\Ejemplo\ABC~1.TXT
C:\DOCUME~1\Ejemplo\DIAD~1.TXT
C:\DOCUME~1\Ejemplo\HID~1
C:\DOCUME~1\Ejemplo\FONTS~1.CAC
```

```
C:\Documents and Settings\Ejemplo>FOR %X IN (*) DO @ECHO %X
mueve.bat
Excursiones.txt
Excursiones del año 2004.txt
Excursiones del año 2005.txt
A B C.txt
dia D.txt
.hid
.fonts.cache-1
```

O simplemente **%~X**, que **elimina las comillas** (si las tuviera) como vimos arriba. Podemos consultarlos todos ejecutando **FOR /?** o **HELP FOR**.