## Dos: script dell'interprete dei comandi



Uno script dell'interprete dei comandi, conosciuto solitamente con il nome di file *batch*, potrebbe essere definito come un file di testo normale in cui può essere indicato un elenco di comandi da eseguire. Tuttavia, questi script consentono l'uso anche di strutture di controllo elementari, per cui si possono realizzare dei programmi molto semplici, senza troppe pretese.

È interessante osservare che questi script vengono individuati solo attraverso l'estensione che ha il nome: '`.BAT`'. Inoltre, non esiste la necessità di renderli «eseguibili» come si fa nei sistemi Unix.

### Parametri, variabili ed espansione

Gli script dell'interprete dei comandi hanno accesso agli argomenti che vengono loro forniti. Si possono gestire solo nove di questi argomenti alla volta, attraverso i parametri posizionali relativi, da '**`%1`**' a '**`%9`**'. Come avviene nelle shell Unix, è disponibile il comando interno '**`SHIFT`**' per fare scorrere in avanti gli argomenti nei parametri disponibili.

Bisogna ricordare che in Dos i caratteri jolly non vengono espansi dalla shell, per cui la limitazione a soli nove parametri posizionali, non dovrebbe costituire un problema.

Nell'ambito di uno script possono essere dichiarate e utilizzate delle variabili di ambiente. È già stato mostrato in precedenza l'uso del comando '**`SET`**' per impostare o eliminare le variabili di ambiente. Per fare riferimento al contenuto di una variabile, si usa la notazione seguente:

```
%nome_variabile%
```

L'esempio seguente rappresenta il caso tipico di estensione di un percorso di ricerca degli eseguibili, quando si ritiene che la variabile '**`PATH`**' sia già stata usata:

```
SET PATH=%PATH%;C:\PIPPO
```

### Chiamate di altri script

Tradizionalmente, il Dos ha un baco molto grave, ormai divenuto una caratteristica fondamentale, riguardante l'avvio di script all'interno di altri script. In generale, quando si chiama un programma che in realtà corrisponde a uno script, al termine di questo non riprende l'esecuzione di quello chiamante. Per ottenere la ripresa dell'interpretazione dello script di partenza occorre usare il comando speciale '**`CALL`**'.

```
CALL nome_script [argomenti_dello_script]
```

## Strutture di controllo

Le strutture di controllo per la programmazione attraverso gli script dell'interprete dei comandi sono molto limitate. È disponibile una struttura condizionale semplificata e un ciclo di scansione di file, che vengono descritti brevemente.

### IF

```
IF [NOT] ERRORLEVEL valore_di_uscita_ultimo_comando comando
```

```
IF [NOT] stringa_1==stringa_2 comando
```

```
IF [NOT] EXIST file comando
```

La struttura condizionale degli script dell'interprete dei comandi Dos è in pratica un comando interno dello stesso interprete. Come si può vedere dagli schemi sintattici, viene fornita una condizione che può essere invertita con la parola chiave '**NOT**' e il risultato è solo l'esecuzione di un altro comando se la condizione risulta vera.

Nel primo caso, la condizione si riferisce alla verifica del valore di uscita dell'ultimo comando eseguito. La condizione si verifica se il numero indicato è inferiore o uguale al valore restituito effettivamente da tale comando; nel secondo, la condizione si verifica se le due stringhe (non delimitate) sono identiche; nel terzo si verifica la condizione se il file indicato esiste effettivamente.

Segue la descrizione di alcuni esempi.

- `IF ERRORLEVEL 1 GOTO :errore`

  Se il comando precedente ha restituito un valore maggiore o uguale a uno, salta all'etichetta '**:errore**'.

- `IF %1==ciao ECHO L'argomento è corretto`

  In questo caso, se l'espansione del parametro '**%1**', corrispondente al primo argomento ricevuto all'avvio, si traduce nella stringa '**ciao**', viene emesso un messaggio per mezzo del comando '**ECHO**'.

- `IF %1x==x ECHO L'argomento è mancante`

  Quello che si vede è il trucco necessario per poter verificare se un parametro contiene la stringa nulla: si aggiunge una lettera, in questo caso una «x», verificando che la corrispondenza avvenga solo con la stessa lettera.

- `IF NOT EXIST LETTERA.TXT ECHO Scrivi! > LETTERA.TXT`

  Qui, se non esiste il file 'LETTERA.TXT' nella directory corrente, questo file viene creato attraverso il comando '**ECHO**' che invia il suo standard output verso un file con lo stesso nome.

### FOR

```
FOR [%]%x IN (nome…) DO comando [argomenti_del_comando]
```

Si tratta di un comando interno che svolge un ciclo di scansione di un gruppo di nomi, generalmente file, attraverso il quale viene creato un parametro variabile speciale, il cui nome si compone di una sola lettera, a cui viene assegnato a ogni ciclo uno dei nomi contenuti tra parentesi tonde. A ogni ciclo viene eseguito il comando, che a sua volta può fare uso del parametro.[1]

Quando viene usato all'interno di uno script dell'interprete dei comandi, il parametro viene indicato con due simboli di percentuale ('**%%**x'); al contrario, se il comando viene impartito dalla riga di comando, se ne usa uno solo.

Segue la descrizione di alcuni esempi.



- `FOR %A IN (uno due tre) DO ECHO %A`

  In questo modo, si ottiene la visualizzazione delle parole '**uno**', '**due**' e '**tre**'. In pratica, è come se fosse stato fatto:

- ```
  ECHO uno
  ECHO due
  ECHO tre
  ```

  Volendo fare la stessa cosa dalla riga di comando, è necessario il raddoppio del simbolo '**%**':

  `C:\>FOR %%A IN (uno due tre) DO ECHO %%A` [*Invio*]

- `FOR %A IN (*.TMP *.BAD) DO DEL %A`

  Cancella, uno a uno, tutti i file che terminano con le estensioni '.TMP' e '.BAD'.

### GOTO

```
GOTO etichetta
```

Gli script dell'interprete dei comandi dispongono dell'istruzione di salto incondizionato, non avendo di meglio. Anche questa istruzione può essere presa come un comando interno dell'interprete, con la differenza che non c'è modo di utilizzarlo al di fuori di uno script.

Nel corso di uno script del genere, possono apparire delle righe che contengono solo un'etichetta, nella forma:

```
:nome_etichetta
```

La posizione corrispondente a queste etichette può essere raggiunta con il comando '**GOTO**', che può fare riferimento solo al nome dell'etichetta, oppure a tutta l'etichetta, includendo anche i due punti.

Segue la descrizione di alcuni esempi.

- ```
  IF EXIST LETTERA.TXT GOTO riprendi
  ECHO Il file LETTERA.TXT è assente
  :riprendi
  ```

  In questo esempio, se il file 'LETTERA.TXT' esiste, si salta all'etichetta '**:riprendi**'; altrimenti si esegue il comando '**ECHO**'.

- ```
  IF EXIST LETTERA.TXT GOTO :riprendi
  ECHO Il file LETTERA.TXT è assente
  :riprendi
  ```

  Esattamente come nell'esempio precedente, con la differenza che il comando '**GOTO**' indica l'etichetta con i suoi due punti iniziali.

### Emulazione di un ciclo iterativo

Dal momento che non è disponibile una struttura di controllo per il ciclo iterativo, questo può essere ottenuto solo attraverso l'uso del comando '**GOTO**'. Vale la pena di mostrare in che modo si può ottenere tale risultato.

```
:etichetta_di_ingresso
IF condizione GOTO :etichetta_di_uscita
    …
    …
    …
    GOTO etichetta_di_ingresso
:etichetta_di_uscita
```

```
:etichetta_di_ingresso
    …
    …
    …
IF condizione GOTO :etichetta_di_ingresso
:etichetta_di_uscita
```

I due modelli sintattici mostrano due esempi di cicli iterativi. Nel primo caso si verifica una condizione, in base alla quale si decide se proseguire o se terminare il ciclo; nel secondo si esegue una volta il
```

ciclo e quindi si verifica una condizione per decidere se ripeterlo o se uscire.

## Comandi utili negli script

Alcuni comandi sono particolarmente utili all'interno di script dell'interprete dei comandi. Vengono descritti brevemente nelle sezioni seguenti.

### REM

```
REM commento
```

I commenti negli script dell'interprete dei comandi si indicano attraverso un comando apposito: '**REM**'. Il funzionamento è evidente: tutto quello che segue il comando, fino alla fine della riga, viene ignorato.

Alcune edizioni del Dos hanno introdotto anche l'uso del punto e virgola, come simbolo per indicare l'inizio di un commento. Segue un esempio tipico di utilizzo di questo comando:

```
REM
REM (c) 2000 Pinco pallino
REM
```

### ECHO

```
ECHO [ON|OFF]
```

```
ECHO stringa
```

Il comando interno '**ECHO**' ha un significato duplice: da una parte consente di visualizzare un testo; dall'altra controlla la visualizzazione dei comandi contenuti in uno script. Infatti, si distingue il fatto che l'eco dei comandi sia attivo o meno. utilizzando il comando '**ECHO**' senza argomenti, si ottiene l'informazione sul suo stato di attivazione. Di solito si disattiva l'eco dei comandi negli script.

Per disattivare l'eco di un comando particolare, senza disattivare l'eco in generale, basta inserire inizialmente il simbolo '@'.

Segue la descrizione di alcuni esempi.

- `@ECHO OFF`
  Disattiva l'eco dei comandi, facendo in modo che anche questo comando non venga visualizzato (si usa per questo il simbolo '@').
- `ECHO Premi un tasto per continuare`
  Mostra un messaggio per spiegare come comportarsi.

### PAUSE

```
PAUSE
```

Il comando interno '**PAUSE**' sospende l'esecuzione di uno script in attesa della pressione di un tasto. Il comando emette attraverso lo standard output un messaggio di avvertimento in tal senso. Di solito, per evitare di vedere tale messaggio, si ridirige lo standard output in un file nullo.

L'esempio seguente, prima mostra un messaggio in cui si avverte che per proseguire occorre premere un tasto, quindi si usa il comando '**PAUSE**' che sospende l'esecuzione dello script, senza però mostrare altri messaggi:

```
ECHO Premere un tasto per proseguire
PAUSE > C:\NULL
```



### CLS

```
CLS
```

Il comando interno '**CLS**' ripulisce lo schermo. Si utilizza senza argomenti.

### CHOICE

```
CHOICE [opzioni] [testo_di_invito]
```

Il comando '**CHOICE**' serve a presentare una richiesta per l'inserimento di una lettera, tra un elenco determinato. La pressione del tasto corrispondente alla lettera scelta, da parte dell'utilizzatore, provoca la conclusione del funzionamento di '**CHOICE**' che restituisce un valore corrispondente alla scelta: zero per la prima lettera, uno per la seconda,...

Si osservi che l'ultimo argomento rappresenta un messaggio che serve all'utente per comprendere il senso della scelta che sta facendo.

Tabella u182.15. Alcune opzioni.

| Opzione | Descrizione |
|---|---|
| `/C:`*lettera*[*lettera*...] | Permette di fissare l'elenco di lettere che possono essere usate nella risposta. Se non si indica questa opzione, la scelta avviene solo tra '**y**' e '**n**'. |
| `/N` | Questa opzione fa in modo di escludere la visualizzazione delle lettere che possono essere scelte. In questo modo si fa affidamento esclusivamente sul testo indicato come ultimo argomento. |
| `/S` | Distingue tra maiuscole e minuscole per quanto riguarda le lettere tra cui scegliere. |

L'esempio seguente, salta a un punto differente dello script in base alla scelta di una lettera da «a» a «f». Si osservi che non sarebbe possibile eseguire l'analisi secondo una sequenza differente, perché '**IF ERRORLEVEL**' prende in considerazione tutti i valori di uscita maggiori o uguali a quanto indicato nella condizione.

```
CHOICE /C:abcdef Inserisci una lettera
IF ERRORLEVEL 5 GOTO :f
IF ERRORLEVEL 4 GOTO :e
IF ERRORLEVEL 3 GOTO :d
IF ERRORLEVEL 2 GOTO :c
IF ERRORLEVEL 1 GOTO :b
IF ERRORLEVEL 0 GOTO :a
…
```

[1] Questo parametro assomiglia a una variabile di ambiente, ma non si comporta allo stesso modo. Si tratta di una particolarità del comando '**FOR**'.

2050