

GUIDA BATCH

Sommario

1. Cosa sono gli script Batch	pag. 2
2. La Sintassi	pag. 3
3. Su XP tutto diverso	pag. 4
4. Istruzione Echo	pag. 5
5. Istruzione Pause	pag. 7
6. Lo script Hello, World!	pag. 8
7. Istruzione Type e More	pag. 9
8. Leggere dalla riga di comando	pag. 10
9. Usare gli If per decidere a seconda dei parametri di comando	pag. 11
10. Muoversi nel programma	pag. 12
11. Le variabili di ambiente e i commenti	pag. 14
12. Istruzione Start	pag. 15
13. Lo script Autorun	pag. 16
14. Ricapitolando Parte I	pag. 17
15. Istruzione Color	pag. 18
16. Strutturiamo Batch	pag. 19
17. Internet e Batch Parte I	pag. 20
18. Internet e Batch Parte II	pag. 21
19. Usare le conoscenze su Internet e Batch	pag. 22
20. Istruzione Call	pag. 23
21. Istruzione Choice	pag. 24
22. Istruzione If Parte I	pag. 25
23. Istruzione If Parte II	pag. 26
24. Istruzione If Parte III	pag. 27
25. Istruzione If Parte IV	pag. 28
26. Istruzione If Parte V	pag. 29
27. Istruzione For	pag. 30
28. Appendice Utilizzo dei Filtri e degli Operatori	pag. 31
29. Risorse e strumenti utili	pag. 32

CAPITOLO 1

Cosa sono gli script Batch

Gli Script **Batch** apparvero con il debutto delle prime versioni di Windows. Di solito un file del genere, ha estensione ".BAT" ed è apprezzato per i seguenti motivi:

- Risparmia il lavoro di digitazione dei comandi DOS
- Consente di eseguire velocemente operazioni ridondanti

Possiamo definire uno script Batch, come anche gli script Python, non compilati, ma interpretati dalla Shell. La Shell è l'interprete dei comandi.

Questo tipo di script è così legato a Windows, tanto che viene eseguito il notissimo "AUTOEXEC.BAT" nella fase di bootstrap (operazioni che precedono l'avvio di Windows).

CAPITOLO 2

La Sintassi

Un file batch non ha una sintassi vera e propria, bensì usa i comandi del DOS. Per questo motivo non può essere definito un vero e proprio linguaggio di programmazione anche perchè, tra l'altro, non viene compilato ma direttamente interpretato dal DOS.

Come ottenere informazioni riguardo ai comandi?

Basta andare sul prompt (Start>Esegui>cmd) e digitare questa sintassi:

<comando>/?

Ad esempio: IF/?

CAPITOLO 3

Su XP tutto diverso

Io uso Windows XP Professional e devo dire che sono stato deluso dal punto di vista batch. Avendo prima Windows 98, ho notato che su XP mancano alcune funzioni importanti (ad esempio choice che consentiva di effettuare scelte), credo che sia possibile avere queste funzioni prendendole dalla cartella C:\Windows\COMMAND di Windows 98. Su Windows XP è tutto diverso, ci sono addirittura funzioni nuove come Shutdown (che consente di spegnere/riavviare il sistema) oppure Tasklist (che permette di visualizzare i processi attivi). Purtroppo questa limitazione è stata un po' drastica, direi...

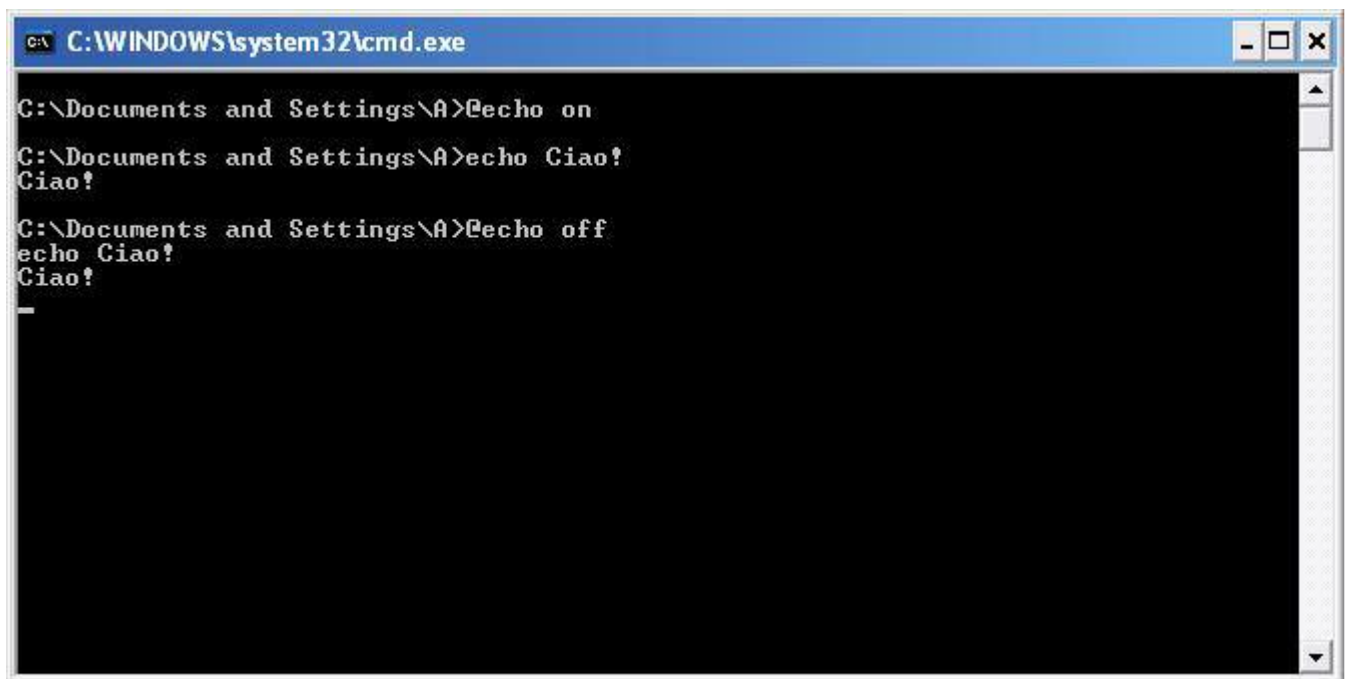
CAPITOLO 4

Istruzione Echo

Questa istruzione, è forse quella più importante. Ed ha due principali funzioni: quella di mostrare o nascondere il percorso in cui si sta lavorando, l'altra è quella di mostrare semplici messaggi a video. Altra cosa da dire sul batch, a condizione che l'echo sia spento, è che non viene mostrato a video la sintassi, ma solo l'output.

Accendere e spegnere l'echo:

Naturalmente quando apriamo il prompt l'echo è acceso, basta digitare "echo" senza parametri e sapremo se è attivo o meno. In questa immagine vedremo cosa accade quando "spegniamo" l'echo.



```
C:\WINDOWS\system32\cmd.exe

C:\Documents and Settings\A>@echo on
C:\Documents and Settings\A>echo Ciao!
Ciao!
C:\Documents and Settings\A>@echo off
echo Ciao!
Ciao!
```

Nel secondo caso vediamo che il percorso sparisce.

SINTASSI:

@echo on

Questa istruzione ci permette di lasciare acceso (ON) l'echo.

@echo off

Questa istruzione è il contrario di quella precedente, l'echo è spento (OFF).

Stampare un messaggio:

Le capacità di Echo non terminano qui, possiamo anche stampare un testo sullo schermo, o meglio come dice la sua guida, "Visualizza messaggi [...]". Se spegneremo l'echo, vedremo solo e solamente il testo già "stampato".

SINTASSI:

echo <Messaggio>

Dove messaggio è il testo da stampare. Visualizza il testo sullo schermo.

echo.

Visualizza una riga vuota.

Ho fornito (e anche ben spiegato) come usare questa istruzione, vi lascio ora con una screen che vi farà vedere la guida che dà il DOS su questo comando.



```
C:\WINDOWS\system32\cmd.exe

C:\Documents and Settings\A>echo/?
Visualizza messaggi, oppure attiva/disattiva la ripetizione dei comandi.

    ECHO [ON|OFF]
    ECHO [messaggio]

Digitare ECHO senza parametri per visualizzare l'impostazione corrente.
C:\Documents and Settings\A>
```

CAPITOLO 5

Istruzione Pause

Prima abbiamo visto cos'è l'istruzione Echo, adesso, prima di fare i primi esempi, spiegherò l'istruzione Pause, per un motivo che dirò in seguito.

Se digitiamo al prompt "pause/?", apparirà "Interrompe l'elaborazione di un programma batch [...]", ma al solito mio, desidero spiegare in modo più approfondito.


Pause, come si intuisce, mette in pausa il "flow", cioè la sequenza delle informazioni, che poi ripartiranno quando si preme un tasto. Nel prompt apparirà "Premere un tasto per continuare..."

SINTASSI:

pause

Mette in pausa il programma visualizzando "Premere un tasto per continuare"

Alta "personalizzabilità":



```
C:\Documents and Settings\A>pause
Premere un tasto per continuare . . .
```

Non proprio possiamo personalizzare il pause, ma possiamo nascondere. Metteremo prima di esso un echo col nostro messaggio personale.

SINTASSI:

pause>nul

Mettendo il >nul subito dopo il comando, nasconderemo l'output.

Adesso per mettere in pausa useremo questo codice:

CODICE:

echo Testochevoglio

pause>nul

Adesso vi chiederete perchè ho parlato di questo, prima di continuare con la pratica... per un semplice motivo, se facciamo file con l'echo e li "lanciamo", non vedremo una mazza, perchè lo script terminerebbe subito, quindi per evitare questo, inseriremo il pause...

CAPITOLO 6

Lo Script Hello, World!

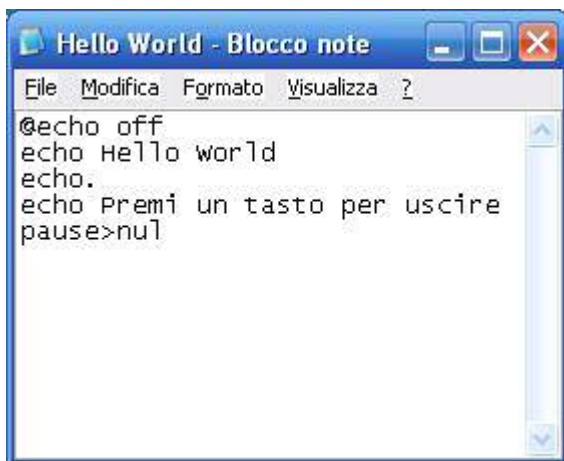
Dopo che abbiamo appreso correttamente le istruzioni "Echo" e "Pause", possiamo addentrarci nella programmazione vera e propria.

Per fare tutto ciò basta il Blocco note.

1. Apriamo il blocco note
2. Inseriamo la sintassi

CODICE:

```
@echo off  
echo Hello World  
echo.  
echo Premi un tasto per uscire  
pause>nul
```



3. Salviamo il file con il nome "1.bat" sul desktop
4. Avviamo lo script batch.



Se avete inserito il codice corretto, il risultato nel prompt sarà simile.

CAPITOLO 7

Istruzione Type e More

Per rendere più chiaro il codice, soprattutto quando devono essere stampati a video interi testi (ad esempio il Proemio dell' Iliade), possiamo risparmiarci di usare gli echo, ma di creare un file di testo contenente ciò che vogliamo visualizzare a video. Passiamo subito all'esempio:

CODICE:

```
@echo off
type proemio.txt
echo.
echo.
pause
```

Per prima cosa, dobbiamo salvare il file proemio.txt nella cartella in cui metteremo il file batch. Ricordiamo di mettere almeno un' echo., altrimenti l'output successivo verrà attaccato al testo.

Ah, evitiamo gli accenti, usiamo al posto di essi l'apostrofo. Ad esempio, "è" diventerà "e".

Usare |more

Se vogliamo che il testo venga letto un po' alla volta e schermata per schermata (soprattutto se è molto lungo), modifichiamo il codice come segue:

CODICE:

```
@echo off
type proemio.txt | more
echo.
echo.
pause
```

Dobbiamo ricordare queste cose, perchè potranno servirci nei prossimi esempi.

CAPITOLO 8

Leggere dalla riga di comando

Questa utile funzionalità degli script batch è quella di poter leggere argomenti dalla riga di comando.

Ad esempio se volessi richiamare il primo argomento scrivo "%1", se voglio leggere il secondo argomento, inserirò "%2" e così via.

ESEMPIO:

```
@echo off
echo Hai immesso '%1' dalla riga di comando
echo Premi un tasto per uscire.
pause>nul
```

Salviamo in C:\WINDOWS il file leggi.bat

Adesso inseriamo nel prompt:

```
leggi "IO SONO PEPPE"
```

Apparirà a video quello che è stato immesso, presentato in una forma simpatica.

CAPITOLO 9

Usare gli If per decidere a seconda dei parametri di comando

L'istruzione IF fa eseguire operazioni condizionali al programma batch. Ogni if restituisce un Errorlevel, che ha risultati numerici a seconda se l'operazione è andata a buon fine o no. Adesso, vale la pena dire, che l'uguaglianza tra due stringhe (quali gli argomenti) si esprimono con "stringa1==stringa2". Passiamo subito al codice...

CODICE:

```
@echo off
echo SONO IL TUO ROBOT
echo.
IF %1==ciao echo Salve Padrone
IF %1==grazie echo Prego
echo.
echo PREMA UN TASTO PER USCIRE
pause>nul
```

Salviamo il file in C:\Windows col nome "robot1.bat"

Adesso nel prompt inseriamo "robot1 ciao" e vedremo che il nostro robot ci risponderà, se invece inseriamo "robot1 grazie", ci risponderà dicendoci "prego".

CAPITOLO 10

Muoversi nel programma

Partiamo da un programma fatto così:

CODICE:

```
@echo off
IF %1==DO
IF %1==RE
IF %1==MI
IF %1==FA
IF %1==SOL
IF %1==LA
IF %1==SI
IF %1==esci
pause
```

Non è possibile scrivere più di un comando in un if, ma dobbiamo trovare un modo per poterci muovere nel programma, andare cioè in piccoli pezzi, uscendo dalla sequenzialità delle operazioni.

Ciò che ci serve è un SEGNAPOSTO, a cui possiamo assegnare un nome. Un segnaposto si crea in questo modo:

```
:nomesegnaposto
```

dove "nomesegnaposto" può essere uno qualsiasi. Si possono eseguire le istruzioni di un segnaposto tramite GOTO che ha questa sintassi:

```
GOTO nomesegnaposto
```

Adesso modifichiamo l'esempio

CODICE:

```
@echo off
:main
IF %1==DO GOTO do
IF %1==RE GOTO re
IF %1==MI GOTO mi
IF %1==FA GOTO fa
IF %1==SOL GOTO sol
IF %1==LA GOTO la
IF %1==SI GOTO si
```

```
IF %1==esci GOTO close
:do
echo Hai scelto DO
goto close
:re
echo Hai scelto RE
goto close
:mi
echo Hai scelto MI
goto close
:fa
echo Hai scelto FA
goto close
:sol
echo Hai scelto SOL
goto close
:la
echo Hai scelto LA
goto close
:si
echo Hai scelto SI
goto close
:close
pause
```

Il codice è un po' lunghetto, ma ora potrebbe servirci un codice per chiudere il prompt. Il comando che serve al nostro caso è:
EXIT

Adesso nel codice di prima, cambiamo i comandi del segnaposto "close" in:

```
:close
pause
exit
```

Fate una prova...

CAPITOLO 11

Le variabili di ambiente e i commenti

Scrivete "set" nel prompt...

Troverete una bella lista con info sul vostro sistema, sono le variabili di ambiente (variano da computer a computer), queste ci servono per capire qual'è ad esempio l'unità principale (in molti casi C) eccetera... Potete "scartabellare" tra le variabili d'ambiente. Se vogliamo aggiungere una nostra variabile d'ambiente, digitiamo:

```
set nomevariabile=valorevariabile
```

Ad esempio:

```
set peppe=padrone
```

Creeremo la variabile d'ambiente peppe.

Possiamo anche rappresentare la stringa di una variabile (ad esempio se la vogliamo utilizzare per l'istruzione "echo". Vi propongo un programmino che crei la variabile peppe e la stampi a schermo.

CODICE:

```
@echo off  
set peppe=padrone  
echo %peppe%
```

Dove il simbolo di percentuale, rappresenta la stringa della variabile. Quindi %peppe% rappresenta il valore della variabile.

Adesso passiamo alla creazione di commenti... usiamo l'istruzione REM con questa semplice sintassi:

```
REM commento
```

CAPITOLO 12

Istruzione Start

Adesso entriamo in un bivio: Windows XP oppure no? Ci sono istruzioni che ai possono fare solo in Windows XP... Parliamo per adesso di Windows XP:

Lo scopo di Start è quello di avviare un'applicazione. La sintassi è:

`START nomeprogramma`

Possiamo aggiungere `/MIN` dopo `nomeprogramma` per avviarlo ridotto a icona oppure `/MAX` per ingrandirlo, `/WAIT` per attendere sino alla chiusura, non sono comunque obbligatori. Possiamo passare anche argomenti al programma.

Adesso iniziamo con l'esempio:

```
@echo off
echo Stai inviando '%1' nel file '%2'
echo Premi un tasto per confermare
pause>nul
cls
echo Invio output al blocco note...
echo %1>%2
start notepad %2
echo Invio riuscito, premere un tasto per uscire.
pause>nul
exit
```

Adesso chiamiamo da riga di comando `"createtxt "SI LA SOL" prova$1.txt"` e apriremo notepad con la scritta `"SI LA SOL"`.

CAPITOLO 13

Lo script Autorun

Adesso vi farò vedere quanto è facile trasformare un semplice script in un (quasi)programma, il tutto è molto semplice... Fino ad ora abbiamo scritto nei files in una riga sola, ma adesso impariamo a scrivere in più righe. Prima usavamo questa sintassi:

SINTASSI:

```
echo CIAO>ciauz.txt
```

Ma se volessimo scrivere sulla seconda riga ? Anzichè usare > useremo >> . Dopo aver puntualizzato questo, ecco la sintassi per creare autorun:

SINTASSI:

```
@echo off
```

```
goto main
```

```
IF %1==help echo Sintassi: AUTORUN nomeprogramma  
echo nomeprogramma = programma da avviare  
echo.  
echo Premere un tasto per uscire  
pause>nul  
goto end
```

```
:main  
echo [autorun] >autorun.inf  
echo open=%1 >> autorun.inf  
cls  
echo File AUTORUN.INF creato con successo.  
echo Premere un tasto per uscire  
pause>nul
```

```
:end  
pause>nul  
exit  
:end
```

Concludiamo la prima parte sulla panoramica, adesso addentriamoci nella parte un tantino più teorica.

CAPITOLO 14

Ricapitolando Parte I

Ricapitoliamo adesso mettendo le informazioni brevi sui comandi appena imparati:

ECHO e TYPE:

Il primo serve per stampare a schermo una riga di testo, ma può essere usato per mostrare/nascondere l'invio dei comandi al prompt. Si usa "@echo off" per nascondere e "@echo on" per mostrare.

Il secondo serve per stampare sullo schermo il contenuto di un file tramite la sintassi "type nomefile".

PAUSE:

Consente di mettere in pausa l'esecuzione in un certo punto del programma, può essere nascosto tramite la sintassi "pause>nul".

GOTO:

Serve per andare ad un certo segnaposto (etichetta) del programma identificato da un nome preceduto dal simbolo ":". La sintassi per andare all'etichetta desiderata è "echo etichetta", per creare l'etichetta si usa ":etichetta".

START:

Serve per avviare un programma ed ha la semplice sintassi di "Start nomefile". Può avere anche parametri.

MORE:

Visualizza lo script schermata per schermata.

Per chi è ancora alle prime armi e ha ancora dubbi, è invitato a rileggere dal capitolo 4° al capitolo 13°.

Chi non ha alcun dubbio può proseguire con la lettura.

CAPITOLO 15

Istruzione Color

Questa istruzione è disponibile solo su Windows XP.

Per dare un colore allo sfondo e al testo del prompt, si usa questa istruzione. L'istruzione ha la sintassi "COLOR n1n2" dove n1 è il carattere che identifica il colore di sfondo e n2 che identifica il colore del testo.

Ecco i caratteri che corrispondono ai colori:

0 = Nero	8 = Grigio
1 = Blu scuro	9 = Blu
2 = Verde	A = Verde limone
3 = Verde acqua	B = Azzurro
4 = Bordeaux	C = Rosso
5 = Viola	D = Fucsia
6 = Verde oliva	E = Giallo
7 = Grigio chiaro	F = Bianco

Se inserisco "Color 14" Avrò testo Bordeaux su sfondo Blu scuro perchè 1 corrisponde al blu e 4 al bordeaux. Passiamo ora ad un semplice esempio:

CODICE:

```
@echo off  
COLOR 0%1
```

Adesso salviamo con il nome "coloratesto" e lascio a voi immaginare il risultato che avrete nel prompt...

CAPITOLO 16

Strutturiamo Batch

Dal fatto che Batch non è un vero linguaggio di programmazione, non esistono nè oggetti, nè dati, nè strutture. Però, tramite l'uso dei segnaposto, possiamo strutturare gli script, soprattutto se sono lunghi. D'ora in poi tutti gli esempi saranno strutturati. La strutturazione (da me inventata) è molto semplice ed è composta da tre parti principali:

begin
end
main

Ecco l'esempio d'uso della strutturazione:

CODICE:

```
:begin
REM Inizio dello script
@echo off
Color FC

:main
REM Metodo Principale
echo Hai inserito %1 e %2

:end
REM Fine dello script
echo.
echo Premi un tasto per uscire
pause>nul
```

Se la struttura è ben fatta, possiamo risparmiare tempo alla digitazione in quanto, ad esempio, per uscire dallo script, scriviamo "goto end".

CAPITOLO 17

Internet e Batch Parte I

In questo e nel successivo capitolo, affronteremo la tematica di "Internet + Batch". Questa tematica è molto affascinante e ricca di sorprese. In questa parte ci dedicheremo al PING.

Cos'è il Ping ?

Il PING è l'acronimo di "Packet InterNet Groper", esso consente di inviare pacchetti a un host che può essere anche un sito web. E' molto spesso uno strumento di conoscenze molto costruttivo, ma può trasformarsi in un vero e proprio attacco DoS. Attenzione a non confondere DoS con DOS, perchè sono due cose distinte, infatti DoS significa "Denial Of Service" (Negazione del servizio). Chi vorrebbe approfondire DoS, può leggere questa voce di Wikipedia:

<http://it.wikipedia.org/wiki/DoS>

Istruzione ping

L'istruzione Ping di Windows consente di inviare 4 pacchetti ad un sito web. Ad esempio: ping www.pierotofy.it , vi darà la seguente risposta:

Esecuzione di Ping pierotofy.it [72.232.69.250] con 32 byte di dati:

Risposta da 72.232.69.250: byte=32 durata=288ms TTL=50

Risposta da 72.232.69.250: byte=32 durata=281ms TTL=51

Risposta da 72.232.69.250: byte=32 durata=281ms TTL=51

Risposta da 72.232.69.250: byte=32 durata=296ms TTL=50

Statistiche Ping per 72.232.69.250:

Pacchetti: Trasmessi = 4, Ricevuti = 4, Persi = 0 (0% persi),

Tempo approssimativo percorsi andata/ritorno in millisecondi:

Minimo = 281ms, Massimo = 296ms, Medio = 286ms

Ecco quindi come usare ping.

CAPITOLO 18

Internet e Batch Parte II

Nella parte precedente abbiamo parlato di ping, questa volta parliamo di ipconfig e pathping.

Usare ipconfig

Questo speciale programma, ipconfig, consente di trovare l'indirizzo IP del nostro computer. Per sapere l'indirizzo IP, digitiamo "ipconfig" o "ipconfig /all"; se digitiamo il secondo otterremo le stesse informazioni del primo, ma più dettagliate.

Usare pathping

Questo è uno strumento che include le caratteristiche di ping e tracert (che consente di visualizzare i router che portano all'indirizzo web). Si usa "pathping sitoweb", ad esempio "pathping www.pierotofy.it". Vengono eseguite le analisi sui router, viene eseguito il ping e infine appaiono le statistiche in 4-6 minuti circa (250-400) di attività.

CAPITOLO 19

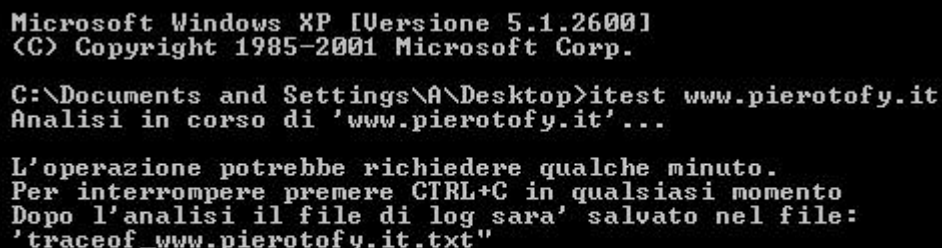
Usare le conoscenze su Internet e Batch

Le conoscenze affrontate nei capitoli precedenti, possono essere applicate per creare un progetto che dovrà eseguire analisi di un indirizzo web. Ecco un esempio:

CODICE:

```
@echo off
echo Analisi in corso di '%1'...
echo.
echo L'operazione potrebbe richiedere qualche minuto.
echo Per interrompere premere CTRL+C in qualsiasi momento
echo Dopo l'analisi il file di log sarà salvato nel file:
echo 'traceof_%1.txt'
pathping %1>nul > traceof_%1.txt
echo Analisi terminata.
start notepad traceof_%1.txt
:end
```

Proviamo a creare il file batch e a chiamarlo dalla riga di comando (io l'ho chiamato itest). Ecco come apparirà:



```
Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\A\Desktop>itest www.pierotofy.it
Analisi in corso di 'www.pierotofy.it'...

L'operazione potrebbe richiedere qualche minuto.
Per interrompere premere CTRL+C in qualsiasi momento
Dopo l'analisi il file di log sarà salvato nel file:
'traceof_www.pierotofy.it.txt'
```

CAPITOLO 20

Istruzione Call

Può esserci la necessità di chiamare un file batch da un altro file batch senza l'interruzione di quest'ultimo. Il comando che ci consente di fare tutto questo è CALL.

SINTASSI:

```
call filebatch [parametri] [:segnaposto]
```

In pratica al posto di filebatch, inseriremo il nome dello script che vogliamo richiamare. Possiamo inviare anche parametri e/o chiamare un segnaposto (etichetta).

ESEMPIO:

```
@echo off  
REM Chiamare un file batch  
REM Al segnaposto "ifcalled"  
call %1 :ifcalled  
echo Hai richiamato %1
```

In questo modo sarà possibile eseguire un'altro script e non interrompere quest'ultimo.

CAPITOLO 21

Istruzione Choice

Se siamo su Windows XP, ci serve l'eseguibile CHOICE.EXE che è possibile estrarre scaricando il **Disco Supplementare MS-DOS 6.22** dal link <http://download.microsoft.com/download/win31/update/1/dos/en-us/sup622.exe>. L'eseguibile dovrà essere copiato nella cartella Windows. Il problema non si verifica su Windows 98.

Choice, ci permette di effettuare scelte, nel senso che l'utente può scegliere una tra più opzioni. Ecco la sintassi:

SINTASSI:

choice /c:scelte

Dove scelte sono le lettere delle opzioni.

ESEMPIO:

choice /c:ab

In questo caso il prompt ci mostrerà:

[A,B]?

Dove viene richiesta una scelta.

Possiamo anche immettere una descrizione come in questo esempio:

ESEMPIO:

choice Scegliere sì o no /c:sn

Il prompt ci mostrerà:

Scegliere sì o no [S,N]?

Possiamo anche nascondere le scelte e il punto interrogativo.

ESEMPIO:

choice Scegliere A oppure B /c:ab /n

In una prossima lezione scopriremo come eseguire operazioni a seconda delle scelte.

CAPITOLO 22

Istruzione If Parte I

L'istruzione IF, è la famosa istruzione che ci consente di operare in modo condizionale. IF ha svariati usi e serve principalmente per: elaborare l'errorlevel (ossia il codice di uscita che da un comando), confrontare due stringhe, verificare l'esistenza di un file.

Confronto di due stringhe:

E' questo il modo d'uso più semplice, anche perchè bisogna usare una sintassi non molto complessa:

SINTASSI:

```
if stringa1 == stringa2
```

Le stringhe possono essere i parametri della riga di comando, delle parole, delle variabili d'ambiente.

ESEMPIO:

```
@echo off
if "%1"=="A" echo ESATTO
if "%1"=="" echo NON HAI SCRITTO NULLA
```

CAPITOLO 23

Istruzione If Parte II

Elaborare a seconda dell'ERRORLEVEL:

Adesso è arrivato il momento di scoprire cos'è questo già citato "ERRORLEVEL"; ogni istruzione, da un valore d'uscita che rappresenta l'esito dell'operazione. Impareremo ora ad eseguire operazioni a partire dalla scelta effettuata col comando choice. Questo comando, da un errorlevel orientato a seconda del posto della scelta. Ad esempio se ho "choice /c:ab", "ERRORLEVEL 1" sarà la scelta A, mentre "ERRORLEVEL 2" sarà la scelta B. Ecco la sintassi:

SINTASSI:

if ERRORLEVEL numero ISTRUZIONI

Numero rappresenta il valore di errorlevel

ESEMPIO:

```
@echo off
choice Effettua una scelta /c:ab
if ERRORLEVEL 1 goto err1
if ERRORLEVEL 2 goto err2
```

```
:err1
echo Hai scelto il Proemio dell'Iliade
goto end
```

```
:err2
echo Hai scelto il proemio dell'Odissea
```

```
:end
echo Premi un tasto per uscire
pause>nul
```

CAPITOLO 24

Istruzione If Parte III

Confronto di due stringhe:

E' possibile, tramite l'uso di IF, confrontare due stringhe in questo modo:

SINTASSI:

if stringa1 operatore stringa2 ISTRUZIONI

L'operatore è una "sigla" che permette di confrontare due stringhe. Ecco gli operatori

EQU = uguale a

NEQ = diverso da

LSS = minore di

LEQ = minore o uguale a

GTR = maggiore di

GEQ = maggiore o uguale a

ESEMPIO:

@echo off

if "%1" GEQ "3" echo %1 è maggiore o uguale a 3

if "%1" EQU "prova" echo Hai scritto prova ed hai una penitenza

if "%1" LSS "200" echo %1 è minore di duecento

echo premi un tasto per uscire

pause>nul

E' possibile anche imporre al comando di non fare differenze tra maiuscole e minuscole. Per far questo basta anteporre "/I" prima della stringa 1.

CAPITOLO 25

Istruzione If Parte IV

Verificare l'esistenza di un file

Con If è possibile stabilire se un file esista o meno. A questo scopo si usa:

SINTASSI:

if EXIST nomefile ISTRUZIONI

Ecco un semplice programmino per la ricerca dei files:

ESEMPIO:

```
@echo off
if EXIST %1 echo Il file %1 esiste
pause
```

Tutto qui, EXIST è ciò che ci permette di stabilire se un file esiste.

CAPITOLO 26

Istruzione If Parte V

La clausola ELSE:

Partiamo da questo codice:

ESEMPIO:

```
@echo off  
If "%1"=="prova" echo Scelta giusta
```

Come si fa ad eseguire istruzioni se non inseriamo "prova"? Non possiamo di certo prevedere tutto ciò che viene immesso! Allora entra in gioco la clausola else, che ci permette di eseguire istruzioni se scegliamo qualcos'altro. Ecco l'esempio con questa clausola:

ESEMPIO:

```
@echo off  
If "%1"=="prova" (echo Scelta giusta) else echo Scelta sbagliata
```

In questo modo si riesce ad eseguire istruzioni se non si ottiene l'input desiderato.

L'operatore NOT:

E' possibile eseguire istruzioni anche se non si verifica la condizione. In questo modo, antepriamo prima della condizione l'operatore logico NOT.

ESEMPIO:

```
@echo off  
if EXIST %1 echo %1 esiste  
if NOT EXIST %1 echo %1 non esiste
```

Tutto qui...

CAPITOLO 27

Istruzione For

Questa istruzione, ci permette di eseguire operazioni su un gruppo di file.

SINTASSI:

for %%variabile in (files) do istruzione

Come potrete intuire, si può eseguire una istruzione su dei files, rappresentati dalla %%variabile.

ESEMPIO:

```
@echo off
for %%f in (%1) do goto dsfiles
:dsfiles
echo File %%f :
echo.
type %%f
echo.
```

In questo modo creeremo un programma che mostra il contenuto dei files. %%f è la variabile, %1 rappresenta i files su cui lavorare e goto dsfiles indica l'azione da eseguire per ogni file.

CAPITOLO 28

Appendice Utilizzo dei filtri e degli operatori

Sono disponibili tre filtri, che consentono di fare una specie di "crivello" per trovare e ordinare dei dati che si trovano in un file di testo. Per scrivere in un file di testo abbiamo usato l'operatore > che consente di scrivere in un file cancellando eventualmente il testo già contenuto. Possiamo invece usare l'operatore >> che accoda alla fine del testo già presente altro testo.

ESEMPIO:

```
@echo off
echo Prof. Rossi > prof.txt
REM Scrive la stringa "Prof. Rossi" nel file di testo "prof.txt"
echo Prof. Gialli > prof.txt
REM Scrive la stringa "Prof. Gialli" nel file di testo "prof.txt"
REM cancellando prima il suo contenuto.
echo Prof. Bianchi >> prof.txt
REM Scrive la stringa "Prof. Bianchi" nel file di testo "prof.txt"
REM accodando al contenuto preesistente.
```

E' possibile anche visualizzare una schermata alla volta, come descritto nella Lezione 7 (Istruzioni Type e More).

Può essere effettuata anche una ricerca all'interno del file, in questo modo:

ESEMPIO:

```
find "Prof. Gialli" < prof.txt
```

Si può anche ordinare alfabeticamente il testo in questo modo:

ESEMPIO:

```
type prof.txt | sort
```

C'è anche un modo per ordinare e cercare in un file di testo:

ESEMPIO:

```
find "Prof. Gialli" < prof.txt | sort
```

Avendo un po' di fantasia, possiamo creare un generatore di query:

ESEMPIO:

```
find "Prof. Gialli" < prof.txt | sort > query.txt
```

In poche parole possiamo fare un semplice script per creare query, con due sole righe!

ESEMPIO:

```
@echo off
find %1 < %2.txt | sort > %3.txt
```

CAPITOLO 29

Risorse e strumenti utili

Da Batch ad Eseguiibile:

Dopo una lunga ricerca, sono riuscito a trovare un freeware per convertire da BAT ad EXE. Potete scaricarlo da questo sito web:

<http://snipurl.com/16n0h>

Informazioni sulla riga di comando:

Questa è un riepilogo delle funzioni della Shell, direttamente dal "Microsoft TechNet". Peccato che è più specifica per "Windows Server 2003", ma è anche utile per gli utenti di Windows XP.

<http://snipurl.com/16n0f>

Disco supplementare MS-DOS 6.22:

E' lo stesso download citato nella pagina in cui si spiegava l'istruzione "Choice". E' utile, se non obbligatorio a chi usa Windows XP e vuole avere le funzioni aggiuntive MS-DOS.

<http://snipurl.com/16n0i>

Notepad++:

Ottimo e Open Source programma per la scrittura di codice, può essere usato come editor per file batch.

<http://snipurl.com/16n0j>