

Introduzione

Un **file batch** è un file di testo di tipo ASCII che contiene una sequenza di comandi DOS.

I file batch devono avere estensione **.BAT**

I comandi DOS all'interno dei programmi batch sono eseguiti sequenzialmente, a partire dal primo, come se fossero immessi dopo il prompt.

Esempio 1

Si desidera realizzare un file batch che sposti tutti i file con estensione **.TXT** dalla directory **PARTENZA** alla directory **ARRIVO** dell'unità **A:**

Soluzione: SPOSTA_1.BAT

```
COPY A:\PARTENZA\*.TXT A:\ARRIVO  
DEL A:\PARTENZA\*.TXT.
```

Comandi Specifici per i File Batch

- **REM** permette di inserire in un file batch commenti che descrivano lo scopo delle istruzioni
- **PAUSE** sospende temporaneamente (fino alla digitazione di un tasto) l'esecuzione del file batch. Visualizza il messaggio:
Premere un tasto per continuare...
- **ECHO** abilita o disabilita la visualizzazione, all'esecuzione, delle righe del file batch; inoltre consente di visualizzare messaggi sullo schermo

Esempio 2

Come l'Esempio 1, ma con interfaccia-utente migliorata.

Soluzione: SPOSTA_2.BAT

```
@ECHO OFF
CLS
REM Comando per spostare tutti i file .TXT
REM dalla dir PARTENZA alla dir ARRIVO
ECHO Comando per spostare tutti i file .TXT
ECHO dalla dir PARTENZA alla dir ARRIVO
ECHO;
ECHO Per annullare premere Ctrl-C
PAUSE
COPY A:\PARTENZA\*.TXT A:\ARRIVO
DEL A:\PARTENZA\*.TXT
ECHO Spostamento completato
ECHO ON
```

Alcune motivazioni per l'uso dei file batch

a) Comandi ripetitivi. Esempio: AUTOEXEC.BAT

```
SET BLASTER=A220 I5 D1 T4
SET SOUND16=C:\SOUND16
C:\SOUND16\sndinit /b
rem ----- MTM ATAPI CD-ROM -----
C:\DOS\MSCDEX.EXE /D:MTMIDE01 /M:10
rem ----- MTM ATAPI CD-ROM -----
C:\DOS\SMARTDRV.EXE /X
@ECHO OFF
PROMPT $p$g
PATH C:\WINDOWS;C:\DOS
SET TEMP=C:\DOS
MODE CON CODEPAGE PREPARE=(850) C:\DOS\EGA.CPI)
MODE CON CODEPAGE SELECT=850
KEYB IT,,C:\DOS\KEYBOARD.SYS
PATH %PATH%;C:\SOUND16
```

b) Comandi lunghi e noiosi. Esempio: PULISCI.BAT
(elimina i file ausiliari creati in precedenza da alcuni applicativi che non interessano più).

```
@ECHO OFF
ECHO                (distrugge i file *.log, *.aux, *.dvi, *.ps)
DEL *.log
DEL *.aux
del *.dvi
del *.ps
EXIT
```

c) Come promemoria. Esempio: USOLATEX.BAT

(LATEX è un sistema di scrittura che permette di realizzare testi formattati come quelli di un libro, con formule, tabelle, ecc. Si parte da un file di tipo testo scritto con una specifica sintassi e successivamente per visualizzare o stampare il testo occorre processare il file con una serie di programmi parametrizzati. Per evitare di dover ricordare la sequenza e la sintassi dei comandi, si è realizzato il seguente file batch).

```
ECHO OFF
CLS
ECHO          "Modalita' d'uso di LATEX"
ECHO LATEX nomefile.tex          (per compilare)
ECHO PREVIEW nomefile          (per visualizzare)
ECHO DVIPS nomefile          (per produrre file per
POSTSCRIPT)
ECHO COPY /B nomefile.ps PRN:    (per stampare su LP POSTSCRIPT)
ECHO PULISCI.BAT          (distrugge i file *.log, *.aux, *.dvi,
*.ps)
EXIT
```

d) Comandi complessi, difficili da ricordare. Esempio: VIRUS.BAT

(i programmi utilizzati per controllare che i file o i dischi non siano infettati da virus cambiano nome e modalità d'uso a seconda del costruttore. Inoltre possono richiedere switches di volta in volta differenti. Si realizza un file batch che "lancia" l'antivirus sempre con la stessa modalità di chiamata e con un nome "mnemonico" per l'utente).

```
@echo off
CLS
if "%1" == "" goto messaggio
echo.
echo "Verifica del drive %1"
echo.
c:\devirus\scan %1          (scan.exe è l'antivirus attualmente in
uso)
goto end
:messaggio
ECHO.
rem
echo "UTILIZZO"
ECHO VIRUS d:      (ES.:  VIRUS C:)
ECHO.
:END
```

e) Per usare un applicativo sempre con lo stesso nome. Esempio: XTREE.BAT
(quando esisteva solo il DOS, per visualizzare i direttori e in generale per gestire con efficienza il File System era stato creato da un'azienda un programma, XTREE.EXE, che aveva avuto una larghissima diffusione tra gli utenti di PC. Il successo aveva spinto i costruttori a potenziare ulteriormente il programma, realizzando nuove versioni: il programma aveva cambiato nome in XTGOLD.EXE, ma gli utenti erano affezionati al vecchio).

```
@ECHO OFF  
c:\xtgold\xtgold.exe
```

I Parametri Sostituibili nei File Batch

- Un parametro è una sequenza di caratteri e/o cifre introdotte dopo l'istruzione nella riga di comando.
- Un parametro sostituibile è un simbolo costituito da % seguito da un numero (da %1 a %9, %0 è il nome del file batch) che può essere incluso in un file batch e che prende il posto del rispettivo parametro passato nella riga di comando.

In tal modo è possibile specificare il valore reale delle variabili al momento dell'esecuzione del file.

Esempio 3

Come l'Esempio 1, ma l'estensione dei file da spostare, così come i path delle directory sorgente e destinazione devono poter essere specificati all'atto della chiamata del file batch.

Soluzione: SPOSTA_3.BAT

```
COPY %2\*.*%1 %3  
DEL %2\*.*1
```

La riga di comando sarà, per esempio:

```
SPOSTA_3 TXT A:\PARTENZA A:\ARRIVO
```


Comandi Avanzati per i File Batch (1)

- **IF** consente l'esecuzione condizionale di un comando. Il comando IF verifica le seguenti tre condizioni:
 - Se il codice di uscita di un programma è maggiore o uguale ad un valore. (Solitamente ERRORLEVEL = 0 corrisponde al caso di terminazione corretta).
 - Se una stringa è uguale ad un'altra.
 - Se esiste un file.
- **GOTO** salta alla riga seguente un'etichetta specificata.

Esempio 4

Come l'Esempio 3, ma si vuole controllare il corretto svolgimento delle operazioni. In particolare si deve verificare:

- Che la riga di comando contenga i parametri necessari.
- Che nella directory sorgente esistano dei file con l'estensione specificata.
- Che l'operazione di copia avvenga senza errori o interruzioni

e solo in questo caso procedere alla cancellazione dei file nella directory sorgente.

Soluzione: SPOSTA_4.BAT

```
IF "%1"==" " GOTO ERR_EST
IF "%2"==" " GOTO ERR_EST
IF "%3"==" " GOTO ERR_EST
IF NOT EXIST %2\*.*%1 GOTO ERR_FILE
md %3
xcopy %2\*.*%1 %3
IF ERRORLEVEL 1 GOTO ERR_INT
del %2\*.*%1
ECHO Spostamento completato
GOTO FINE
:ERR_EST
ECHO Sintassi:%0 Estens Dir_Sorg Dir_Dest
GOTO FINE
:ERR_FILE
ECHO File non trovato
GOTO FINE
:ERR_INT
ECHO Operazione interrotta
:FINE
```

Comandi Avanzati per i File Batch (2)

- **CHOICE** serve per sospendere temporaneamente l'esecuzione di un file batch e consentire all'utente di operare una scelta.

La sintassi del comando è

```
CHOICE /C:scelte messaggio
```

/C:scelte elenca i tasti che possono essere premuti. Se omissso, per default essi sono [S,N].

messaggio è il testo opzionale che può essere visualizzato.

In base alla scelta fatta dall'utente viene assegnato un valore ad **ERRORLEVEL** (al primo tasto viene assegnato il valore 1, al secondo 2 ecc.). In pratica è possibile analizzare il tasto premuto tramite una serie di comandi **IF ERRORLEVEL**

La condizione *IF ERRORLEVEL num* è verificata qualora la variabile **ERRORLEVEL** sia maggiore od uguale a num.

Esempio 4bis

Come l'Esempio 4, ma con possibilità di scegliere se cancellare o meno i file dalla directory sorgente

Soluzione: SPOSTA4b.BAT

```
...  
CHOICE Cancellare i file  
IF NOT ERRORLEVEL 2 del %2\*.*1
```

```
...  
Il messaggio visualizzato durante l'esecuzione sarà  
Cancellare i file [S,N]?.
```

Comandi Avanzati per i File Batch (3)

- **FOR** permette di eseguire più volte un comando DOS per un determinato insieme di parametri.

La sintassi del comando è:

```
FOR %%variabile IN (a,b,...) DO
```

La variabile simbolica è impostata in sequenza per ciascuno degli elementi del gruppo.

Esempio 5

Si desidera realizzare un file batch che sposti tutti i file specificati sulla linea di comando dalla directory PARTENZA alla directory ARRIVO dell'unità A. Si assume che i file siano meno di dieci.

Soluzione: SPOSTA_5.BAT

```
FOR %%x IN (%1 %2 %3 %4 %5 %6 %7 %8 %9) DO  
copy a:\partenza\%%x a:\arrivo  
FOR %%x IN (%1 %2 %3 %4 %5 %6 %7 %8 %9) DO  
del a:\partenza\%%x
```

Comandi Avanzati per i File Batch (4)

• **SHIFT** il comando shift sposta i parametri nella linea di comando che ha avviato il file batch. Ciascun parametro si muove di una posizione a sinistra.

%0←%1: il valore contenuto in %0 viene sostituito da quello precedentemente contenuto in %1

%1←%2

...

%8←%9

%9← parametro successivo: %9 riceve come valore il 10° parametro, prima non accessibile

Il comando SHIFT viene usato per superare il limite imposto dal DOS di nove parametri sostituibili.

Esempio 6

Come l'Esempio 5, ma si assume che i file possano essere più di dieci.

Soluzione: SPOSTA_6.BAT

```
IF NOT "%1"==" " GOTO SPOSTA
ECHO Sintassi: %0 File_1 File_2 ...
GOTO FINE
:SPOSTA
copy a:\partenza\%1 a:\arrivo
del a:\partenza\%1
SHIFT
IF NOT "%1"==" " GOTO SPOSTA
ECHO Spostamento completato
:FINE.
```

Le Variabili nei File Batch

• Per assegnare un valore ad una variabile si utilizza l'istruzione **SET**. Per leggere il contenuto di una variabile si scrive il suo nome tra due caratteri %

- **SET tmp = val**

assegna alla variabile *tmp* il valore *val*

- **SET tmp = %var%**

assegna alla variabile *tmp* il contenuto della variabile *var*

Esempio 7

Come l'Esempio 6, ma i path delle directory sorgente e destinazione devono poter essere specificati all'atto della chiamata del file batch.

Le due directory nell'ordine saranno specificate come primi due parametri e i file dal terzo in avanti.

Soluzione: SPOSTA_7.BAT

```
IF NOT "%1"==" " GOTO INIZ
ECHO Sintassi: %0 Dir_Partenza Dir_Arrivo File_1 File_2 ...
GOTO FINE
:INIZ
SET Dir_Partenza=%1
SHIFT
SET Dir_Arrivo=%1
SHIFT
:SPOSTA
copy %Dir_Partenza%\%1 %Dir_Arrivo%
del %Dir_Partenza%\%1
SHIFT
IF NOT "%1"==" " GOTO SPOSTA
ECHO Spostamento completato
:FINE
```

Richiamare un File Batch all'interno di un altro

- Per **passare permanentemente il controllo** da un file batch ad un altro si deve includere il nome del secondo file batch all'interno di una riga del primo.
- Per **richiamare un file batch** all'interno di un altro, eseguirlo e quindi tornare al file originario per completarne l'esecuzione si usa il comando **CALL** .

Esempio 8

Come l'Esempio 7, ma si vuole passare come parametri non più un elenco di file, ma un elenco di estensioni, e spostare tutti i file con le estensioni specificate dalla directory sorgente alla directory destinazione.

SPOSTA_8.BAT

```
IF NOT "%1"==" " GOTO INIZ
ECHO Sintassi: %0 Dir_Partenza Dir_Arrivo Estens_1 Estens_2 ...
GOTO FINE
:INIZ
SET Dir_Partenza=%1
SHIFT
SET Dir_Arrivo=%1
SHIFT
:SPOSTA
CALL sposta4b.bat %1 %Dir_Partenza%
%Dir_Arrivo%
SHIFT
IF NOT "%1"==" " GOTO SPOSTA
ECHO Spostamento completato
GOTO FINE.
```

Esercizio 1: DIR2.BAT

Scrivere il testo di un file DIR2.BAT attivabile come:

```
DIR2.BAT primo secondo
```

dove *primo* e *secondo* sono i pathname di due directory specificate dall'utente.

Il comando DIR2 deve creare un file *lista.lis* nel direttorio *\tmp* dell'unità C contenente la lista di *primo* e *secondo*.

Il contenuto di *lista.lis* deve poi essere visualizzato su video e il file cancellato.

Soluzione: DIR2.BAT

Versione 1

```
@echo off
dir %1 > c:\tmp\lis1.lis
dir %2 > c:\tmp\lis2.lis
copy c:\tmp\lis1.lis +
c:\tmp\lis2.lis
c:\tmp\lista.lis
type c:\tmp\lista.lis
del c:\tmp\lis1.lis
del c:\tmp\lis2.lis
del c:\tmp\lista.lis
echo on. Soluzione: DIR2.BAT
```

Versione 2

```
@echo off
dir %1 > c:\tmp\lista.lis
dir %2 >> c:\tmp\lista.lis
type c:\tmp\lista.lis
del c:\tmp\lista.lis
echo on
```

Esercizio 2: LISTA.BAT

Scrivere il testo di un file LISTA.BAT contenente comandi MS-DOS ed attivabile come

```
LISTA file_x dir_1 dir_2 dir_3
```

Per ognuno dei direttori *dir_1 dir_2 dir_3* specificati sulla linea di comando la procedura deve:

- verificare l'esistenza del file *file_x* e visualizzare un apposito messaggio in caso negativo;
- visualizzare il contenuto del file sullo schermo una pagina alla volta
- copiare il file nel direttorio \save dell'unità C. Soluzione: LISTA.BAT

Soluzione: LISTA.BAT

```
REM lista.bat
@echo off
FOR %%i in (%2 %3 %4) DO CALL
lista1.bat %1 %%i
echo on
REM lista1.bat
@echo off
IF EXIST %2\%1 GOTO F_OK
echo Il file %2\%1 non esiste
GOTO FINE
:F_OK
echo Contenuto del file %2\%1:
type %2\%1 | more
copy %2\%1 c:\save
:FINE
echo Operazione terminata
```


Esercizio 3: SALVA.BAT

Scrivere il testo di un file batch SALVA.BAT attivabile come

SALVA percorso unità

in grado di effettuare le seguenti operazioni:

- richiedere l'inserimento di un dischetto nell'unità specificata ed attendere il consenso a proseguire tramite tastiera;
- formattare il dischetto e creare il direttorio *save* nella sua radice;
- copiare tutti i file presenti nel direttorio individuato da *percorso* nel direttorio *save* creato in precedenza;
- creare nel direttorio *save* un file *indice.txt* che contenga la lista dei file salvati.

Soluzione: SALVA.BAT

```
@echo off
echo Inserire un dischetto
nell'unità "%2"
pause
format %2
md %2\save
copy %1\*. * %2\save
dir %1 > %2\save\indice.txt
echo on
```

Esercizio 4: CERCA.BAT

Realizzare un file batch che permetta di ricercare all'interno di un file di testo tutte le linee nelle quali sia presente una certa sequenza di caratteri. Il path per raggiungere il file testo e la sequenza di caratteri devono essere specificati all'atto della chiamata del batch.

La lista generata e comprendente tutte le linee contenenti la sequenza data deve poter essere inviata a scelta

- su video
- su stampante
- su un altro file il cui nome deve essere specificato quando si chiama il batch.

Soluzione: CERCA.BAT

```
@echo off
echo Sintassi: %0 Stringa File Opzione File_Dest
IF %3 == V GOTO VID
IF %3 == F GOTO FIL
IF %3 == S GOTO PRN
GOTO FINE
:VID
find/n "%1" %2
GOTO FINE
:FIL
find /n "%1" %2 >%4
GOTO FINE
:PRN
find /n "%1" %2 >lpt1
:FINE
echo Operazioni terminate
echo on
```

Esercizio 5: ORDINA.BAT

Realizzare un file batch che permetta di ordinare alfabeticamente tutte le linee di un file testo specificato e di salvare il risultato così ottenuto su **n** nuovo file. I path da seguire per raggiungere i files sorgente e destinazione devono essere specificati all'atto della chiamata del batch.

Modificare il batch in modo che alla sua chiamata possa anche essere specificato il tipo di ordinamento (normale o inverso).

Soluzione: ORDINA.BAT

```
@echo off
echo Sintassi: %0 File_Sorg File_Dest Opzione
IF %3 == D GOTO INFI
IF %3 == I GOTO FIIN
GOTO FINE
:INFI
sort %1 >%2
GOTO FINE
:FIIN
sort /r %1 >%2
:FINE
echo Operazioni terminate
echo on
```

Esercizio 6: COPIA.BAT

Realizzare un file batch MS-DOS che esegua le operazioni di seguito indicate tra due direttori passati come parametri:

- copia tutti i file con estensione .AUX dal primo direttorio al secondo direttorio;
- crea il sottodirettorio DVI nel primo direttorio;
- copia tutti i file con estensione .DVI dal primo direttorio al sottodirettorio DVI appena creato.
- stampa (su PRN) del contenuto dei due direttori passati come parametri..

Soluzione: COPIA.BAT

```
@echo off
IF NOT "%1"==" " GOTO CP
echo Sintassi: %0 Dir_1 Dir_2
GOTO FINE
:CP
copy %1\*.AUX %2
md %1\DVI
copy %1\*.DVI %1\DVI
dir %1 > PRN:
dir %2 > PRN:
:FINE
echo Operazioni terminate
echo on
```