

I file batch sono molto utili a chi opera in ambiente CLI perché consentono di velocizzare e rendere più sicura l'esecuzione di alcuni compiti. Si possono verificare alcune situazioni tipiche, elencate qui di seguito; ma nella pratica comune insorgono moltissime situazioni in cui si evidenzia la necessità di accorpare un insieme di comandi.

Comandi ripetitivi

Molti applicativi (DEV+++, Proteus, ecc.) durante il loro funzionamento producono file ausiliari che servono solo all'applicativo stesso e non sono di alcun interesse per l'utente, che sarebbe ben contento di non vederli più. Si può creare un comando che li elimini tutti, per esempio Pulisci.BAT:

Esempio1:

```
@ECHO OFF
```

```
ECHO Questo batch elimina i file *.log, *.aux, *.dvi, *.ps dalla directory corrente
```

```
DEL *.log
```

```
DEL *.aux
```

```
DEL *.dvi
```

```
DEL *.ps
```

In questo modo l'operatore scrive solo **Pulisci**, un solo comando senza parametri invece di 4 con wildchar, e con la possibilità di sbagliarsi nella compilazione (e dover riscrivere il singolo comando)

Esercizio: creare il file batch **myBackup.bat** per automatizzare il backup dei file utili presenti nella cartella di lavoro WIP, salvandoli nella cartella BACKUP che si trova allo stesso livello di WIP; questo esercizio richiede di popolare la cartella WIP con alcuni file, con diverse estensioni

Comandi meglio documentati

I comandi DOS e di molti applicativi non sono particolarmente "auto documentanti"; se l'operatore non ricorda bene la sintassi deve ricorrere al manuale (cartaceo o help). Un modo per superare questa difficoltà è quello di creare un batch con un nome facile da ricordare che, nel caso di un errore di digitazione o di mancanza di un parametro, suggerisca la sintassi corretta (parametri&opzioni).

Esercizio: creare il batch **Nascondi.bat** che nasconde il file indicato come parametro; in assenza di parametro emette un help sulla sintassi del comando; consultare la guida DOS per conoscere scopo e sintassi di ATTRIB

Comandi complessi, difficili da ricordare, con tanti parametri/opzioni, usati come "alias"

Un tipico comando di scansione del disco alla ricerca di errori è:

```
c:\...\chkdsk C:
```

Consultare l'help inline del comando per scoprire a cosa serve e come si usa (la sintassi)

Si può inserire la riga precedente in un batch che controlla la presenza del parametro e – qualora mancasse - emetta un messaggio che ne richiama l'inserimento; inoltre tale comando può essere corredato di commenti interni (con REM) e messaggi esterni con info per l'utente.

```
@ECHO OFF
```

```
CLS
```

```
IF "%1" == "" GOTO messaggio
```

```
ECHO.
```

```
ECHO "Verifica del drive %1"
```

```
ECHO.
```

```
c:\...\chkdsk %1
```

```
GOTO end
```

```
:messaggio
```

```
ECHO.
```

```
ECHO UTILIZZO:
```

```
ECHO chkdsk <disco> (ES.: chkdsk C:)
```

```
ECHO.
```

```
:end
```

L'espressione IF "%1"== "" controlla l'esistenza del primo parametro

Esercizio: scopiazzando l'esempio precedente considerare un comando dalla sintassi complessa e produrre un batch come alias.

Per usare un applicativo sempre con lo stesso nome (cross version) o con un nome più semplice

Quando esisteva solo il DOS, per visualizzare le directory e in generale per gestire con efficienza il File System era stato creato da un'azienda il 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 nome; ecco una buona opportunità di creare **XTREE.BAT** e riempirlo con:

```
@ECHO OFF
c:\xtgold\XTGOLD.EXE
```

Creare comandi avanzati, con parametri particolari

Esercizio: Scrivere il testo di DIR2.BAT attivabile come:

```
DIR2.BAT param1 param2
```

dove param1 e param2 sono i path di due directory specificate dall'utente. Il comando DIR2 deve creare un file lista.lis nella directory \tmp dell'unità C: contenente l'elenco dei file in param1 e in param2. Il contenuto di lista.lis deve poi essere visualizzato su video e il file cancellato. Se non sono presenti entrambi i parametri il batch deve interrompere l'esecuzione del batch ed emettere opportuni messaggi di avviso all'utente.

Esercizio: Scrivere il testo del batch LISTA.BAT contenente comandi MS-DOS ed attivabile come LISTA filename dir1 dir2

Per ogni directory specificata sulla linea di comando la procedura deve:

- verificare l'esistenza del file file_x e visualizzare un apposito messaggio in caso negativo
- visualizzare sullo schermo il path e il nome del file, poi il contenuto del file una pagina alla volta
- copiare il file nella directory \save dell'unità C: (creare tale cartella se non esiste)

Esercizio: 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 (a scelta dell'utente). Il path per raggiungere il file di testo e la sequenza di caratteri devono essere specificati come parametri del batch. La lista generata, comprendente tutte le linee contenenti la sequenza, deve poter essere inviata a scelta (mediante opzioni) su: video / stampante / file ; nel caso di output su file il nome deve essere specificato quando si invoca il batch

Esercizio: Creare un file batch che consenta di ordinare alfabeticamente tutte le linee di un file di testo (indicato dal primo parametro) contenuto in una determinata cartella (indicata dal secondo parametro) nell'ordine (SU = A->Z opp. GIU = Z->A) specificato dal terzo parametro.

Esercizio: Creare il batch **SCAMBIA.BAT** che riceve come parametri 2 file e le cartelle dove risiedono, da scambiare nelle reciproche posizioni. In caso di errori nei parametri (parametro mancante o file inesistente) deve essere emesso un opportuno messaggio indicante la sintassi corretta o il possibile errore, e lo scambio non deve essere eseguito.