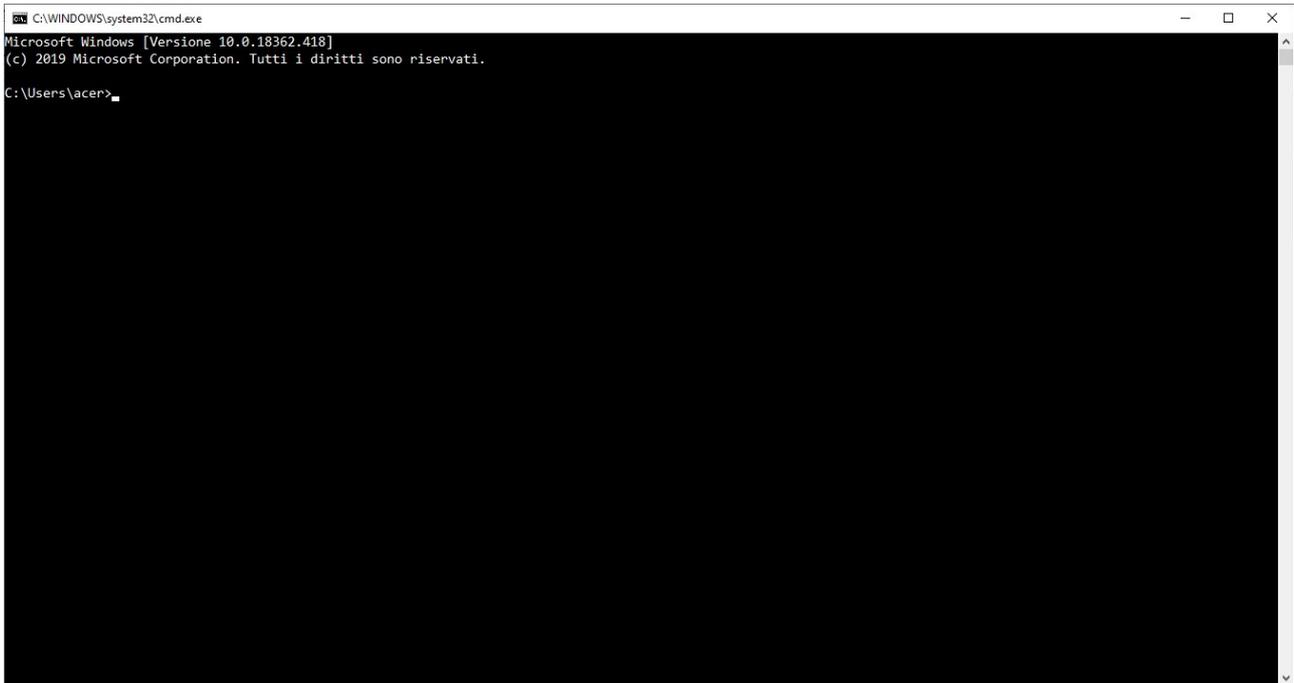


WINDOWS COMMAND LINE – INTRODUZIONE

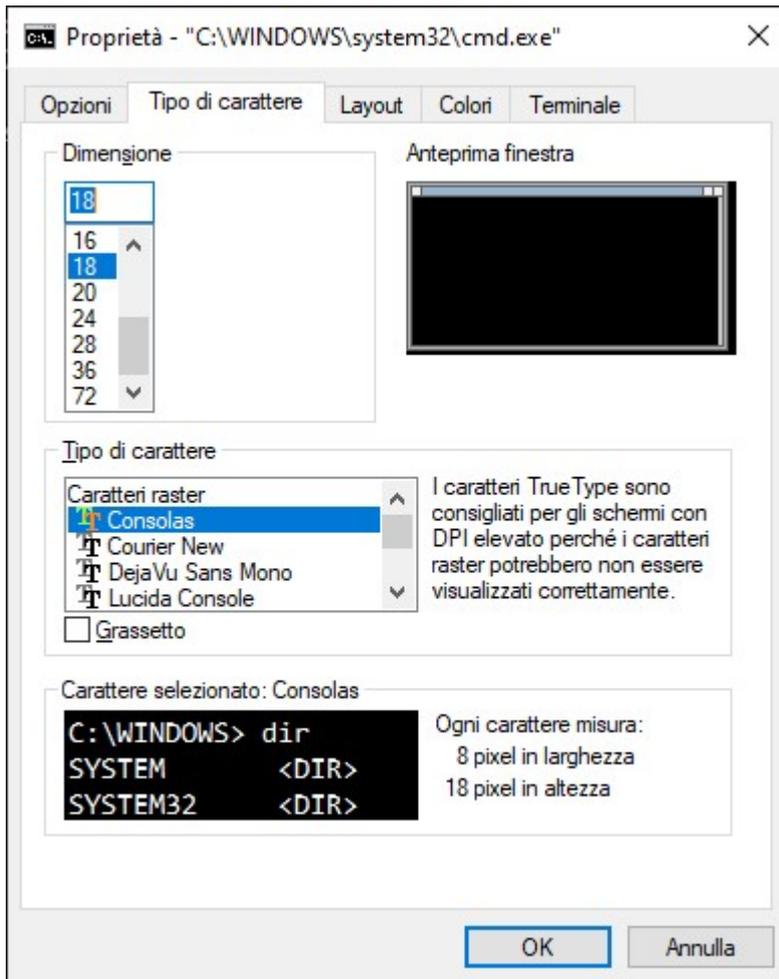
(Prof. Fischetti Pietro)

La shell di comando di Windows (cmd.exe) e' un programma del sistema operativo che permette l'esecuzione di comandi interni, esterni e di scripts. Cmd.exe si trova nella sottodirectory system32 di installazione di windows (es. C:\Windows\System32). Una volta avviata (Esegui->cmd)si presenta come una finestra con sfondo nero:



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versione 10.0.18362.418]
(c) 2019 Microsoft Corporation. Tutti i diritti sono riservati.
C:\Users\acer>_
```

Le caratteristiche della finestra (Font, colori,History, etc.) possono essere modificate cliccando sul menu in alto a sinistra in corrispondenza della voce:'Proprieta':



FILE & PATH

Un file viene individuato da un percorso (path) che può essere assoluto o relativo, dal nome e dall'estensione. Ad es:

C:\inetpub\wwwroot\iisstart.png

rappresenta un percorso assoluto dove C:\inetpub\wwwroot\ rappresenta il Path e C: l'identificativo del disco.

iisstart il nome del file

e .png l'estensione

Se invece sono posizionato nella directory C:\inetpub\wwwroot\

Posso specificare un percorso relativo (relativo alla directory corrente) e digitare semplicemente:

dir iisstart.png

oppure dir .\iisstart.png

se mi trovo nella dir: C:\inetpub\wwwroot\resources

posso utilizzare i due puntini .. e digitare dir ../iisstart.png

e se volessi visualizzare in modo relativo il contenuto del disco C:\ da C:\inetpub\wwwroot\resources?

dir ../../..

N.B. Nel caso in cui siano presenti spazi in un nome questo va racchiuso tra virgolette!!!

Digitando il comando help si ottiene una lista dei comandi interni presenti nella shell.

Name	Description
ASSOC	Visualizza o modifica le associazioni alle estensioni dei file.
ATTRIB	Visualizza o modifica gli attributi del file.
BREAK	Attiva o disattiva il controllo esteso di CTRL+C.
BCDEDIT	Imposta le proprietà nel database di avvio per il controllo del
	caricamento avvio.
CACLS	Visualizza o modifica gli elenchi di controllo di accesso
	(ACL) dei file.
CALL	Richiama un programma batch da un altro.
CD	Visualizza il nome della directory corrente o consente
	di passare a un'altra directory.
CHCP	Visualizza o imposta il numero di tabella codici attiva.
CHDIR	Visualizza il nome della directory corrente o consente
	di passare a un'altra directory.
CHKDSK	Controlla un disco e visualizza il relativo
	rapporto sullo stato.
CHKNTFS	Visualizza o modifica la verifica di un disco durante l'avvio.
CLS	Cancella lo schermo.
CMD	Avvia una nuova istanza dell'interprete dei comandi di Windows.
COLOR	Imposta i colori predefiniti in primo piano e dello sfondo
	della console.
COMP	Confronta il contenuto di due file o di due gruppi di file.
COMPACT	Visualizza o modifica la compressione di file su
	partizioni NTFS.
CONVERT	Converte volumi FAT in NTFS. Non è possibile convertire
	l'unità in uso.
COPY	Copia uno o più file in un'altra posizione.
DATE	Visualizza o imposta la data.
DEL	Elimina uno o più file.
DIR	Visualizza un elenco di file e sottodirectory in una directory.
DISKPART	Visualizza o configura le proprietà di Partizione disco.
DOSKEY	Modifica righe di comando, richiama comandi di Windows e crea macro.

DRIVERQUERY	Visualizza stato e proprietà del driver di dispositivo corrente.
ECHO	Visualizza messaggi o attiva e disattiva la ripetizione a video dei comandi.
ENDLOCAL	Termina la localizzazione di modifiche di ambiente in un file batch.
ERASE	Elimina uno o più file.
EXIT	Termina il programma CMD.EXE (interprete dei comandi).
FC	Confronta due file o set di file e ne visualizza le differenze.
FIND	Ricerca una stringa di testo in uno o più file.
FINDSTR	Ricerca stringhe nei file.
FOR	Esegue un comando specificato per ogni file in un set di file.
FORMAT	Formatta un disco per l'utilizzo con Windows.
FSUTIL	Visualizza o configura le proprietà del file system.
FTYPE	Visualizza o modifica i tipi di file utilizzati nelle associazioni delle estensioni di file.
GOTO	Indirizza l'interprete dei comandi di Windows a una riga con etichetta in un programma batch.
GPRESULT	Visualizza le informazioni relative a Criteri di gruppo per il computer o l'utente.
GRAFTABL	Abilita Windows alla visualizzazione di un set di caratteri estesi in modalità grafica.
HELP	Fornisce informazioni della Guida per i comandi di Windows.
ICACLS	Visualizza, modifica ed esegue il backup o il ripristino degli ACL per file e directory.
IF	Esegue un'elaborazione condizionale in un programma batch.
LABEL	Crea, cambia o elimina l'etichetta di volume di un disco.
MD	Crea una directory.
MKDIR	Crea una directory.
MKLINK	Crea collegamenti simbolici e reali
MODE	Configura un dispositivo di sistema.
MORE	Visualizza l'output una schermata alla volta.
MOVE	Sposta uno o più file da una directory a un'altra directory.
OPENFILES	Visualizza i file aperti dagli utenti remoti per una determinata condivisione di file.
PATH	Visualizza o imposta un percorso di ricerca per file eseguibili.
PAUSE	Sospende l'elaborazione di un file batch e visualizza un messaggio.
POPD	Ripristina il valore precedente della directory

	corrente salvato con PUSHD.
PRINT	Stampa un file di testo.
PROMPT	Cambia il prompt dei comandi di Windows.
PUSHD	Salva la directory corrente e poi la cambia.
RD	Rimuove una directory.
RECOVER	Recupera le informazioni leggibili da un disco danneggiato o difettoso.
REM	Registra commenti (note) in file batch o CONFIG.SYS.
REN	Rinomina uno o piu' file.
RENAME	Rinomina uno o piu' file.
REPLACE	Sostituisce i file.
RMDIR	Rimuove una directory.
ROBOCOPY	Utilità avanzata per la copia di file e alberi di directory
SET	Visualizza, imposta o rimuove variabili di ambiente di Windows.
SETLOCAL	Inizia la localizzazione di modifiche di ambiente in un file batch.
SC	Visualizza o configura i servizi (processi in background).
SCHTASKS	Pianifica comandi e programmi da eseguire su un determinato computer.
SHIFT	Modifica la posizione di parametri sostituibili in file batch.
SHUTDOWN	Consente il corretto arresto del computer in modalità locale e remota.
SORT	Ordina l'input.
START	Avvia una finestra separata per l'esecuzione del programma o comando specificato.
SUBST	Associa il percorso a una lettera di unità
SYSTEMINFO	Visualizza la configurazione e le proprietà specifiche del computer.
TASKLIST	Visualizza tutte le attività in esecuzione inclusi i servizi.
TASKKILL	Interrompe o arresta un processo o un'applicazione in esecuzione.
TIME	Visualizza o imposta l'ora del sistema.
TITLE	Imposta il titolo della finestra per una sessione CMD.EXE.
TREE	Visualizza graficamente la struttura di directory di un'unità o percorso.
TYPE	Visualizza il contenuto di un file di testo.
VER	Visualizza la versione di Windows.
VERIFY	Indica se effettuare o meno la verifica della corretta scrittura dei file sul disco.
VOL	Visualizza l'etichetta di volume e il numero di serie del disco.

XCOPY	Copia file e alberi di directory.
WMIC	Visualizza le informazioni relative a WMI all'interno della shell dei comandi interattivi.

La sintassi dei comandi generalmente si ottiene specificando /? Dopo il nome del comando.

PANORAMICA

La shell dei comandi fornisce un ambiente potente per lavorare con comandi e script. E' possibile eseguire molti tipi di comandi dalla riga di comando, inclusi comandi integrati, utilità di Windows e applicazioni con estensioni di riga di comando. Indipendentemente dalla fonte, ogni comando che si utilizza segue le stesse regole di sintassi. Queste regole affermano che un comando è costituito da un nome di comando seguito da qualsiasi argomento richiesto o facoltativo. Gli argomenti possono anche utilizzare il reindirizzamento per specificare le origini per input, output ed errori.

Quando si esegue un comando nella shell dei comandi, si avvia una serie di eventi simili ai seguenti:

La shell dei comandi sostituisce tutte le variabili che hai inserito nel testo del comando con i loro valori effettivi.

Più comandi che sono concatenati o raggruppati e passati su una sola riga, sono suddivisi in singoli comandi, separati in nome-comando e argomenti correlati. I singoli comandi vengono quindi elaborati.

Se il nome del comando ha un percorso file, la shell dei comandi utilizza questo percorso per trovare il comando. Se il comando non può essere trovato nella posizione specificata, la shell dei comandi restituisce un errore.

Se il nome del comando non specifica un percorso di file, la shell dei comandi tenta di risolvere il nome del comando internamente. Una corrispondenza significa che hai fatto riferimento a un comando integrato che può essere eseguito immediatamente. Se non viene trovata alcuna corrispondenza, la shell dei comandi cerca nella directory corrente l'eseguibile del comando, quindi cerca nei percorsi specificati nella variabile d'ambiente PATH. Invece se viene specificato un percorso nel comando il comando viene cercato nel percorso dell'eseguibile. Se il comando non può essere trovato in nessuna di queste posizioni, la shell dei comandi restituisce un errore.

Se il comando si trova, il comando viene eseguito utilizzando qualsiasi argomento specificato, inclusi quelli che specificano gli input da utilizzare. L'output del comando e gli eventuali errori vengono scritti nella finestra dei comandi o nelle destinazioni specificate per l'output e l'errore.

VARIABILI D'AMBIENTE

Le variabili d'ambiente (*environment variables*) sono utilizzate e impostate in fase di avvio nei principali sistemi operativi per memorizzare alcune impostazioni relative al funzionamento del

sistema operativo o delle applicazioni installate. Di seguito una lista con alcune variabili d'ambiente:

Environment Variable	Path
%ALLUSERSPROFILE%	C:\ProgramData
%APPDATA%	C:\Users\ <i>Username</i> \AppData\Roaming
%COMMONPROGRAMFILES%	C:\Program Files\Common Files
%COMMONPROGRAMFILES(x86)%	C:\Program Files (x86)\Common Files
%COMSPEC%	C:\Windows\System32\cmd.exe
%HOMEDRIVE%	C:
%HOMEPATH%	C:\Users\ <i>Username</i>
%LOCALAPPDATA%	C:\Users\ <i>Username</i> \AppData\Local
%PROGRAMDATA%	C:\ProgramData
%PROGRAMFILES%	C:\Program Files
%PROGRAMFILES(X86)%	C:\Program Files (x86) <i>(only in 64-bit version)</i>
%PUBLIC%	C:\Users\Public
%SystemDrive%	C:
%SystemRoot%	C:\Windows
%TEMP% and %TMP%	C:\Users\ <i>Username</i> \AppData\Local\Temp
%USERPROFILE%	C:\Users\ <i>Username</i>
%WINDIR%	C:\Windows

Per verificarne il valore sul proprio PC digitare nella shell: ECHO + il nome con i simboli di percentuale (%). Ad esempio per visualizzare la directory di installazione di Windows

```
C:\>echo %WINDIR%
```

La lista completa delle variabili d'ambiente di sistema impostate sul proprio Sistema si ottiene con il comando Set:

```
C:\>set
ALLUSERSPROFILE=C:\ProgramData
APPDATA=C:\Users\acer\AppData\Roaming
CommandPromptType=Native
CommonProgramFiles=C:\Program Files\Common Files
CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files
CommonProgramW6432=C:\Program Files\Common Files
COMPUTERNAME=LAPTOP-POJRGMSE
ComSpec=C:\WINDOWS\system32\cmd.exe
DevEnvDir=C:\Program Files (x86)\Microsoft Visual Studio\2017\Community\Common7\IDE\
DriverData=C:\Windows\System32\Drivers\DriverData
ExtensionSdkDir=C:\Program Files (x86)\Microsoft SDKs\Windows Kits\10\ExtensionSDKs
FPS_BROWSER_APP_PROFILE_STRING=Internet Explorer
FPS_BROWSER_USER_PROFILE_STRING=Default
```

Framework40Version=v4.0
FrameworkDir=C:\Windows\Microsoft.NET\Framework64\
FrameworkDir64=C:\Windows\Microsoft.NET\Framework64\
FrameworkVersion=v4.0.30319
FrameworkVersion64=v4.0.30319
FSHARPINSTALLDIR=C:\Program Files (x86)\Microsoft SDKs\F#\10.1\Framework\v4.0\
FTSDK_DIR=C:\Program Files\Microsoft SDKs\Kinect\Developer Toolkit v1.8.0\
HOMEDRIVE=C:
HOMEPATH=\Users\acer
HTMLHelpDir=C:\Program Files (x86)\HTML Help Workshop
INCLUDE=C:\Program Files (x86)\Microsoft Visual
Studio\2017\Community\VC\Tools\MSVC\14.13.26128\ATLMFC\include;C:\Program Files (x86)\Microsoft
Visual Studio\2017\Community\VC\Tools\MSVC\14.13.26128\include;C:\Program Files (x86)\Windows
Kits\NETFXSDK\4.6.1\include\um;C:\Program Files (x86)\Windows
Kits\10\include\10.0.16299.0\ucrt;C:\Program Files (x86)\Windows
Kits\10\include\10.0.16299.0\shared;C:\Program Files (x86)\Windows
Kits\10\include\10.0.16299.0\um;C:\Program Files (x86)\Windows
Kits\10\include\10.0.16299.0\winrt;C:\Program Files (x86)\Windows
Kits\10\include\10.0.16299.0\cppwinrt
KINECTSDK10_DIR=C:\Program Files\Microsoft SDKs\Kinect\v1.8\
KINECT_TOOLKIT_DIR=C:\Program Files\Microsoft SDKs\Kinect\Developer Toolkit v1.8.0\
LIB=C:\Program Files (x86)\Microsoft Visual
Studio\2017\Community\VC\Tools\MSVC\14.13.26128\ATLMFC\lib\x64;C:\Program Files (x86)\Microsoft
Visual Studio\2017\Community\VC\Tools\MSVC\14.13.26128\lib\x64;C:\Program Files (x86)\Windows
Kits\NETFXSDK\4.6.1\lib\um\x64;C:\Program Files (x86)\Windows
Kits\10\lib\10.0.16299.0\ucrt\x64;C:\Program Files (x86)\Windows Kits\10\lib\10.0.16299.0\um\x64;
LIBPATH=C:\Program Files (x86)\Microsoft Visual
Studio\2017\Community\VC\Tools\MSVC\14.13.26128\ATLMFC\lib\x64;C:\Program Files (x86)\Microsoft
Visual Studio\2017\Community\VC\Tools\MSVC\14.13.26128\lib\x64;C:\Program Files (x86)\Microsoft
Visual Studio\2017\Community\VC\Tools\MSVC\14.13.26128\lib\x86\store\references;C:\Program Files
(x86)\Windows Kits\10\UnionMetadata\10.0.16299.0;C:\Program Files (x86)\Windows
Kits\10\References\10.0.16299.0;C:\Windows\Microsoft.NET\Framework64\v4.0.30319;
LOCALAPPDATA=C:\Users\acer\AppData\Local
LOGONSERVER=\\LAPTOP-POJRGMSE
MOZ_PLUGIN_PATH=C:\Program Files (x86)\Foxit Software\Foxit Reader\plugins\
NETFXSDKDir=C:\Program Files (x86)\Windows Kits\NETFXSDK\4.6.1\
NUMBER_OF_PROCESSORS=4
OneDrive=C:\Users\acer\OneDrive
OS=Windows_NT
Path=C:\Program Files (x86)\Microsoft Visual
Studio\2017\Community\VC\Tools\MSVC\14.13.26128\bin\HostX64\x64;C:\Program Files (x86)\Microsoft
Visual Studio\2017\Community\Common7\IDE\VC\VCackages;C:\Program Files (x86)\Microsoft
SDKs\TypeScript\2.6;C:\Program Files (x86)\Microsoft Visual
Studio\2017\Community\Common7\IDE\CommonExtensions\Microsoft\TestWindow;C:\Program Files
(x86)\Microsoft Visual
Studio\2017\Community\Common7\IDE\CommonExtensions\Microsoft\TeamFoundation\Team
Explorer;C:\Program Files (x86)\Microsoft Visual
Studio\2017\Community\MSBuild\15.0\bin\Roslyn;C:\Program Files (x86)\Microsoft Visual
Studio\2017\Community\Team Tools\Performance Tools\x64;C:\Program Files (x86)\Microsoft Visual
Studio\2017\Community\Team Tools\Performance Tools;C:\Program Files (x86)\Microsoft Visual
Studio\Shared\Common\VSPerfCollectionTools\x64;C:\Program Files (x86)\Microsoft Visual
Studio\Shared\Common\VSPerfCollectionTools\;C:\Program Files (x86)\Microsoft
SDKs\Windows\v10.0A\bin\NETFX 4.6.1 Tools\x64;C:\Program Files (x86)\HTML Help

```

Workshop;C:\Program Files (x86)\Microsoft SDKs\F#\10.1\Framework\v4.0\;C:\Program Files
(x86)\Windows Kits\10\bin\10.0.16299.0\x64;C:\Program Files (x86)\Windows Kits\10\bin\x64;C:\Program
Files (x86)\Microsoft Visual
Studio\2017\Community\MSBuild\15.0\bin;C:\Windows\Microsoft.NET\Framework64\v4.0.30319;C:\Prog
ram Files (x86)\Microsoft Visual Studio\2017\Community\Common7\IDE\;C:\Program Files (x86)\Microsoft
Visual
Studio\2017\Community\Common7\Tools\;C:\ProgramData\Oracle\Java\javapath;C:\WINDOWS\system32
;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\Pro
gram Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\;C:\Program Files (x86)\Microsoft SQL
Server\140\Tools\Binn\;C:\Program Files\Microsoft SQL Server\140\Tools\Binn\;C:\Program
Files\Microsoft SQL Server\140\DTS\Binn\;C:\Program Files (x86)\Microsoft SQL Server\Client
SDK\ODBC\130\Tools\Binn\;C:\Program Files (x86)\Microsoft SQL Server\140\DTS\Binn\;C:\Program Files
(x86)\Microsoft SQL Server\140\Tools\Binn\ManagementStudio\;C:\Program Files\dotnet\;C:\Program
Files\Microsoft SQL
Server\130\Tools\Binn\;C:\WINDOWS\System32\OpenSSH\;C:\Users\acer\AppData\Local\Microsoft\Wind
owsApps
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
Platform=x64
PROCESSOR_ARCHITECTURE=AMD64
PROCESSOR_IDENTIFIER=AMD64 Family 21 Model 101 Stepping 1, AuthenticAMD
PROCESSOR_LEVEL=21
PROCESSOR_REVISION=6501
ProgramData=C:\ProgramData
ProgramFiles=C:\Program Files
ProgramFiles(x86)=C:\Program Files (x86)
ProgramW6432=C:\Program Files
PROMPT=$P$G
PSModulePath=%ProgramFiles%\WindowsPowerShell\Modules;C:\WINDOWS\system32\WindowsPowerSh
ell\v1.0\Modules;C:\Program Files (x86)\Microsoft SQL Server\140\Tools\PowerShell\Modules\
PUBLIC=C:\Users\Public
SESSIONNAME=Console
SystemDrive=C:
SystemRoot=C:\WINDOWS
TEMP=C:\Users\acer\AppData\Local\Temp
TMP=C:\Users\acer\AppData\Local\Temp
.....

```

Importante per chi programma (specialmente in Linguaggio C) sono le variabili PATH,INCLUDE e LIB.

La variabile PATH indica i percorsi di ricerca degli eseguibili, cioè se lancio un programma dalla shell specificandone solo il nome, se questo non viene trovato nella directory corrente il S.O. non va alla ricerca nell'intero Disco ma nei percorsi specificati nella variabile PATH. La variabile INCLUDE indica i percorsi dove si trovano i file con estensione .h del C (ad esempio stdio.h), e LIB i file di libreria statica che hanno estensione .lib (ad esempio se volessi sviluppare in C applicazioni multimediali per Windows probabilmente utilizzerò la libreria di Microsoft: mmdevapi.lib).

REINDIRIZZAMENTO DI INPUT, OUTPUT ED STANDARD ERROR

Per impostazione predefinita, i comandi ricevono l'input da tastiera e inviano quindi l'output, compresi i messaggi di errore alla finestra della console standard. Tuttavia è possibile ricevere l'input da altra origine tipicamente un file e inviare l'output a un file o ad una periferica di output ad esempio una stampante. E' possibile inviare inoltre gli errori ad un file anziché alla finestra di console. Queste e altre operazioni di reindirizzamento sono riportate nella Tab. seguente.

Table: Redirection Techniques for Input, Output, and Errors

Redirection Technique	Description
command1 command2	Sends the output of the first command to be the input of the second command.
command < [path]filename	Takes command input from the specified file path.
command > [path]filename	Sends output to the named file, creating the file if necessary or overwriting it if it already exists.
command >> [path]filename	Appends output to the named file if it exists or creates the file and then writes to it.
command < [path]filename > [path]filename	Gets command input from the specified file and then sends command output to the named file.
command < [path]filename >> [path]filename	Gets command input from the specified file and then appends command output to the named file.
command 2> [path]filename	Creates the named file and sends any error output to it. If the file exists, it is overwritten.
command 2>&1 filename	Sends error output to the same destination as standard output.

Reindirizzamento degli output standard ad altri comandi

La maggior parte dei comandi genera un output che puo' essere reindirizzato a un altro comando come input. A tale scopo viene utilizzata una tecnica nota come piping mediante la quale l'output di un comando viene inviato come input al comando successivo. La sintassi per il piping e':

comando1 | comando2 | comando3 ...

Ad esempio se si vuole visualizzare in ordine alfabetico i nomi dei file contenuti nella cartella c:\windows

Dir /B c:\windows | sort

Con:

Dir /B c:\windows | sort /R

Visualizza in ordine alfabetico inverso.

Se l' output di un comando e' particolarmente corposo si puo' visualizzare una pagina alla volta:

dir | more

Reindirizzamento I/O ai file e dai file

Un'altra tecnica di reindirizzamento dei comandi consiste nell'ottenere l'input da un file utilizzando il simbolo di reindirizzamento di input (<). Ad esempio il seguente comando ordina il contenuto del file users.txt:

sort < users.txt

Ma e' anche possibile inviare l'output a un file, con il simbolo (>) per sovrascrivere su un file o con il simbolo (>>) per aggiungere in coda ad un file. Ad esempio se si desidera salvare lo stato della rete in un file, si puo' utilizzare il comando:

```
netstat -a > netstatus.txt
```

Si possono combinare i simboli di reindirizzamento come ad esempio:

```
sort < users.txt > userso.txt
```

Si utilizzeranno le tecniche qui descritte soprattutto per salvare su file l'output dei programmi scritti in C al fine di visualizzare graficamente su foglio elettronico.

Reindirizzamento degli errori

Per impostazione predefinita gli errori dei comandi sono scritti nella finestra di console, ma si possono anch'essi scrivere su file. A questo scopo si utilizzano i numeri di identificazione dell'I/O (0 per lo standard input, 1 per lo standard output e 2 per lo standard error). Ad esempio se vogliamo inviare i messaggi di errore del comando `chkdsk` su file:

```
chkdsk /r 2> diskerr.txt
```

Invece per scrivere output ed errori sul medesimo file:

```
chkdsk /r > diskerr.txt 2>&1
```

Concatenamento e raggruppamento di comandi

E' possibile concatenare comandi ed eseguirli in sequenza anche in modo condizionale alla riuscita o meno dei comandi precedenti. E' inoltre possibile raggruppare insieme di comandi da eseguire in modo condizionale. Queste tecniche sono riportate nella tabella seguente.

Table: Quick Reference for Chaining and Grouping Commands		
Symbol	Syntax	Description
&	Command1 & Command2	Execute Command1 and then execute Command2.
&&	Command1 && Command2	Execute Command2 if Command1 is completed successfully.
	Command1 Command2	Execute Command2 only when Command1 doesn't complete successfully.
()	(Command1 & Command2) && (Command3)	Use parentheses to group sets of commands for conditional execution based on success.
	(Command1 & Command2) (Command3)	Use parentheses to group sets of commands for conditional execution based on failure.

Esempio cambiare una directory e visualizzarne il contenuto in una unica riga:

```
C:\Programmi>cd c:\tmp & dir /B
```

Se la directory `c:\tmp` non esiste viene stampato: Impossibile trovare il percorso specificato. E visualizzato il contenuto di `C:\Programmi`. Se si vuole impedire l'esecuzione del secondo se il primo fallisce:

```
C:\Programmi>cd c:\tmp && dir /B
```

Ad esempio se si vuole che un file venga spostato solo se esiste:

```
dir c:\temp\dati.txt && move dati.txt D:\hist\dati_n.txt
```

Ad esempio se si vuole utilizzare il seguente comando ma non si è sicuri circa la locazione della directory data che potrebbe trovarsi sotto la directory c:\working o sotto la radice c:\:

```
cd C:\working\data || cd C:\data
```

Infine se si vuole ad esempio salvare su file il nome del PC l'indirizzo di rete e lo stato delle connessioni:

```
C:\>hostname & ipconfig & netstat -a > z1.txt
```

Ma se si esamina il contenuto del file z1.txt si nota solo i dati relativi all'ultimo comando circa le connessioni di rete. Se vuole scrivere il contenuto di tutti i comandi si utilizza il raggruppamento:

```
C:\>(hostname & ipconfig & netstat -a) > z2.txt
```

LE VARIABILI D'AMBIENTE

Negli script da riga di comando, ciò che comunemente chiamiamo variabili sono più propriamente chiamate variabili di ambiente. Le variabili d'ambiente possono provenire da molte fonti. Alcune variabili sono integrate nel sistema operativo o derivate dall'hardware del sistema durante l'avvio. Queste variabili, chiamate variabili di sistema integrate, sono disponibili per tutti i processi Windows indipendentemente dal fatto che qualcuno abbia effettuato l'accesso in modo interattivo. Le variabili di sistema possono anche provenire dal registro di Windows. Altre variabili vengono impostate durante l'accesso e sono chiamate variabili utente integrate. Le variabili utente integrate disponibili sono le stesse, indipendentemente da chi è connesso al computer. Come ci si potrebbe aspettare, sono validi solo durante una sessione di accesso effettiva, ovvero quando un utente è connesso.

Puoi vedere un elenco di tutte le variabili conosciute nell'istanza corrente della shell dei comandi digitando set al prompt. Oltre al normale sistema e alle variabili utente, puoi creare variabili ogni volta che Windows è in esecuzione, che è esattamente ciò che farai quando programmi nella shell dei comandi. Definisci le variabili per l'istanza corrente della shell dei comandi utilizzando il comando SET e la seguente sintassi:

```
SET nome_variabile=valore_variabile
```

Esempi:

```
set working=C:\Work\Data  
set value=5  
set string="Hello World"
```

script:

```
@echo off  
SET a = 5  
SET b = 10  
SET c = %a% + %b%  
echo %c%
```

Una variabile di ambiente molto importante è errorlevel, che tiene traccia del codice di uscita del comando utilizzato più di recente. Per convenzione consolidate nei decenni, se il comando viene eseguito

normalmente, il livello di errore è zero (0). Se si verifica un errore durante l'esecuzione del comando, il livello di errore viene impostato su un valore diverso da zero appropriato.

Vediamo velocemente un esempio:

```
C:\>dir
...
13/06/2021 13:49 <DIR>    Searches
26/09/2021 22:31 <DIR>    source
29/07/2021 00:15          367 Sti_Trace.log
05/08/2021 23:37 <DIR>    Videos
...
```

Esiste un file di nome Sti_Trace.log

Quindi se digito:

```
C:\>dir Sti_Trace.log
...
29/07/2021 00:15          367 Sti_Trace.log
          1 File          367 byte
          0 Directory 236.269.998.080 byte disponibili

C:\>echo %errorlevel%
0
```

Dato che il file esiste errorlevel ritorna 0.

MA se ora digito un file non esistente:

```
C:\>dir Sti_Trace.lxg
...
File non trovato

C:\>echo %errorlevel%
1
```

Si nota che la variabile errorlevel restituisce un valore diverso da zero in caso di errore

Vediamo ora un semplice esempio di programma in C, e lo compiliamo in el.exe:

```
el.c
int main(){
    return 0;
}
```

Di seguito un semplice script che lo utilizza:

```
go.bat
el.exe
echo off
echo %errorlevel%
if "%ERRORLEVEL%"=="0" (GOTO QUIT) ELSE (echo "Error %errorlevel%")

:QUIT
echo "Bye"
```

Eseguendo il bat precedente si ottiene a video:

```
0
"Bye"
```

Ma se si modifica il codice in C:

```
el.c
int main(){
    return 1;
}
```

Si ottiene:

```
1
"Error 1"
"Bye"
```

Che dimostra il diverso comportamento dello script a seconda del valore ritornato dal programma.

Quando si finisce di lavorare con le variabili, è buona norma eliminarle. Si fa per liberare la memoria utilizzata dalla variabile e prevenire problemi o risultati imprevisti se si fa riferimento accidentalmente alla variabile in futuro.

Ad esempio lo script seguente:

```
Es.bat
SET V1="CIAO"
ECHO %V1%
```

Produce a video il risultato seguente;

```
C:\>"CIAO"
```

Il problema si presenta se eseguo da shell:

```
C:\> echo %v1%
```

Riottengo

```
C:\>"CIAO"
```

Quindi di seguito lo script corretto che evita effetti indesiderati:

```
Es.bat
SET V1="CIAO"
ECHO %V1%
SET V1=
```

Per cancellare una variabile, semplicemente impostare la variabile uguale a niente, come ad esempio

```
set val1=
```

Ora la variabile è stata cancellata dalla memoria e non è più disponibile.