

Interfaccia da linea di comando – Command Line Interface (CLI) in Windows

Lezione introduttiva

CLI: Command Line Interface (interfaccia da linea di comando)

vs

GUI: Graphical User Interface (interfaccia utente di tipo grafico)

Cosa sono?

2 differenti modalità di interazione con il Sistema Operativo ed i suoi programmi applicativi.

L'interfaccia utente di tipo grafico (GUI) richiede strumenti hardware specifici per la sua piena funzionalità. Ad esempio, può essere necessario l'uso di un mouse, uno schermo touch, o altri dispositivi di input.

La CLI, invece, ovvero l'interfaccia da linea di comando, permette di interagire con il Sistema Operativo e i suoi programmi applicativi mediante l'utilizzo della "semplice" tastiera.

Per aprire la Command Line di Windows, utilizza la seguente combinazione di tasti:

Tasto LOGO di Windows + tasto r; poi, scrivi *cmd* e premi invio.

Lezione 2

Quando si apre la CLI, si viene automaticamente posizionati nel **percorso**: `C:\Users\nomeutente` (del/nel filesystem – riprenderemo questo concetto in seguito). È come se, utilizzando la GUI, si navigasse manualmente tra icone e finestre, fino a raggiungere l'unità *C*, poi la cartella *Users*, e infine la cartella *nomeutente*.

Una volta all'interno di questa cartella nella GUI, si possono eseguire varie operazioni, come la creazione di una nuova cartella o di un file. Allo stesso modo, nella CLI, qualsiasi comando che digitiamo verrà eseguito in quel percorso specifico, operando sui files e le cartelle presenti in quella specifica posizione.

Cosa significa C:?

C: si riferisce ad una *unità* correlata al disco fisso. Ovvero: entità *logica* non necessariamente corrispondente alla totalità dell'entità fisica/dell'unità fisica/del disco fisso.

Più in dettaglio, la lettera “C” rappresenta un'unità di archiviazione nel computer, solitamente un disco fisso o un'unità a stato solido (SSD). In Windows, le unità di archiviazione vengono assegnate con lettere dell'alfabeto, e “C” è di solito la lettera assegnata all'unità principale, dove è installato il sistema operativo.

Quando si parla di “C” come “entità logica”, ci si riferisce al fatto che questa rappresentazione non corrisponde necessariamente all'intera unità fisica (l'hardware effettivo del disco).

Ad esempio, un disco fisso di 500 GB potrebbe essere suddiviso (o **partizionato**) in più unità logiche, come “C”, “D”, “E”, ecc. In questo caso, l'unità “C” potrebbe avere un'assegnazione di spazio di, ad esempio, 100 GB, mentre le restanti parti del disco potrebbero essere assegnate ad altre unità logiche (ad esempio “D”).

La scelta di suddividere, o per meglio dire, **partizionare**, un singolo disco fisico in più unità logiche, ormai divenuta una consuetudine ampiamente adottata, permette, quindi, una gestione più flessibile dei dati e del sistema operativo.

Ad esempio: potrei facilmente procedere alla formattazione dell'unità “C” (o alla reinstallazione del sistema operativo) lasciando intatta la seconda unità, nella quale potrebbe altrimenti essere conveniente memorizzare/preservare i propri dati.

Come anticipato, riprendiamo e ribadiamo il significato di `C:\Users\nomeutente`

Quella stringa di testo, “banalmente”, rappresenta un “**percorso** all'interno del *computer*”. Ma potremmo chiederci: un percorso che conduce dove esattamente? E poi, è davvero corretto dire “percorso all'interno del *computer*”?

La risposta sta nel contesto di riferimento a cui il percorso appartiene, che in questo caso è il **Filesystem** del computer. Quindi, `C:\Users\nomeutente` non è solo una stringa di caratteri, né un percorso “dentro il computer”, bensì indica una specifica posizione all'interno della struttura organizzativa del Filesystem, che ci permette di identificare/di comprendere il percorso completo all'interno di esso.

Adesso possiamo meglio definire il termine **Filesystem**, ovvero: modalità con la quale sono organizzati/rappresentati i contenuti (dati) all'interno di una unità disco.

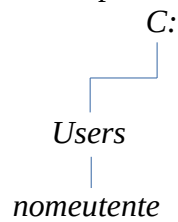
Il Filesystem viene spesso visualizzato come una struttura ad albero, dove le cartelle possono contenere altre cartelle o file.

Nel percorso di esempio, `C:\Users\nomeutente`, la cartella *nomeutente* si trova all'interno della cartella *Users*. La cartella *Users* si trova, a sua volta, all'interno dell'unità disco *C*:

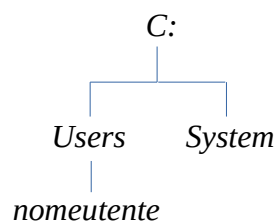
Questa modalità organizzativa crea una relazione gerarchica o di “parentela”, dove ogni livello superiore contiene uno o più livelli inferiori. Tornando al nostro caso specifico, *C*: è il livello più

alto (la radice dell'albero), *Users* è un livello subordinato, e *nomeutente* è ulteriormente subordinato a *Users*. Questa struttura permette di navigare facilmente tra i file e le cartelle e di capire la loro posizione all'interno del filesystem.

Ecco una semplice rappresentazione del percorso *C:\Users\nomeutente*:



Qualora ad esempio l'unità *C:* contenesse, oltre alla cartella *Users*, anche la cartella *System*, quest'ultima sarebbe posizionata allo stesso livello di *Users*, ovvero:



Il tal caso, le cartelle *Users* e *System* potrebbero essere considerate “fratelli” (o “sorelle”), entrambe “figlie” dell'unità *C:*.

nomeutente, invece, è “figlio” di *Users* e, al contempo, nipote di *C:*.

Lezione 3

Si riportano, di seguito, i comandi essenziali con cui interagire con la CLI ed il filesystem.

- **cd ..**
Questo comando consente di uscire dalla cartella corrente e di accedere alla cartella genitore, ovvero di salire di un livello nella struttura ad albero del filesystem.
- **dir**
Utilizzato per visualizzare il contenuto della directory corrente, mostrando sia file che cartelle presenti in quella directory.
- **cd [percorso]:** Permette di cambiare la cartella corrente, spostandoti nella cartella o nel percorso specificato. Il percorso può essere **assoluto** o **relativo** (di seguito si riporta un breve inciso con dettagli e differenze).
- **mkdir [nomecartella]:** Utilizzato per creare una nuova cartella. Il nome della cartella da creare viene specificato dopo il comando.

Questi comandi sono fondamentali per la navigazione e la gestione dei file e delle cartelle all'interno del filesystem da una CLI.

Percorso assoluto: Un percorso assoluto è la descrizione completa di un percorso di file o cartella, a partire dalla radice del file system. Esso fornisce l'intera sequenza di directory che devono essere attraversate per raggiungere il file o la cartella desiderata.

Ad esempio, il percorso assoluto `C:\Users\nomeutente\file.txt` indica la posizione esatta del file, partendo dall'unità disco C:. Mentre `C:\Users\nomeutente\Desktop` indica la posizione esatta (ovvero il percorso assoluto) della cartella *Desktop*, partendo dall'unità radice C:.

Percorso relativo: A differenza del percorso assoluto, un percorso relativo è definito in relazione alla directory corrente in cui ci si trova. Non inizia dalla radice del file system, ma utilizza la posizione corrente come punto di partenza.

Ad esempio, se ci troviamo nella directory `C:\Users\nomeutente` (così come comunemente accade all'apertura della command line), e vogliamo accedere alla cartella *Documenti*, contenuta in *nomeutente*, il percorso relativo sarà semplicemente *Documenti*. Dunque per accedere alla cartella *Documenti*, nella CLI mi basterà impartire il comando:

`cd Documenti`

piuttosto che `cd C:\Users\nomeutente\Documenti`, che sarebbe altresì corretto, poiché rappresentativo di un percorso assoluto corretto, ma ovviamente più “dispendioso” da trascrivere.

Creazione di **script** da eseguire all'interno della CLI

La CLI consente l'automazione di processi attraverso l'uso di **script**. Gli script sono essenzialmente file di testo (con estensione *.bat*) che contengono una sequenza di comandi CLI.

Quando uno script viene eseguito, ovvero viene avviato come fosse un programma (con un doppio click!), i comandi contenuti al suo interno vengono eseguiti in ordine sequenziale (uno di seguito all'altro, partendo dal primo).

Questa modalità, **automatizza** le attività che altrimenti dovrebbero essere eseguite manualmente, cioè una alla volta attraverso la linea di comando o, peggio ancora, mediante la GUI!

Utilizzando gli script, quindi, è possibile **automatizzare** sequenze di operazioni, anche quelle più complesse, rendendo la/le attività da svolgere (l'interno processo) più efficiente, facilmente trasferibile su altri PC (è sufficiente trasferire lo script ed avviarlo), meno soggetta ad eventuali errori umani.

Note conclusive, breve riepilogo

All'apertura della Command Line, il percorso di default generalmente è:

`C:\Users\nomeutente`.

Questo significa che, salvo eccezioni, possiamo ragionevolmente aspettarci che questo percorso effettivamente esista nel sistema. Inoltre, spesso rappresenta il punto di partenza per l'esecuzione degli script.

In termini pratici, ciò suggerisce che gli script dovrebbero essere salvati in questo percorso, o almeno in una posizione accessibile da esso, per poter essere eseguiti efficacemente.

Quando si crea uno script, quindi, potete partire da questa assunzione come base.

Tuttavia, si incoraggia ad includere una nota introduttiva o esplicativa che chiarisca tale assunto.

Per accedere alla cartella *Desktop* da tale percorso, mi è sufficiente impartire il comando:

`cd Desktop`

Il percorso in cui sono posizionato adesso è:

`C:\Users\nomeutente\Desktop`

Creiamo una nuova cartella (all'interno di *Desktop*):

`mkdir 21-nov-23`

Entriamo dentro la cartella 21-nov-23:

`cd 21-nov-23`

Creiamo un file vuoto, dal nome *script.bat*:

`type NUL > script.bat`

Adesso visualizziamo il contenuto della cartella (comando **`dir`**): sicuramente è presente il file *script.bat* precedentemente creato!

In aggiunta – aspetto estremamente importante –, sono al contempo elencate due ulteriori “pseudocartelle”: **punto e punto-punto.**

La cartella *punto-punto* (..) si riferisce sempre alla cartella padre, ovvero alla cartella che contiene la cartella corrente. Nel nostro caso specifico, la cartella *punto-punto* si riferisce quindi a *Desktop*. Più in generale, per salire di un livello all'interno del filesystem, ovvero per accedere alla propria cartella “padre”, mi basterà impartire il comando:

`cd ..`

Dalla posizione corrente, per accedere alla cartella “padre del padre”, ovvero al “nonno”, o per meglio dire, per salire di 2 livelli nell'organizzazione gerarchica del filesystem, dovrò impartire:

`cd ..`

`cd ..`

Si noterà che, al contempo, il *prompt* si modificherà di conseguenza, sempre indicando la posizione corrente all'interno del filesystem.

Ad esempio:

```
C:\Users\nomeutente\Desktop> cd ..  
C:\Users\nomeutente> cd ..  
C:\Users>
```

Adesso apriamo il file *script.bat* direttamente dalla CLI:

notepad script.bat

Il file *script.bat* verrà aperto con il programma Notepad.

Possiamo adesso procedere alla creazione del nostro primo script.

L'obiettivo dello *script* è il seguente e si articola in 2 punti:

1. creare una cartella dal nome *pippo*.
2. creare un file vuoto, dal nome *pluto.txt*, all'interno della cartella *pippo*.

Risoluzione

Ovviamente i comandi sarebbe opportuno inserirli all'interno di uno script. Quindi, per prima cosa debbo crearlo.

Apro la command line.

Sono presumibilmente posizionato in *C:\Users\nomeutente>*

Accedo al Desktop: *C:\Users\nomeutente> cd Desktop*

Creo un file vuoto: *C:\Users\nomeutente\Desktop> type NUL > script.bat*

Apro il file con Notepad, affinché possa inserirvi i comandi richiesti: *C:\Users\nomeutente\Desktop> notepad script.bat*

All'interno dello script, adesso, posso inserire i comandi richiesti dall'esercizio, ovvero:

mkdir pippo

cd pippo

type NUL > pluto.txt

Lo script, se eseguito, creerà la cartella *C:\Users\nomeutente\Desktop\pippo* ed il file *pluto.txt* al suo interno.

Lezione 4

Esercizio per casa

Creare uno script *bat* che automatizzi il seguente processo:

1. Posizionarsi all'interno della cartella Desktop.
2. Creare una nuova cartella, avente per nome la data odierna.
3. Accedere alla cartella appena creata.
4. Creare un file vuoto, dal nome *hello.txt*, all'interno della cartella padre.

Risoluzione

[Come prima cosa, creare uno script *.bat* vuoto]

```
cd Desktop
mkdir 28-nov-23
cd 28-nov-23
cd ..
type NUL > hello.txt
```