

GUIDA – TASKLIST E TASKKILL

A CURA DI FRANCESCO DE DOMINICIS

Tasklist e Taskkill sono due tra i più utili comandi per DOS. Permettono infatti la completa gestione dei processi in esecuzione sul sistema ed offrono maggiori potenzialità del classico Task Manager. In questa guida verrà spiegato il loro utilizzo.

ATTENZIONE: per questa guida è consigliata (ma non necessaria) una minima conoscenza del funzionamento del prompt dei comandi di Windows.

TASKLIST

Il comando tasklist è un comando del prompt per la visualizzazione dei processi in esecuzione sul PC. Essendo a riga di comando, il comando, utilizzato assieme al Taskkill, ha maggiori funzionalità del classico Task Manager. Inoltre, tramite quello, si possono gestire anche i processi di altri computer comandati in remoto.

A questo punto, possiamo cominciare dando un'occhiata a come appare il prompt dopo l'esecuzione del comando:

```
Microsoft Windows [Versione 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. Tutti i diritti riservati.

C:\Users\...>tasklist
```

Nome immagine	PID	Nome sessione	Sessione n.	Utilizzo mem
System Idle Process	0	Services	0	24 K
System	4	Services	0	1.320 K
smss.exe	544	Services	0	560 K
csrss.exe	616	Services	0	4.880 K
wininit.exe	676	Services	0	3.820 K
csrss.exe	688	Console	1	12.540 K
services.exe	720	Services	0	6.472 K
lsass.exe	732	Services	0	1.916 K
lsm.exe	744	Services	0	4.856 K
winlogon.exe	908	Console	1	4.860 K
svchost.exe	936	Services	0	6.792 K
nvsvc.exe	984	Services	0	3.300 K
svchost.exe	1012	Services	0	7.152 K
svchost.exe	1072	Services	0	15.032 K
svchost.exe	1152	Services	0	11.444 K
svchost.exe	1180	Services	0	54.940 K
svchost.exe	1196	Services	0	57.756 K
audiodg.exe	1320	Services	0	12.972 K

Si può notare che ci sono cinque diverse colonne; osserviamole una ad una. La prima, quella chiamata **"Nome immagine"**, indica il nome dei processi. La seconda, chiamata **PID**, indica il **ProcessID** (o **Process Identifier**); si tratta di un numero esadecimale (in genere a quattro cifre), che determina un solo processo; più tardi vedremo come utilizzarlo per bloccare i vari processi. Per ora ci basta sapere che ce li identifica. Diamo uno sguardo alle colonne **tre** ("**Nome sessione**") e **quattro** ("**Sessione n.**"): indicano rispettivamente il nome ed il numero associato della sessione di lavoro; quando si trova scritto **"Services"**, vuol dire che la sessione è di sistema, perciò bisogna lavorarci con prudenza; quando invece c'è scritto **"console"** indica la sessione dell'utente che al momento lavora da prompt. Infine, la quinta colonna ("**Utilizzo mem**") indica la quantità di memoria RAM richiesta dal processo.

Detto questo, possiamo iniziare ad osservare i vari parametri del comando.

Il primo parametro che andiamo a vedere è **/s**. Questo parametro permette di stabilire su quale computer della rete locale lavorare. Nel prompt si utilizza così:

```
tasklist /s NomeComputer
```

Ovviamente NomeComputer va sostituito con il nome del PC sulla rete locale, Quando il comando non viene specificato, viene usato di default il computer locale.

Il prossimo parametro che andiamo ad analizzare è il parametro `/u`. Quando si setta questo parametro si scelgono quali permessi utenti utilizzare. Si utilizza così:

```
tasklist /u Dominio\Utente
```

Scegliendo un utente, si utilizza il comando con i permessi di suddetto utente. Il dominio può anche non essere inserito. Quando non viene specificato questo parametro si utilizza l'utente attualmente attivo.

Nel caso l'utente sia protetto da password, è necessario aggiungere, subito dopo, un altro parametro, il parametro `/p`. A questo punto, sul prompt si digiterà così:

```
tasklist /u Dominio\Utente /p Password
```

Per impostare il metodo di visualizzazione si usa il parametro `/fo`. Questo parametro ammette solo tre tipi di impostazioni: table, list e csv. Quando non si specifica il sistema imposterà in automatico "table". Se, per esempio, vogliamo vedere i processi utilizzando come impostazione "list", allora scriveremo così:

```
tasklist /fo list
```

Stesso modo per impostare csv. Ecco come appare il prompt quando si imposta "list":

```
PID: 1196
Nome sessione: Services
Sessione n.: 0
Utilizzo memoria: 35.464 K

Nome immagine: audiodg.exe
PID: 1320
Nome sessione: Services
Sessione n.: 0
Utilizzo memoria: 13.264 K

Nome immagine: svchost.exe
PID: 1344
Nome sessione: Services
Sessione n.: 0
Utilizzo memoria: 4.316 K

Nome immagine: SLsvc.exe
PID: 1368
Nome sessione: Services
Sessione n.: 0
Utilizzo memoria: 3.748 K

Nome immagine: svchost.exe
PID: 1436
```

Se, quando si sceglie di visualizzare o in "table" o in "csv", non si vogliono visualizzare le intestazioni delle colonne, allora si ricorre al parametro `/nh` (nh sta per "no headers"). Perciò si digiterà (ad esempio) così:

```
tasklist /fo csv /nh
```

È ovvio che quando il parametro `/fo` non è inserito il parametro `/nh` si applicherà alla visualizzazione di default.

Ora andiamo a vedere uno dei parametri più utili del comando, il parametro `/fi`. Questo parametro (che

poi vedremo sarà utile anche con taskkill) permette di visualizzare solo i processi con determinate caratteristiche. Il comando si scrive così:

tasklist /fi Nome Operatore Valore

Il nome rappresenta il tipo di filtro, il valore indica le caratteristiche del filtro e l'operatore mette in relazione il nome ed il valore. Siccome è abbastanza contorta la spiegazione suppongo sia più facile spiegarla tramite un esempio; nel caso volessi visualizzare tutti i processi che non rispondono, allora scriverò così:

tasklist /fi "status eq not responding"

dove "status" indica il tipo di filtro da utilizzare (in questo caso lo stato del processo), "not responding" è il Valore (non rispondono) e "eq" è l'operatore. Se avessimo dovuto scrivere lo stesso comando in italiano avremmo scritto: *Mostra tutti i processi il cui stato è "non risponde"*, dove "il cui stato" sarebbe il nome, "non risponde" il valore e "è" l'operatore.

Oltre a "eq" un altro operatore da ricordare è "ne", che sarebbe l'esatto opposto. Infatti, se volessimo scrivere *Mostra tutti i processi il cui stato non è "non risponde"* dovremmo scrivere così:

tasklist /fi "status ne not responding"

Siccome le impostazioni di questo parametro sono troppo numerose per essere esposte una ad una (ci vorrebbe una guida solo per quelle), mi limiterò a fare una tabella:

Nome	Operatore	Valore
Status	eq, ne	running, not responding
Imagename	eq, ne	ogni stringa valida
PID	eq, ne, gt, lt, ge, le	ogni numero PID valido
Session	eq, ne, gt, lt, ge, le	ogni numero di sessione valido
SessionName	eq, ne	ogni stringa valida
CPUTime	eq, ne, gt, lt, ge, le	orario valido nel formato <i>hh:mm:ss</i> ; mm e ss hanno valori compresi tra 0 e 59, hh, invece, non ha limiti
Memusage	eq, ne, gt, lt, ge, le	ogni intero valido
Username	eq, ne	ogni nome utente valido
Services	eq, ne	ogni stringa valida
Windowtitle	eq, ne	ogni stringa valida
Modules	eq, ne	ogni stringa valida

Quando, invece, vogliamo sapere anche quali moduli vengono utilizzati dai singoli processi, usiamo il parametro */m*. Perciò sulla riga di comando andremo a scrivere così:

tasklist /m

Se poi vogliamo vedere quali processi utilizzano un determinato modulo, scriveremo così:

tasklist /m NomeModulo

Faccio un esempio. Mettendo caso che voglia sapere quali processi utilizzano il modulo *ntdll.dll*, scriverò così:

tasklist /m ntdll.dll

ed avrò questo risultato:

```

Microsoft Windows [Versione 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. Tutti i diritti riservati.

C:\Users\>tasklist /m ntdll.dll

Nome immagine          PID Moduli
=====
dwm.exe                 2144 ntdll.dll
taskeng.exe             2220 ntdll.dll
explorer.exe            2240 ntdll.dll
MSASCui.exe             2472 ntdll.dll
RtHDVCpl.exe            2516 ntdll.dll
jused.exe               2660 ntdll.dll
ashDisp.exe             2672 ntdll.dll
Returnil.exe            2784 ntdll.dll
TFTray.exe              2816 ntdll.dll
rundll32.exe            2864 ntdll.dll
wmde.exe                2884 ntdll.dll
UCDDaemon.exe           2936 ntdll.dll
ehtray.exe              2968 ntdll.dll
wmpnscfg.exe            2980 ntdll.dll
ehmsas.exe              3012 ntdll.dll
MagicFormation.exe      3068 ntdll.dll
conime.exe              4720 ntdll.dll
WINWORD.EXE             5140 ntdll.dll

```

Questo parametro non è compatibile però con i parametri `/svc` e `/v` che adesso vedremo.

Passiamo a vedere il parametro `/svc`. Premetto dicendo che è utilizzabile solo quando `/fo` è impostato su `table`, e non può essere usato contemporaneamente con i parametri `/m` e `/v`. Comunque, lo scopo di questo parametro è quello di mostrare nella lista dei processi anche i servizi che ciascun processo ospita. Nel prompt andremo a scrivere così:

```
tasklist /svc
```

Se invece vogliamo avere una descrizione del processo, allora useremo il parametro `/v`. Per utilizzarlo digiteremo così:

```
tasklist /v
```

Questo parametro non può essere utilizzato assieme a `/svc` o `/m`.

L'ultimo parametro che andremo a vedere per questo comando è il parametro di aiuto. Quando non riusciamo a ricordarci qualcosa e vogliamo saperla nello immediato, ci basterà digitare:

```
tasklist /?
```

per avere una visione di insieme del comando.

TASKKILL

Il comando `taskkill` è il comando utilizzato per chiudere i processi in esecuzione. Ha molte cose in comune con il comando `tasklist`, che andremo a vedere.

Per prima cosa, è opportuno vedere come terminare normalmente un singolo processo. Farlo è abbastanza semplice, ci basterà digitare sulla riga di comando:

```
taskkill NomeProcesso
```

Ovviamente il comando, come d'altronde `tasklist`, ha anche dei parametri che possono essere impostati. Vediamoli uno ad uno.

Parametri che conosciamo bene sono `/s`, `/u` e `/p`, già visti in `tasklist`, che funzionano allo stesso modo, quindi ritengo inutile spiegarli. Un parametro che invece conviene rivedere è `/fi`. Anche se non molte,

questo parametro presenta alcune caratteristiche differenti con quelle viste il tasklist. Tuttavia, anche questa volta, non le posso spiegare una ad una perché troppe, quindi, mi limiterò nuovamente a fare una tabella.

Nome	Operatore	Valore
Hostname	eq, ne	Ogni stringa valida
Status	eq, ne	Running, Not Responding
Imagename	eq, ne	Ogni stringa valida
PID	eq, ne, gt, lt, ge, le	Ogni intero positivo valido
Session	eq, ne, gt, lt, ge, le	Ogni numero di sessione valido
CPUTime	eq, ne, gt, lt, ge, le	Ogni tempo valido nel formato hh:mm:ss. mm e ss sono compresi tra 1 e 59, hh può essere qualsiasi numero positivo
Memusage	eq, ne, gt, lt, ge, le	Ogni intero valido
Username	eq, ne	Ogni username valido (anche dominio\username)
Services	eq, ne	Ogni stringa valida
Windowtitle	eq, ne	Ogni stringa valida

Come per tasklist, nel prompt digiteremo così:

```
taskkill /fi "Nome Operatore Valore"
```

A questo punto, andiamo a vedere il parametro */pid*, che nel prompt scriveremo così:

```
taskkill /pid ProcessID
```

Come avrete sicuramente capito, con questo parametro possiamo dire al sistema di chiudere il processo con un determinato PID a nostra scelta.

Se invece vogliamo specificare quale processo chiudere basandoci sul nome del processo, andremo a scrivere così:

```
taskkill /im NomeProcesso
```

Per chiudere tutti i processi (cosa che sconsiglio), bisogna digitare dopo il parametro */im* il carattere * (ovviamente lasciando lo spazio).

Quando però vogliamo chiudere un intero albero processi (molte volte è necessario) allora questi parametri non bastano più, perciò dovremo ricorrere al parametro */t*, che nel prompt si scrive così:

```
taskkill /t NomeProcesso
```

Così facendo, andremo a chiudere il processo assieme a tutti i processi ad esso collegati.

Per fare l'uscita forzata (molto utile su quel crash ambulante che risponde al nome di Windows) ricorremo al parametro */f*. Quindi digiteremo così sulla riga di comando:

```
taskkill /f NomeProcesso
```

Infine, per avere un aiuto immediato sul comando, digiteremo

```
Taskkill /?
```

per avere una panoramica sui parametri del comando.

FINE