

date in depth

mobytrick

19 Settembre 2015

Il comando `date` serve per gestire l'ora e, di riflesso, la data di un pc. Per poter eseguire delle modifiche sono necessari dei privilegi. In questa prima fase viene fatto un breve excursus limitato all'uso più tradizionale: estrazione di ciò che serve e sua visualizzazione. Ma operando in tale modo si fa un torto al comando. Combinando assieme aritmetica e lingua inglese si possono ottenere risultati sorprendenti. Tale modalità non la si coglie appieno nella *man page*. Qualcosa di più la si ottiene usando il comando `info date`. Lo scopo di questo articolo è quello di gettare uno sguardo oltre l'uso semplice.

Dall'orologio di macchina si possono ottenere molte più informazioni che non le tradizionali 6 (giorno, mese, anno, ora, minuto, secondo). Nella gestione dell'orologio è essenziale anche il fuso orario che in informatica è più noto come *time zone*. È individuato, nella maggioranza dei casi, dal nome del continente e dalla capitale dello stato. Viene usato per operare in automatico il passaggio dall'ora solare a quella legale e viceversa. L'importanza è tale che in fase di installazione del sistema operativo (Linux e Windows) è richiesta la sua immissione. Sono oltre una quarantina le informazioni che si possono ottenere dalla macchina. Ma in certi casi ciò che più colpisce, o sorprende, non è tanto cosa si può estrarre dalla macchina, quanto come questo viene presentato. Perché un ruolo importante lo gioca la configurazione dell'ambiente. Un esempio è la rappresentazione della data. In un ambiente internazionale vengono seguite le convenzioni anglosassoni che per la data prevedono la seguente scaletta: mese, giorno, anno. In un ambiente configurato per l'Italia le prime due entità sono scambiate. Pure il nome dei giorni della settimana e dei mesi appaiono tradotti. E non è detto che tutti usino il calendario in vigore nel mondo occidentale né tantomeno l'alfabeto internazionale.

Della quarantina e passa informazioni, spesso interessano solo alcune, al limite una sola. Ciascuna informazione è singolarmente individuata da una lettera. Per estrarre l'ora della macchina tale lettera è la H (si tenga bene a

mente che Linux è *case sensitive*). Il comando per estrarre l'ora è il seguente:

```
date +%H
```

ovvero la lettera che lo individua l'entità che lo individua preceduta dal carattere % (percento), a sua volta preceduto dal carattere + (più). Il comando

```
date +%H%M%S
```

fornisce la terna (ora, minuto, secondo) di informazioni relative all'ora di macchina, ma l'output non è gradevole, in quanto le informazioni appaiono di seguito, senza alcuna distinzione reciproca. Meglio usare il comando

```
date +"%H:%M:%S"
```

L'apice doppio (") permette di inserire altri caratteri, in questo caso il : (doppio punto) quale separatore tra le varie entità. Un esempio un po' più elaborato è:

```
date +"Data odierna: %A, %g %B %Y"
```

ove la lingua usata per il nome del giorno (%A) e del mese (%B) dipende dalla configurazione.

Per l'uso avanzato, accanto alle specifiche del formato, si deve usare, oltre all'eventuale specifica del formato, l'opzione `--date="stringa"`, ove la potenza sta tutta proprio in stringa. Un esempio introduttivo potrebbe essere

```
date --date="next sun"
```

che fornisce la data completa di ora e quant'altro della prossima domenica. Va da sé che `next` può essere sostituito con `last` ed al posto di `sun` possono essere usati: `day`, `month`, `week`, `year`, il nome (breve o completo) di qualsiasi giorno della settimana, con l'eventuale iniziale maiuscola. Non si possono usare però i nomi dei mesi. Qui però si deve usare la terminologia inglese, perché la configurazione non può intrufolarsi all'interno del codice del programma. L'output è lasciato all'immaginazione del lettore.

Una modesta applicazione permette di sapere il giorno della settimana di una ricorrenza, passata o futura che sia. Gli esempi sottostanti mostano il giorno della settimana in cade in natale dell'anno in corso ed il capodanno dello scorso anno.

```
date --date="25 Dec 'date +%Y'" +%A
date --date="1 jan last year" +%A
```

A margine e fuori tema: sia il natale che il capodanno sono feste cosiddette *fisse* perché cadono in un determinato giorno/mese dell'anno; la pasqua invece è *mobile* ma chi conosce Linux sa che può ottenere la sua data tramite il comando `ncal -e` seguito dall'anno (4 cifre).

E se volessi conoscere la data del terzo venerdì successivo alla data odierna? Il comando è:

```
date --date="next Fri + 2 weeks" +"%-g %B %Y"
```

Analizziamolo. `next Fri` calcola la data del prossimo venerdì a cui aggiunge 2 settimane, ottenendo quanto richiesto (**terzo** venerdì). Il trattino che compare in `%-g` fa sì che se il numero del giorno è inferiore a 10, e quindi è di una sola cifra, non subisca il *padding* (premessa di uno 0, che è il comportamento default). Un dubbio filologico: `weeks` oppure `week`? Nessun problema: Linux è un sistema operativo mica un professore di lettere con la matita rossa e blu! Col comando precedente abbiamo introdotto l'aritmetica. E valgono quindi le sue regole. È perfettamente legale un comando del tipo

```
date --date="+ 1 year -3 week"
```

Coloro che volessero verificare le risposte per altra via, possono usare il misconosciuto comando Unix `cal` (per calendario). Se gli si fornisce mese numerico seguito da uno spazio e l'anno (4 cifre), stampa a video il calendario relativo al mese richiesto. Se viene fornito il solo anno viene stampato il calendario annuale.

I seguenti 3 comandi sortiscono lo stesso effetto

```
date --date="last month"
date --date="- 1 mont"
date --date="1 month ago"
```

da cui si può prendere lo spunto per variazioni sul tema. Va da sé che quanto detto relativamente alla data, può essere applicato anche al tempo. Qui le entità si chiamano `hour`, `minute` e `second`.