

\vadjust, questo sconosciuto

Claudio Beccari

Sommario

Il comando `\vadjust` è virtualmente sconosciuto alla maggior parte degli utenti di $\text{\LaTeX}$. Serve per eseguire cose un po' insolite, ma semplici e con poco sforzo. In questo articolo verranno mostrate alcune applicazioni.

Abstract

Command `\vadjust` is virtually unknown to the majority of $\text{\LaTeX}$ users. This primitive command allows to do some actions that are unusual, but at the same time rather simple and with little effort. This article describes some applications.

1 Introduzione

Il comando `\vadjust` è nativo di $\text{\TeX}$ -the-program; quindi è disponibile in tutti i motori di composizione del sistema $\text{\TeX}$, dato che devono essere compatibili con il capostipite ideato da Donald E. Knuth.

Esso entra nelle definizioni di molti comandi sia di Plain $\text{\TeX}$, sia di $\text{\LaTeX}$; ma gli utenti di questi mark-up difficilmente ne conoscono l'esistenza. $\text{\LaTeX}$ è stato creato con lo scopo di funzionare da interfaccia fra l'utente e l'interprete primitivo del mark-up, pertanto separa l'utente dalla difficoltà di usare un linguaggio di programmazione ricco di comandi (circa 300 nativi di $\text{\TeX}$) non molto semplici da usare. Persino Knuth, nello scrivere i suoi libri, ha sviluppato pacchetti di macro proprio per rendere più sicure molte azioni ripetitive. Gli utenti di $\text{\LaTeX}$ raramente hanno aperto il file `latex.ltx` per vedere che cosa contiene; se lo facessero, rimarrebbero impressionati dal numero di macro che vi vengono definite e dalla moltitudine di comandi nativi che vi vengono usati.

La disponibilità di molti *moduli* di espansione¹ rende ancora più potente l'uso di $\text{\LaTeX}$, ma difficilmente l'utente si rende conto di quello che c'è dietro questa architettura modulare. È giusto che sia così, ma questo non toglie che qualche volta anche un normale utente non possa desiderare di crearsi qualche semplice macro per svolgere compiti che forse sono già svolti da moduli esistenti (quasi 3000), eppure difficili da trovare anche qua-

lora si disponesse di una installazione completa del sistema $\text{\TeX}$.

Ecco allora che sapere qualcosa sui comandi nativi di $\text{\TeX}$ può essere utile a chiunque, specialmente per fare cose semplici anche se insolite.

2 Cosa fa \vadjust

Il comando `\vadjust` richiede un argomento che deve essere costituito da comandi che si possono eseguire in modo verticale. Questi comandi verranno eseguiti appena l'impaginatore, che fa parte del motore di composizione, passa in modo verticale. Il *modo verticale*, in parole povere, è quella modalità di funzionamento del motore di composizione nella quale esso dispone degli oggetti uno sopra l'altro, tipicamente quando il testo di un capoverso viene spezzato in righe che vengono collocate l'una sull'altra a formare una lista verticale. Il motore di composizione si trova in modo verticale anche fra un capoverso e l'altro, fra un capoverso e la matematica in display, eccetera. In questa modalità il motore di composizione può inserire fra un oggetto e l'altro degli spazi (verticali), delle penalità e altri oggetti come il contenuto dell'argomento di `\vadjust`. Però il comando `\vadjust` deve essere usato solo in modo orizzontale, cioè quando il motore di composizione sta raccogliendo il testo prima di scomporlo in una lista di righe giustaposte verticalmente. Per questo genere di dettagli si può andare a fondo consultando sia il testo di KNUTH (1984), sia quello di ELJKHOUT (2014).

La particolarità dell'argomento di `\vadjust` è che esso viene sviluppato ed eseguito solo quando viene trasferito nella lista verticale, non quando il motore di composizione trova il comando `\vadjust` che lo contiene; quindi l'esecuzione di quell'argomento viene differita al momento in cui esso viene trasferito alla lista verticale.

Il discorso potrebbe apparire astratto, ma gli esempi che seguono ne mostrano il funzionamento e i capoversi precedenti diventano più chiari.

3 Spezzare una colonna di testo

Quanto viene descritto qui di seguito vale anche quando si sta componendo in modalità `onecolumn`, ovvero la modalità di composizione predefinita di quasi tutte le classi, o quando si sia specificato il comando `\onecolumn` in un testo altrimenti composto a due colonne. Ma trovo che quanto descritto qui di seguito sia più interessante quando si compone a due colonne senza ricorrere al pacchetto

1. Generalmente chiamati *pacchetti*, perché solitamente non sono composti da una sola file con estensione `.sty`, ma i vari file componenti sono impacchettati in un unico archivio compresso, la cui installazione nelle distribuzioni del sistema $\text{\TeX}$ viene gestita da appositi programmi.

multicol, che certamente è molto utile, ma ha anche alcune sue limitazioni; in particolare non consente di inserire oggetti flottanti in colonna, ma solo fuori delle colonne e a piena pagina. Tuttavia, uno dei suoi pregi è quello di consentire il bilanciamento delle colonne nell'ultima parte composta dentro il suo ambiente *multicols*.

Componendo invece a due colonne, o perché si è specificata l'opzione *twocolumn* alla classe, o perché si è dato il comando `\twocolumn` esplicitamente o implicitamente attraverso altri comandi o ambienti², le colonne non vengono pareggiate nell'ultima pagina. Esiste il modulo *balance*, che però modifica la routine di output di L^AT_EX e per questo può dare problemi non indifferenti, specialmente se le colonne da bilanciare contengono note e oggetti flottanti. Si può ottenere lo stesso risultato in modo meno automatico, ma molto più semplice, usando `\vadjust`.

Infatti basta definire il comando `\pareggia` in questo modo:

```
\newcommand*\pareggia{\vadjust{\newpage}}
```

e intercalarlo fra due parole della riga dopo la quale si vuole spezzare la prima colonna. Sembra troppo semplice e lo è, ma non è automatico.

Bisogna procedere così: si compone il testo senza usare quel comando; si vede quanto sia più corta la seconda colonna dell'ultima pagina rispetto alla prima colonna; si contano le righe di differenza e si divide il loro numero per due; si cerca la riga che sia quel numero di righe prima del fondo della prima colonna; si inserisce quel comando e probabilmente il risultato che si ottiene compilando di nuovo è proprio quello che si desiderava; può succedere di avere mal contato le righe o che il comando vada inserito nella riga precedente, ma in sostanza il gioco è fatto.

Nel comporre a due colonne, la routine di uscita compone ciascuna colonna come se fosse una pagina, solo che nella pagina fisica dispone le due pagine/colonne una accanto all'altra; una volta emessa la colonna di sinistra, si dimentica di quello che conteneva e procede ad elaborare la colonna di destra.

Ecco allora che mentre predispone la colonna di sinistra, ne incolonna le scatole che contengono ogni riga, e quando arriva alla riga che contiene il comando `\vadjust` ne estrae il contenuto per inserirlo subito dopo la riga che lo conteneva; nel nostro caso la definizione di `\newpage` è la seguente:

```
\def \newpage {%
  \ifnoskipsec
    \ifx \@nodocument\relax
      \leavevmode
    \global \@noskipsecfalse
```

2. Per esempio, l'indice analitico viene composto a due colonne anche nei documenti composti ad una colonna per impostazione predefinita.

```
\fi
\fi
\if@inlabel
  \leavevmode
  \global \@inlabelfalse
\fi
\if@nobreak\@nobreakfalse\everypar{}\fi
\par
\vfil
\penalty -\@M}
```

Di questa definizione ci interessano le ultime righe: `\par` termina il capoverso, nel caso si stesse lavorando in modo orizzontale, e questo è il motivo per il quale non si può usare `\newpage` direttamente; ma quando `\newpage` viene estratto siamo già in modo verticale, quindi il comando `\par` è del tutto inerte (non operativo); `\vfil` riempie il fondo della colonna di spazio verticale infinitamente allungabile, così da evitare l'allargamento degli spazi fra ogni capoverso insieme ai messaggi di colonna/pagina (`\vbox`) mal riempita; infine inserisce una penalità negativa che induce la routine di uscita a spezzare la colonna proprio in quel punto. La routine di uscita si dimentica della colonna appena composta e comincia a elaborare la seconda colonna, aggiungendovi subito le righe del capoverso appena interrotto e non ancora emesse, per accodarvi poi le righe dei capoversi successivi. Se la posizione del comando `\vadjust` con il suo argomento è stata scelta in modo corretto, le due colonne della pagina appariranno bilanciate, o al massimo la seconda colonna sarà una riga più corta della colonna di sinistra.

Come ho spiegato sopra, le righe importanti sono le ultime due; tanto varrebbe allora che l'utente definisse semplicemente

```
\newcommand*\pareggia
{\vadjust{\vfill\penalty -10000}}
```

e si sarebbe ottenuto lo stesso risultato, visto che nel caso specifico `\vadjust` manda in esecuzione i comandi del suo contenuto solo quando si è già in modo verticale.

4 Come marcare la riga che contiene un voce riportata nell'indice analitico

È esperienza comune che se si consulta un indice analitico di un testo stampato, si trova facilmente la pagina o l'elenco delle pagine del testo dove la voce indicizzata è usata in modo non casuale. Poi si va alla pagina indicata o, in successione, alle pagine indicate, ma spesso bisogna leggere l'intera pagina prima di trovare la parola che interessa. Se la voce che è stata inviata all'indice non è evidenziata, in qualche modo la lettura dell'intera pagina fa perdere non poco tempo. Ma se la riga che contiene almeno una parola indicizzata è marcata a margine

con un segno qualsiasi, l'operazione diventa più semplice; anche se quella pagina contenesse diverse parole indicizzate, esse sarebbero certamente in numero molto minore delle righe presenti, quindi basta leggere solo le poche righe che hanno il marchio nel margine destro.

Si potrebbe usare una nota marginale; peccato che se le note marginali sono molte e il testo contiene un buon numero di oggetti flottanti, la presenza di molte note marginali può far perdere l'inserimento di alcuni essi.

Mi è capitato di vedere un testo con moltissime figure raccolte in una apposita appendice in fondo al documento e qualche figura mantenuta nel testo; ognuna di queste figure aveva la sua didascalia numerata, ma le figure in appendice venivano segnalate con il loro numero nel margine esterno collocato col meccanismo delle note marginali; le figure nel capitolo erano in numero decisamente limitato e non c'era, quindi, nessun motivo per perderne per la strada³. Ma collocare nel file di uscita degli oggetti flottanti implica fare i conti anche con le note marginali, anch'esse mobili. Di conseguenza la persona che stava componendo quel testo non riusciva ad uscirne; le ho cambiato tutte le note marginali con un comando che faceva uso di `\vadjust` e il problema è sparito.

In entrambi i casi conviene definire una macro. Nel caso delle voci da indicizzare, basta definire un nuovo comando, diciamo `\Index` in questa maniera:

```
\newcommand\Index[1]{#1\index{#1}%
\vadjust{\hbox to \linewidth{\hfill
\rlap{\enspace\smash{\raisebox
{0.3ex}{\rule{1ex}{1ex}}}}}}}
```

e ogni riga che contiene una voce indicizzata appare marcata con un rettangolino nero, come accanto a questa riga. Qualche commento non è fuori luogo. ■

1. `\hbox to\linewidth{\hfill...` è del tutto equivalente a `\makebox[\linewidth][r]{...}` tranne per il fatto che il comando `\hbox` non modifica il modo di composizione, mentre `\makebox` passa al modo orizzontale.
2. Si è usato `\linewidth` che rappresenta sempre la lunghezza corrente della riga di testo, sia essa ad una colonna o a due colonne, sia essa con margini rientrati o non rientrati rispetto ai margini normali; in talune circostanze potrebbe non essere la scelta giusta, ma va bene quasi sempre.
3. `\rlap{...}` è quasi equivalente al comando `\LATEX\makebox[0pt][l]{...}`; anche `\rlap` non passa in modo testo, ma qui non è importante perché è contenuto dentro una `\hbox`; lo

3. Le code degli oggetti flottanti arrivano a contenerne al massimo 18; se si usa il pacchetto *morefloats* si può arrivare fino a 36 oggetti in coda.

si è usato semplicemente perché è più corto da scrivere rispetto al corrispondente comando `\LATEX`.

4. Il rettangolino nero composto con il comando `\LATEX\rule` ha il suo punto di riferimento alla base della scatola che lo contiene, quindi risulterebbe più in basso del desiderato. Ecco perché si usa `\raisebox`: l'ammontare del rialzo è confrontabile con la profondità dei discendenti della riga dopo la quale si trova il contenuto di `\vadjust`; si potrebbe rendere più precisa questa operazione usando `\prevdepth`, che rappresenta la profondità della riga precedente (KNUTH, 1984; EIJKHOUT, 2014).
5. Non a caso si è usata l'unità di misura `ex`, cosicché tutta la costruzione può funzionare correttamente accanto a ogni riga composta con font di qualunque corpo.
6. Un altro comando usato è `\smash`. Esso è definito anche quando si usa `\LATEX`, ma è poco conosciuto: serve per azzerare l'altezza e la profondità della scatola contenente l'argomento di `\smash`. In questo modo l'altezza e la profondità della prima scatola `\hbox` risultano entrambe nulle e il contenuto di `\vadjust` non altera la spaziatura fra le righe del testo circostante.
7. Il secondo problema, quello di mettere i riferimenti delle figure in appendice nel margine destro, si risolve nello stesso modo sostituendo `\rule` e il suoi argomenti con `\ref` e l'etichetta della figura a cui si vuol fare riferimento.
8. Componendo a due colonne sarebbe desiderabile che `\vadjust` avesse argomenti diversi per la colonna di sinistra e quella di destra; si può fare, naturalmente, ma lo si lascia per esercizio al lettore.

5 L'equivalente di `\showidx`

Il modulo *showidx* serve durante la lavorazione di un documento per marcare nel margine il contenuto del comando `\index`; alcuni autori lo considerano molto comodo. Per ottenere questo risultato il modulo ridefinisce alcuni comandi interni di `\LATEX` al fine di usare una nuova categoria di "insert", cioè di oggetti da inserire nel documento finale in modo flessibile, come di fatto viene fatto anche per le note al piede e le note marginali.

Se si vuole evitare di caricare pacchetti che modifichino il cuore di `\LATEX` si può ricorrere a `\vadjust`. Definendo un nuovo comando `\indexshow` si può ottenere il medesimo risultato che si ottiene con il modulo *showidx* senza compromettere il cuore di `\LATEX`.

Il comando potrebbe essere del tipo:

```
\newcommand*\indexshow[1]{\index{#1}%
  \vadjust{\hbox to \linewidth{\hfill
    {\rlap{\enspace
      \smash{\raisebox{0.3ex}{#1}}}}}}}
```

Anche in questo caso, componendo a due colonne si potrebbe mettere il contenuto di `\indexshow` a sinistra della prima colonna e a destra della seconda colonna, ma il trucco è sempre lo stesso che ho lasciato prima al lettore.

È bene ricordare che questo comando potrebbe essere migliorato se l'argomento di `\smash` fosse composto dentro una scatola verticale, cioè, in termini del mark-up di LATEX, mediante una `\parbox`, perché talvolta le stringhe passate come argomento a `\index` possono essere un po' lunghe; così facendo, potrebbero ripiegarsi nella giustezza del margine.

6 `\vadjust` e `\enlargethispage`

Componendo a due colonne può succedere che la prima colonna dell'ultima pagina contenga una nota che va al piede della colonna, anche se si usa il comando `\pareggia` definito sopra.

Questo caso ha una soluzione semplice se si usano in tandem il comando `\pareggia`, definito sopra, con `\vadjust` e `\enlargethispage`⁴, oltre ai comandi `\footnotemark` e `\footnotetext`. Questi ultimi comandi servono per gestire il richiamo della nota separatamente dal testo, cosicché sia possibile mettere il testo della nota in posti diversi da dove li metterebbe il semplice comando `\footnote`.

In sostanza bisogna usare `\footnotemark` nell'esatto posto in cui si vuole mettere il richiamo delle nota e inserire il comando `\footnotetext` con il testo della nota alla fine del testo della seconda colonna, cosicché la nota vada a finire al piede della seconda colonna.

Ciò fatto, si compone il documento e poi si contano quante righe della prima colonna non sono affiancate da righe della seconda colonna; si divide per due questo numero e, a partire dalla riga precedente il piede della prima colonna, si inserisce il comando `\pareggia`. Compilando nuovamente il

documento si vede la prima colonna accorciata, ma la nota resta sempre al piede della seconda colonna. Alla fine del testo sorgente della seconda colonna si inserisce poi il codice per diminuirne l'altezza:

```
\enlargethispage{-<n>\baselineskip}
```

dove $\langle n \rangle$ è pressappoco il numero di righe di cui si è accorciata la prima colonna.

Questo valore $\langle n \rangle$ va regolato finemente con qualche compilazione successiva, ma non è difficile trovare qualche decimale che pareggi esattamente le due colonne. A me di solito riesce dopo due o tre ricompilazioni. Come si vede l'ho fatto anche per pareggiare le colonne di questa pagina.

7 Conclusioni

Credo di avere mostrato con esempi significativi l'utilità del comando `\vadjust` nativo di TEX; i semplici esempi che ho mostrato dovrebbero a mio parere essere sufficienti, ma l'appetito viene mangiando, come si suol dire, e i lettori possono giovare per trovare ulteriori soluzioni per situazioni più complesse.

Riferimenti bibliografici

EIJKHOUT, V. (2014). *TEX by topic*. Dante e.V and Lehmanns Media GmbH, Heidelberg and Berlin.

KNUTH, D. E. (1984). *The TEXbook*. Addison Wesley Publ. Co., Reading, Mass.

▷ Claudio Beccari
claudio dot beccari at gmail
dot com

4. Si ricorda che il comando `\enlargethispage` componendo a due colonne è equivalente ad un ipotetico comando `\enlargethiscolumn`, che non esiste. La variante asteriscata in questo caso non deve essere usata, perché sfrutterebbe lo spazio elastico presente nella colonna per cambiarne la lunghezza il meno possibile.