

Uso del comando `\write18` per comporre l'indice analitico in modo sincrono

Claudio Beccari*

Sommario

Si mostra uno dei possibili usi del comando `\write18` per rendere l'indice analitico assolutamente sincrono con il testo del documento a cui esso si riferisce, evitando di svolgere le tre o quattro compilazioni asincrone di *makeindex* e *latex* oppure *pdflatex* abitualmente necessarie quando non è disponibile il comando `\write18`.

Abstract

One of the possible uses of the `\write18` command is described in order to compose the index in an absolutely synchronous way relative to the document it refers to. The three or four asynchronous runs of *makeindex* and *latex* or *pdflatex* are therefore avoided, as it is not the case when the command `\write18` is not available.

1 Introduzione

Chiunque abbia composto un indice analitico con \LaTeX conosce benissimo l'inconveniente che si manifesta quando, dopo aver pazientemente marcato tutti i lemmi dell'indice, si dimentica di eseguire l'ultima esecuzione multipla dei programmi necessari alla composizione dell'indice aggiornato alla fine del documento. Normalmente bisogna comporre e ricomporre il documento finché le bozze non mostrano più la necessità di nessuna correzione; successivamente bisogna:

1. lanciare *latex* o *pdflatex* per la compilazione definitiva durante la quale vengono raccolti i lemmi per l'indice, le informazioni per la loro formattazione, il numero della pagina nella quale essi cadono; queste informazioni vengono scritte in un file con l'estensione `.idx` esattamente nell'ordine in cui sono state raccolte;
2. lanciare *makeindex*, eventualmente specificando un file di stile per la composizione dell'indice analitico, in modo da produrre la lista alfabetica dei lemmi, per i quali è specificato il loro "rango" e le pagine dove sono descritti; in realtà queste informazioni vengono scritte in un file con l'estensione `.ind` che contiene tutto

*Quest'articolo è stato tradotto e presentato per la pubblicazione su TUGboat. La versione inglese è ripubblicata per gentile concessione dell'autore e di TUGboat a pagina 111.

l'ambiente `theindex` entro il quale ogni lemma, a seconda del suo "rango", è introdotto con i comandi `\item`, oppure `\subitem` oppure ancora `\subsubitem` insieme alle informazioni per la sua composizione tipografica;

3. lanciare ancora *latex* o *pdflatex* che rielabora l'intero documento, comprensivo dell'indice analitico, introdotto nel file sorgente mediante un comando del tipo `\input{filename}.ind` o altri comandi equivalenti forniti da altri pacchetti di estensione o da file di classe.
4. Potrebbe essere necessario lanciare ancora *latex* o *pdflatex* se la classe del documento o le opzioni personali richiedono che l'indice analitico sia indicato anche nell'indice generale.

Ebbene, questi tre o quattro passi possono essere compiuti in un'unica esecuzione di *latex* o *pdflatex* se si usa il comando `\write18{text}`.

Infatti le distribuzioni moderne del sistema \TeX contengono un file eseguibile e file di formato compilati con opportune opzioni cosicché l'interprete *pdfetex* possa produrre un file di uscita in formato `.dvi` oppure `.pdf`; può usare gli ulteriori comandi dell'interprete esteso *etex* oppure ne può essere interdetto; può eseguire un programma esterno (o può esserne impedito) che consenta al sistema di eseguire un compito particolare per poi, esaurito questo compito, riprendere a lavorare normalmente per continuare la composizione tipografica.

Qui si userà quest'ultima capacità in modo da produrre l'indice analitico effettivo, eseguendo una sola volta il compito dell'interprete dei comandi \LaTeX .

2 I file ausiliari

Quando il comando `\makeindex` viene eseguito durante l'elaborazione del preambolo del documento, viene definito e aperto un nuovo flusso di uscita al fine di scrivere le informazioni per l'indice analitico in un file ausiliario `.idx`; nello stesso tempo il comando rende attivi i vari comandi `\index` sparsi lungo il documento i quali sono i responsabili della scrittura dell'informazione suddetta nel file ausiliario. Il nome di questo flusso di uscita vie-

ne conservato dentro la sequenza di controllo del nucleo di \LaTeX `\@indexfile`¹.

Questo file ausiliario ha l'estensione `.idx`, ma il suo nome è generalmente lo stesso del file principale `.tex` che genera il documento composto. Il nome di questo file principale è quello del "job" e viene conservato nella sequenza di controllo `\jobname`, che può venire usata da qualunque altra macro che faccia al caso.

Poiché l'indice analitico è verosimilmente l'ultima cosa che viene composta nel documento, il comando `\input{\jobname.ind}` è presumibilmente l'ultimo prima della specificazione `\end{document}`.

Perciò al fine di produrre il file `\jobname.ind` a partire da `\jobname.idx`, subito prima di introdurre con `\input{\jobname.ind}` il file dell'indice analitico alfabetizzato, bisogna eseguire due semplici operazioni:

1. Chiudere il flusso di uscita identificato dalla sequenza `\@indexfile`, ed
2. eseguire il programma *makeindex* al fine di produrre la nuova versione aggiornata del file `.ind`.

3 I comandi

Bisogna dunque eseguire *immediatamente* i due comandi seguenti:

```
\immediate\closeout\@indexfile
\immediate\write18{makeindex
-s style.ist \jobname.idx}
```

prima del comando `\input{\jobname.ind}` (o di ogni altra espressione equivalente, come per esempio `\printindex`). Ovviamente se non si usa nessuno stile particolare, ma si compone l'indice analitico secondo le impostazioni di default, allora bisogna omettere l'intera locuzione `-s style.ist`.

Il comando primitivo `\immediate` ordina all'interprete di eseguire immediatamente il comando che segue senza dilazionarlo. Infatti i comandi \TeX connessi con i flussi di entrata e, specialmente, di uscita, inclusi i comandi per 'leggere da' oppure 'scrivere su' file esterni vengono solitamente dilazionati al momento in cui l'interprete usa la sua routine di uscita, a meno che questi comandi di "input/output" non siano preceduti dal comando `\immediate`.

1. Questo è quello che il nucleo di \LaTeX e le classi standard fanno. Alcune classi non standard possono usare altre sequenze di controllo per conservare l'informazione relativa al flusso di uscita per l'indice analitico: per esempio, la classe *memoir* usa un nome formato agglutinando il nome del "job" `\jobname` con il suffisso `@idxfile`, ma, siccome *memoir* può costruire diversi indici analitici, ognuno di questi ha un suo flusso di uscita con un nome diverso, il nome che viene specificato come argomento facoltativo del comando `\makeindex`.

Nei due comandi precedenti `\closeout` chiude il flusso di uscita, nel nostro caso quello specificato mediante `\@indexfile`.

Invece il comando `\write18{<text>}` scrive nel flusso di uscita speciale numerato "18"; questo flusso di uscita equivale a scrivere `<text>` in una riga di comando nella finestra comandi della famiglia dei sistemi operativi Windows; oppure nella finestra `xterm` dei sistemi operativi Linux; oppure nella finestra Terminal dei sistemi operativi Mac OS X. Quando l'applicazione esterna ha terminato di eseguire il suo processo di elaborazione, il programma *latex* o *pdflatex* riprende il controllo e prosegue dal punto esatto dove si era fermato per lanciare il programma esterno.

La presenza del segno `@` nel nome del flusso di uscita `\@indexfile` richiede che si scriva il comando `\makeatletter` prima del primo comando e si scriva `\makeatother` subito dopo. Ma se quei due comandi vengono assegnati come sviluppo di una macro inserita in un file di macro personali `.sty` non c'è alcun bisogno di cambiare codice di categoria al segno `@`. Definire una macro personale con i due comandi suddetti è anche una buona occasione per aggiungere un po' di "fronzoli" utili: per esempio, per verificare che il flusso di uscita sia davvero aperto e possa quindi venire chiuso; per verificare che il file `\jobname.idx` esista per davvero al fine di elaborarlo; il secondo comando potrebbe avere un argomento facoltativo per consentire di specificare un eventuale file di stile `.ist` così da inserire o da omettere la locuzione concernente il file di stile nella chiamata del programma esterno; si possono aggiungere avvisi di cautela o messaggi d'errore per informare il compositore se c'è qualcosa di inconsistente. Non è qui il caso di insistere oltre; in fondo ogni utente del linguaggio \LaTeX dovrebbe essere in grado di personalizzare quei comandi secondo le sue necessità.

4 I risultati

Dando quei comandi (o il nuovo comando definito nel proprio file `.sty` personale) subito prima di emettere l'ordine `\input{\jobname.ind}` l'indice analitico viene a contenere esattamente i lemmi e i numeri di pagina raccolti durante l'unica esecuzione di *latex* o *pdflatex*; cade la necessità di ricordarsi di eseguire le ultime tre o quattro esecuzioni ricordate nell'Introduzione, o di aggiornare l'indice dopo ogni correzione dei file sorgente, perché i lemmi e i numeri di pagina sono sempre aggiornati per costruzione.

Ritengo che questi semplici comandi siano utilissimi, e io stesso li uso sempre da quando ho "scoperto" questa possibilità.

Bisogna per altro segnalare che questa possibilità di far sì che *latex* o *pdflatex* lanci un programma esterno viene per così dire passata un po' sotto silenzio, perché sarebbe possibile far eseguire al si-

stema operativo programmi che potrebbero esporre il compositore al pericolo di eseguire inconsapevolmente del codice maligno; per questo motivo, addirittura, alcune distribuzioni del sistema TEX non consentono di default di eseguire il comando `\write18`.

Ci sono però buone notizie in arrivo: la prossima distribuzione di TEXlive 2009 conterrà un versione vincolata di `\write18`, nel senso che in ogni “job” sarà possibile usare questo comando “pericoloso”, ma solo per lanciare alcuni programmi specificati nel file di configurazione `texmf.cfg` e, fra questi programmi leciti, c’è anche **`makeindex`**.

Sono certo che esistono molte altre applicazioni di `\write18`; per esempio, sono al corrente che la trasformazione sincrona del formato grafico dei file è già disponibile nel pacchetto `epstopdf.sty`; tocca agli utenti trovarne delle altre!

▷ Claudio Beccari
Villarbasse (TO)
`claudio dot beccari at gmail
dot com`