

Laboratorio Multimediale

Lezione n. 4

Corso di Laurea in Matematica, a.a. 2006-2007

3 novembre 2005

Il T_EX

Il T_EX è un'applicazione che permette di formattare testo, formule e figure per comporre documenti per la stampa (*typesetting*). In particolare il T_EX è particolarmente adatto alla creazione di testi scientifici, in quanto permette la composizione di formule matematiche in maniera piuttosto agevole.

Il T_EX può essere inteso come un vero e proprio linguaggio di programmazione specializzato nel *typesetting*. Il procedimento di utilizzo del T_EX prevede la stesura di un documento sorgente, chiamato *file T_EX* che contiene il testo da comporre con eventuali comandi che specificano in che modo il testo va rappresentato. Una volta composto il documento sorgente (ad esempio il file `prova.tex`) bisogna invocare il comando `tex prova.tex` che interpreta il file sorgente e crea i files `prova.aux`, `prova.log` e `prova.dvi`. Il file `prova.aux` è un file di servizio utilizzato dal T_EX, mentre il file `prova.log` contiene messaggi e informazioni sull'esecuzione della composizione. Il file più importante, per l'utente, è invece il file `prova.dvi`. Questo file contiene una rappresentazione del documento, così come dev'essere stampato su carta (o visualizzato su schermo). Il termine `dvi` sta per *device independent*, e significa che questo formato è indipendente dal dispositivo che poi sarà utilizzato per la stampa o per la visualizzazione del documento.

Il file `prova.dvi` può essere visualizzato su schermo con il comando `xdvi prova.dvi &` ma ancora non può essere stampato direttamente. Nei sistemi tipo UNIX, il formato universalmente utilizzato per la stampa è il formato *PostScript* (PS). Questo formato, originariamente sviluppato dalla ditta *Adobe*, è un formato di grafica vettoriale utilizzato da moltissime stampanti ad alta risoluzione (in particolare stampanti laser). Se la stampante collegata al sistema UNIX non è di tipo PostScript, in ogni caso il sistema operativo è configurato in modo da permettere la stampa di file PostScript.

Il file `prova.dvi` può essere convertito in PostScript, mediante il comando `dvips prova.dvi -o prova.ps`. Il file `prova.ps` può a sua volta essere visualizzato su schermo, tramite il comando `gv prova.ps &` oppure può essere stampato tramite il comando `lpr -Pnomestampante prova.ps` dove *nomestampante* è un nome che identifica univocamente la stampante da utilizzare. **N.B.** Nel nostro laboratorio il nome della stampante è: `apple` (la stampante si trova nello scantinato) e il comando di stampa NON è `lpr` bensì `Lpr`.

Un altro formato molto diffuso è il formato PDF (*Portable Document Format*) sviluppato sempre dalla *Adobe*. Questo formato, per certi versi simile al PS, è particolarmente adatto allo scambio di documenti e quindi alla pubblicazione di documenti su internet. Infatti è possibile visualizzare e stampare questi documenti, mediante il programma *Adobe Acrobat Reader*, disponibile gratuitamente su qualunque sistema operativo.

Una versione modificata del T_EX permette di generare direttamente un file PDF invece che DVI. Ad esempio con il comando `pdftex prova.tex` si genera il file `prova.pdf`. Il file `prova.pdf` può essere visualizzato con il comando `xpdf prova.pdf &` (oltre che con `acroread prova.pdf &`).

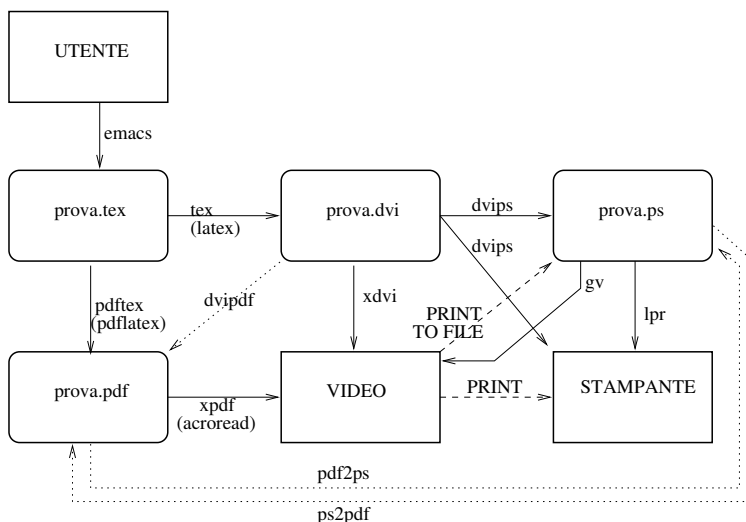


Figura 1: Le fasi di lavorazione di un file $\text{T}_{\text{E}}\text{X}$. Le frecce tratteggiate indicano comandi dell'interfaccia grafica. Le frecce puntinate indicano comandi che possono compromettere la qualità del documento.

Comandi di base del $\text{T}_{\text{E}}\text{X}$

Un file $\text{T}_{\text{E}}\text{X}$, è un normale file di testo in cui alcuni caratteri (tra i quali: \backslash , $\&$, $\$$, $\{$, $\}$) vengono interpretati in maniera particolare. Più importante di tutti è il carattere \backslash che è sempre seguito dal nome di un comando da eseguire. Ad esempio il comando $\backslash\text{TeX}$ produce la scritta $\text{T}_{\text{E}}\text{X}$ all'interno del documento. Altri comandi utili sono: \backslash' e \backslash' che mettono un accento (grave o acuto) sul carattere successivo, ad esempio si può scrivere: `perch\`e Maril\`u \`e andata in Per\`u` per ottenere "perché Marilù è andata in Perù". Per produrre il carattere "ì" bisogna invece utilizzare il comando $\backslash\text{i}\{ \}$ che produce una "ì" senza il punto. Ad esempio: `cos\`{\i}\`e se vi pare` produce il testo "così è se vi pare". Il comando $\backslash\text{end}$ va posto alla fine del file per indicare, appunto, la fine del documento.

I caratteri $\$ \dots \$$ racchiudono una parte che va interpretata come formula. Ad esempio il codice `\$c=a+b\$` la somma dei numeri a e b produce il testo "sia $c = a + b$ la somma dei numeri a e b ". Notiamo che bisogna racchiudere tra $\$ \dots \$$ tutte le formule matematiche, anche se composte da una sola variabile, in quanto il $\text{T}_{\text{E}}\text{X}$ le formatta diversamente dal testo ordinario. La sequenza $\$ \dots \$$ racchiude invece una formula *displayed* che viene centrata su una riga a sè stante. Ad esempio il codice `\$c=a+b\$` produce il risultato:

$$c = a + b$$

I caratteri $\{ \}$ servono a raggruppare il testo (per esempio quando è necessario identificare una zona di testo a cui si applica un comando) e non vengono visualizzati nel documento. Come esempio utilizziamo i comandi \wedge e $_$ che servono per scrivere esponenti e indici nelle formule. Con le parentesi graffe possiamo raggruppare i simboli: `\$x_k^{\{y+z\}}=x_k^y x_k^z\$` produce la formula $x_k^{y+z} = x_k^y x_k^z$.

Molti altri comandi servono per rappresentare simboli e operatori matematici. Ad esempio, il codice `\$ \lim_{x \to 0^+} \sum_{k=0}^{\infty} \frac{x^k}{\sin x} \$` produce la formula

$$\lim_{x \rightarrow 0^+} \sum_{k=0}^{\infty} \frac{x^k}{\sin x}$$

Ogni lettera greca ha un corrispondente comando: $\backslash\alpha$, $\backslash\beta$, $\backslash\gamma$, $\backslash\delta$, $\backslash\epsilon$, $\backslash\epsilon$. Simboli utilizzati nelle espressioni logiche: $\backslash\text{forall}$ $\backslash\text{exists}$ $\backslash\text{to}$ $\backslash\text{Rrightarrow}$ $\backslash\text{Lrightarrow}$ $\backslash\text{Leftrightarrow}$ produce $\forall \exists \rightarrow \Rightarrow \Leftrightarrow$.

Ogni funzione o operatore matematico ha il suo comando: $\backslash\sin \backslash\cos \backslash\log \backslash\lim \backslash\sum \backslash\int$ genera: $\sin \cos \log \lim \sum \int$. Altri comandi: $\backslash\sqrt{x+y} \backslash\over z + \backslash\sqrt[3]{x\over y}$.
genera la formula

$$\frac{\sqrt{x+y}}{z} + \sqrt[3]{\frac{x}{y}}$$

Un esempio completo

Ecco un file completo:

```
Questo \e un esempio di un file \TeX{} minimale. Osserva
come
gli          spazi          nel file
sorgente vengono ignorati.
\E molto facile fare le lettere accentate. Cos\i{} si pu\o
accentare anche la lettera ‘i’.
```

Per iniziare un nuovo paragrafo lascia una riga bianca.

Una semplice formula: $f(x)=x^{\alpha-2}+\sqrt{x+y}$. Una formula centrata:

```
$$$
f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!} (x-x_0)^k.
$$$
\end
```

che genera il seguente testo:

Questo è un esempio di un file $\text{T}_{\text{E}}\text{X}$ minimale. Osserva come gli spazi nel file sorgente vengono ignorati. È molto facile fare le lettere accentate. Così si può accentare anche la lettera “i”. Per iniziare un nuovo paragrafo lascia una riga bianca.

Una semplice formula: $f(x) = x^{\alpha-2} + \sqrt{x+y}$. Una formula centrata:

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k.$$

Compito odierno

1. Creare una directory chiamata `lezione4` in cui metterete tutti i files generati oggi. In questa directory mettere innanzitutto una copia del file `testo.txt` creato nella lezione 3. Convertire il file `testo.txt` in un file $\text{T}_{\text{E}}\text{X}$ `testo.tex` facendo le opportune modifiche in modo tale che le lettere accentate e i cambi di paragrafo vengano interpretate correttamente dal $\text{T}_{\text{E}}\text{X}$ (ricordarsi anche di mettere il comando `\end` alla fine del file). Generare i file `testo.dvi`, `testo.ps` e `testo.pdf` e visualizzarli su schermo.
2. Provare ad inserire nel file un comando sconosciuto (ad esempio `\crash`) per verificare come reagisce il $\text{T}_{\text{E}}\text{X}$ quando interpreta il file.
3. Sempre nel file `testo.tex`, aggiungere la definizione di limite, la definizione di spazio vettoriale e una teorema a piacere completo di dimostrazione. Generare nuovamente i files `testo.ps` e `testo.pdf`.
4. Dalla pagina delle presenze ogni gruppo deve inviare i files: `testo.tex`, `testo.ps` e `testo.pdf`
5. Fare in modo che tutti i componenti del gruppo abbiano una copia del file `testo.tex` per le prossime lezioni.