

Introduzione a *roff

| | | |
|---------|--|------|
| 25.1 | Logica di funzionamento ed esempio di partenza | 1029 |
| 25.1.1 | Avvio degli eseguibili «troff» e «nroff» | 1030 |
| 25.1.2 | Macro e stile | 1031 |
| 25.1.3 | Istruzioni contenute nel testo sorgente | 1031 |
| 25.1.4 | Spazi superflui | 1031 |
| 25.1.5 | Esempio per cominciare | 1031 |
| 25.2 | Istruzioni fondamentali di Troff | 1032 |
| 25.2.1 | Argomenti numerici e unità di misura | 1033 |
| 25.2.2 | Dimensione e distanza tra le righe | 1033 |
| 25.2.3 | Caratteri da stampa | 1034 |
| 25.2.4 | Rientri e dimensioni varie | 1037 |
| 25.2.5 | Allineamento e interruzione del testo | 1038 |
| 25.2.6 | Tabulazioni | 1039 |
| 25.2.7 | Comandi che causano un «break» | 1040 |
| 25.2.8 | Macro | 1040 |
| 25.2.9 | Margini e Intestazioni | 1042 |
| 25.2.10 | Ambienti | 1043 |
| 25.2.11 | Titoli | 1044 |
| 25.2.12 | Importazione di file esterni | 1044 |
| 25.3 | Precompilatori | 1045 |
| 25.3.1 | Tbl | 1046 |
| 25.3.2 | Eqn | 1047 |
| 25.4 | Groff | 1048 |
| 25.5 | Documentazione Man | 1049 |
| 25.6 | Riferimenti | 1051 |

Troff e Nroff sono programmi di elaborazione e impaginazione testi per la produzione di documenti che possano essere riprodotti anche attraverso sistemi di stampa elementare, come gli schermi dei terminali a caratteri. Troff e Nroff sono due programmi più o meno compatibili che si completano a vicenda: il primo permette la stampa di qualità grafica, mentre il secondo è specializzato per la produzione di formati elementari come quello per lo schermo a caratteri. La distinzione è dettata dalla tradizione, dal momento che spesso si tratta dello stesso programma, avviato con nomi differenti.

Troff è nato nel 1973, scritto in linguaggio *assembler* per il PDP-11, riscritto successivamente in C. Ancora oggi Troff fa parte integrante della storia di Unix, in particolare per il fatto che il sistema di presentazione delle pagine di manuale. Pertanto, anche i sistemi GNU ne hanno uno: Groff.

Gli esempi di questo capitolo si basano principalmente su Groff, per il quale, nel momento in cui si scrivono queste note, non esiste la possibilità di leggere un file sorgente in formato UTF-8, mentre è possibile solo generare un risultato adatto a un terminale che utilizza invece tale codifica.

25.1 Logica di funzionamento ed esempio di partenza

Troff e Nroff si occupano di trasformare un testo, scritto con determinati codici di composizione, in un formato intermedio che successivamente deve essere rielaborato da un programma specifico per il tipo di stampa o visualizzazione che si vuole ottenere. Per arrivare a questo, si utilizza normalmente un condotto, più o meno nella forma seguente:

```
troff sorgente_troff | programma_di_rielaborazione
```

```
nroff sorgente_nroff | programma_di_rielaborazione
```

Per il momento, questo deve essere visto solo come un concetto di massima, perché in pratica manca ancora qualcosa. Lo sviluppo di questo sistema di composizione ha portato alla nascita di programmi di contorno che si occupano di semplificare la descrizione tipografica di elementi comuni di composizione. Questi programmi si collocano generalmente prima di `troff` o `nroff`. Nello schema seguente si fa un esempio dell'uso di `Eqn`, un filtro che facilita l'inserimento delle equazioni in un sorgente Troff.

```
eqn sorgente_troff | troff | programma_di_rielaborazione
```

In tal caso, come si vede, `troff` riceve il sorgente dallo standard input.

25.1.1 Avvio degli eseguibili «troff» e «nroff»

Gli eseguibili `troff` e `nroff` trasformano i file indicati tra gli argomenti, oppure lo standard input, in un formato intermedio contenente le informazioni necessarie per ottenerne la stampa o la visualizzazione.

```
troff [opzioni] [sorgente_troff...]
```

```
nroff [opzioni] [sorgente_nroff...]
```

Il sorgente per `troff` è un po' diverso da quello di `nroff`, ma in generale si usa quasi sempre solo il primo, essendo quello che richiede l'indicazione di più dettagli.

Le opzioni seguenti sono comuni al Troff originale e a quello di GNU.

| Opzione | Descrizione |
|---------------------------------|---|
| <code>-a</code> | Genera una composizione approssimativa del risultato, in formato testo (ASCII). |
| <code>-i</code> | Dopo aver letto i file indicati negli argomenti, legge anche dallo standard input. |
| <code>-nn_iniziale</code> | Permette di stabilire esplicitamente il numero della prima pagina. |
| <code>-oelenco_pagine</code> | Permette di specificare le pagine da stampare. L'elenco è separato da virgole (senza l'inserzione di spazi) e si possono indicare degli intervalli attraverso una notazione del tipo <code>'m-n'</code> . In particolare, se in un intervallo non viene indicata la pagina iniziale, si intende la prima, se invece manca quella finale, si intende l'ultima. |
| <code>-rregistro_numero</code> | Permette di definire il valore di un registro. Il registro è espresso da un lettera alfabetica, mentre il numero può essere espresso attraverso un'espressione numerica di Troff. |
| <code>-mnome</code> | Definisce il nome di un gruppo di macro da utilizzare prima di iniziare l'interpretazione dei file. Si tratta del riferimento al file che corrisponde al modello <code>tmac.nome</code> , contenuto in una directory appropriata in base alla realizzazione di Troff. |
| <code>-Tnome_dispositivo</code> | Permette di specificare il tipo di dispositivo di stampa, o di visualizzazione, per il quale viene composto il documento. Anche se il formato generato da Troff non è quello finale, ciò dipende dalla scelta fatta con questa opzione. |
| <code>-Fpercorso</code> | Permette di indicare la directory contenente le informazioni sui caratteri che si vogliono utilizzare. |

Segue la descrizione di alcuni esempi.

```
* $ troff -Tps -ms mio.troff [Invio]
```

Elabora il file `mio.troff`, utilizzando il dispositivo `ps` e il pacchetto di macro `s` (`tmac.s`).

```
* $ troff -Tps -ms -o4,6,8-10 mio.troff [Invio]
```

Come nell'esempio precedente, limitandosi a emettere il risultato riferito alle pagine 4, 6, 8, 9 e 10.

25.1.2 Macro e stile

I sorgenti Troff potrebbero essere realizzati fornendo tutte le indicazioni necessarie a definire l'aspetto finale del documento e utilizzando solo le istruzioni elementari di questo sistema di composizione. In alternativa possono essere definite delle macro, all'interno del sorgente stesso o in file esterni. È normale fare uso di Troff facendo riferimento a un file di macro esterno, che in pratica permette di definire uno stile generale del documento; per questo si utilizza l'opzione standard `-m`, come viene mostrato in seguito negli esempi. In ogni caso il pacchetto di macro più comune, già dalle origini di Troff, è `s`, corrispondente al file `tmac.s`.¹

25.1.3 Istruzioni contenute nel testo sorgente

Le istruzioni di Troff possono distinguersi fondamentalmente in: comandi, che iniziano sempre con un punto singolo nella prima colonna del sorgente, e sequenze di escape che possono essere collocate all'interno del testo normale. Un comando appare sempre da solo in una riga, come nel caso seguente,

```
Testo normale
.ft B
testo in neretto
```

dove `testo normale` e `testo in neretto` sono intesi come una sequenza che potrebbe risultare riprodotta sulla stessa riga, o comunque appartenendo in ogni caso allo stesso paragrafo. In particolare, dopo l'apparizione del comando `.ft B`, il testo viene reso in neretto. Una sequenza di escape, al contrario, non interrompe il testo nel sorgente:

```
Testo normale \fBtesto in neretto
```

In questo caso, `\fB` è una sequenza di escape che indica l'inizio del neretto. Come si può intuire, non è possibile iniziare una riga di testo con un punto, perché questo verrebbe interpretato come un comando di Troff; nello stesso modo, alcune sequenze di escape seguite da testo normale possono essere interpretate in modo erraneo.

25.1.4 Spazi superflui

Troff è sensibile alla presenza di spazi orizzontali e verticali superflui; questi vengono mantenuti nel documento finale. In condizioni normali, Troff ignora le interruzioni di riga inserite nel sorgente: quando quello che segue è un comando o un'altra riga di testo, sostituisce queste interruzioni con uno spazio orizzontale normale, ricomponendo in pratica i paragrafi a seconda del formato finale.

25.1.5 Esempio per cominciare

Anche se non è stato ancora mostrato il «linguaggio» di un sorgente Troff, contando sull'intuizione del lettore, è il caso di proporre un esempio elementare che permetta di verificarne il funzionamento.

```
.\" Questo è un esempio di documento scritto utilizzando il
.\" linguaggio di composizione Troff.
.\"
.\" Viene definita la dimensione del testo: il margine
.\" sinistro di 4 cm e l'ampiezza del testo di 8 cm.
.po 4c
.ll 8c
.\" Inizia il documento.
.ft B
1. Introduzione a Troff

.ft P
Questo \('e un esempio di documento scritto in modo
tale da poter essere elaborato con Troff.
In questo caso, si presume che verr\('a utilizzato
```

```

lo stile '\fBs\FP' (con l'opzione \EB\ms\FP).

.ft B
1.1 Paragrafi

.ft P
Il testo di un paragrafo termina quando nel sorgente viene
incontrata una riga vuota.

Per la precisione, gli spazi verticali vengono rispettati,
per cui le righe vuote si traducono in spazi tra i
paragrafi, anche quando queste sono più di una.

Questo \('e l'inizio di un nuovo paragrafo dopo tre righe
vuote di separazione.

```

Supponendo che il file si chiami 'esempio.troff' e che si utilizzi la versione GNU di Troff, si potrebbe ottenere la conversione in PostScript attraverso il comando seguente, che genera il file 'esempio.ps'.

```
$ troff -Tps -ms esempio.troff | grops > esempio.ps [Invio]
```

In alternativa, si potrebbe ottenere un file adatto per la visualizzazione attraverso un terminale a caratteri, configurato per la codifica ISO 8859-1, con il comando seguente:

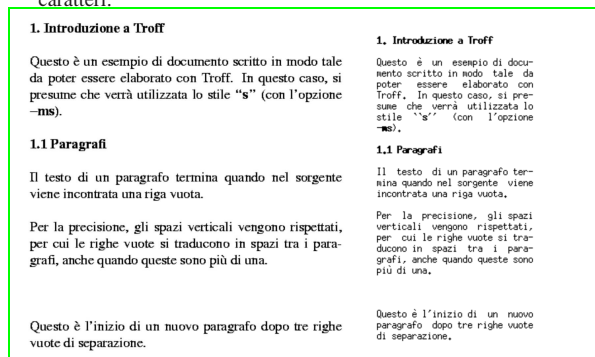
```
$ troff -Tlatin1 -ms esempio.troff | grotty > esempio.tty [Invio]
```

Oppure, se il terminale è configurato per la codifica UTF-8:

```
$ troff -Tutf8 -ms esempio.troff | grotty > esempio.tty [Invio]
```

I risultati si vedono nella figura 25.5. Da queste non si vedono i margini, ma per il momento il problema è trascurabile.

Figura 25.5. A sinistra il risultato della composizione del sorgente Troff di esempio in PostScript; a destra il risultato della composizione per la visualizzazione attraverso un terminale a caratteri.



In questo esempio si fa uso del pacchetto di macro 's', che tra le altre cose definisce i margini del testo. All'inizio del sorgente sono stati usati espressamente dei comandi per modificare i margini, in deroga a quanto prestabilito dallo stile del pacchetto di macro prescelto.

25.2 Istruzioni fondamentali di Troff

In queste sezioni viene mostrato l'uso di alcune istruzioni fondamentali di Troff. La loro descrizione è limitata anche in considerazione del fatto che Troff è un sistema di composizione superato, benché tuttora efficace; infatti, al suo posto conviene approfondire piuttosto l'uso di altri programmi, come TeX per esempio.

Alla fine di queste sezioni si trova una tabella riassuntiva dei comandi «vitali» di Troff, cioè di quelli che vengono descritti qui.

25.2.1 Argomenti numerici e unità di misura

Alcuni comandi hanno un argomento numerico che esprime una quantità o una dimensione. A seconda della circostanza, tale valore può essere espresso in modo fisso, oppure come incremento o riduzione. Se è ammissibile l'incremento, tale numero può essere indicato prefissato dal segno '+', se è possibile la riduzione può essere prefissato dal segno '-'. Gli incrementi e le riduzioni permettono di scrivere istruzioni relative, che si adattano a seconda di altre scelte già fatte nel sorgente Troff. In alcuni casi ci sono dei valori che possono essere espressi in forma frazionaria, utilizzando il punto per separare la parte intera dalle cifre decimali.

Quando gli argomenti riguardano valori che esprimono una lunghezza, possono essere seguiti immediatamente da una lettera che ne esprima l'unità di misura. La tabella 25.6 mostra l'elenco di queste sigle.

Tabella 25.6. Sigle di identificazione dell'unità di misura.

| Sigla | Unità di misura corrispondente |
|-------|--------------------------------|
| i | Pollici. |
| c | Centimetri. |
| p | Pica = 1/6 di pollice. |
| m | Dimensione della lettera «m». |
| n | Dimensione della lettera «n». |
| pt | Punti = 1/72 di pollice. |
| u | Unità base. |
| v | Altezza della riga. |

25.2.2 Dimensione e distanza tra le righe

Normalmente, il corpo dei caratteri è di 10 punti e la distanza tra le righe è di 12 punti. Il comando normale per ridefinire il corpo è '.ps', che sta per *Point size*:

```
.ps [ [+ | - ] n ]
```

Come si vede dallo schema sintattico, questo comando ammette la possibilità di fissare il valore, oppure di incrementarlo e di diminuirlo indicando una dimensione espressa in punti tipografici (lo si intuisce dal nome). Se non viene fornito l'argomento numerico, si intende ripristinare la dimensione al valore fissato precedentemente.

In alternativa può essere usata la sequenza di escape '\s', secondo la sintassi seguente:

```
\s [ + | - ] n
```

È importante osservare che il numero di punti che può essere indicato dipende dalla disponibilità effettiva in base al tipo di carattere a disposizione; inoltre, nel caso della sequenza di escape '\s', possono essere utilizzate solo **due** cifre numeriche. In particolare, se si utilizza la dimensione nulla, cioè il numero zero, si ottiene il ripristino della dimensione precedente.

Quando si usa la sequenza '\s' per specificare un valore composto da una sola cifra numerica, è importante che il carattere successivo non sia un numero, altrimenti si riesce a confondere Troff.

Il ridimensionamento dei caratteri viene usato normalmente assieme al controllo della distanza tra le righe. Per questo si usa il comando '.vs' (*Vertical space*).

```
.vs [ n [ unità_di_misura ] ]
```

L'argomento numerico serve a precisare la distanza tra la base di una riga e la base di quella successiva. Il valore viene espresso normalmente in punti, a meno che sia specificato un tipo di unità di misura speciale. In mancanza dell'argomento numerico, il comando ripristina la distanza precedente.

Di solito, la distanza tipica tra le righe è pari al 120 % del corpo dei caratteri utilizzati.

Tra due righe può essere indicato anche uno spazio aggiuntivo, attraverso il comando `‘.sp’` (*Space*).

```
.sp [n[unità_di_misura]]
```

La sintassi di `‘.sp’` è la stessa di `‘.vs’`, con la differenza che si riferisce a uno spazio aggiuntivo inserito una sola volta in corrispondenza della posizione del comando. In particolare, se `‘.sp’` viene usato senza argomento, si ottiene una riga bianca vuota della dimensione attuale dell'altezza delle righe.

Segue la descrizione di alcuni esempi.

```
• Testo normale
  .ps 24
  testo ingrandito
```

Cambia il corpo del testo che segue il comando, senza curarsi della distanza tra le righe.

```
• Il sistema operativo GNU/Linux funziona su...
```

Ammetto che il testo normale abbia un corpo di 10 punti, fa in modo che la parola «LINUX» sia ottenuta con una lettera iniziale di dimensione normale e la parte restante con lettere leggermente ridotte (otto punti). Alla fine, il corpo precedente viene ripristinato.

```
• .ps 24
  .vs 28p
```

Cambia la dimensione del corpo e della distanza tra le righe.

```
• .sp 1c
```

Inserisce uno spazio verticale aggiuntivo di 1 cm.

25.2.3 Caratteri da stampa

Storicamente, la gestione dei tipi di carattere di Troff è stata piuttosto limitata: erano disponibili quattro aree all'interno delle quali potevano essere «montati» tipi differenti di carattere. Queste aree esistono anche nelle versioni più recenti di Troff e generalmente servono a contenere, nell'ordine: un carattere tondo, un corsivo, un neretto e una serie di simboli (che comunque si ottengono attraverso delle sequenze di escape). Per poter utilizzare caratteri differenti, occorre sostituire il carattere di un'area, montando al suo posto quello desiderato. Attualmente, questa operazione non è più necessaria; generalmente si utilizzano i caratteri normali (quelli appena elencati) e si specifica un tipo di carattere differente da questi solo quando serve, senza bisogno di montarlo esplicitamente.

Quando si fa riferimento ai caratteri delle aree normali, si può utilizzare il comando `‘.ft’` (*Font*), seguito da una sigla alfabetica che ne definisce la forma.

```
.ft [R|I|B]
```

Le lettere `‘R’`, `‘I’`, `‘B’` indicano rispettivamente: *Roman*, *Italic* e *Bold*, riferendosi quindi a un carattere tondo normale, corsivo o neretto. Se non viene specificato l'argomento, il comando `‘.ft’` ripristina il carattere usato precedentemente.

All'interno del testo può essere usata la sequenza `‘\f’`, seguita immediatamente da una delle lettere viste per l'argomento di `‘.ft’`. In particolare, `‘\fP’` serve a ripristinare il carattere precedente.

Prima di proseguire vale la pena di vedere il significato di un comando un po' strano: `‘.ul’` (*Underline*). Letteralmente si tratta di una richiesta di «sottolineatura» che interviene solo nel testo del sorgente che lo segue immediatamente. Tuttavia, secondo la tipografia, il sottolineato è una forma di evidenziamento deprecabile, per cui questo si traduce in pratica in un corsivo.

Per utilizzare altri tipi di carattere oltre quelli standard che si trovano a essere già montati nel sistema di Troff, si può utilizzare il comando `‘.ft’`, seguito da un argomento che esprima direttamente il tipo di carattere scelto. A questo proposito, è bene chiarire che le sigle `‘R’`, `‘I’` e `‘B’` si riferiscono sempre ai tipi di carattere montati nelle prime tre aree standard, per cui, non è possibile caricare un tipo di carattere differente e pretendere poi di ottenerne il corsivo con il comando `‘.ft I’`. La tabella 25.11 riporta l'elenco di alcuni tipi di carattere che dovrebbe essere possibile utilizzare con la propria realizzazione di Troff.

Tabella 25.11. Sigle di identificazione di alcuni tipi di carattere.

| Sigla | Descrizione |
|-------|----------------------------------|
| R | Generalmente è il Times normale. |
| I | Generalmente è il Times obliquo. |
| B | Generalmente è il Times neretto. |
| H | Helvetica. |
| HI | Helvetica corsivo. |
| HB | Helvetica neretto. |
| HBI | Helvetica neretto obliquo. |
| CR | Courier. |
| CI | Courier obliquo. |
| CB | Courier neretto. |
| CBI | Courier neretto obliquo. |

I caratteri speciali, tra cui eventualmente anche le lettere accentate, possono essere ottenuti attraverso delle sequenze di escape che iniziano con `‘\’` e si compongono di altri due caratteri. La tabella 25.12 mostra l'elenco di alcune di queste sequenze riferite alle lettere accentate.

Tabella 25.12. Alcune sequenze di escape per le lettere accentate di Troff.

| | | | | |
|----------------------|----------------------|----------------------|----------------------|----------------------|
| <code>\\('a</code> á | <code>\\('A</code> Á | <code>\\(^a</code> â | <code>\\(^A</code> Â | <code>\\('a</code> à |
| <code>\\('A</code> À | <code>\\(oa</code> â | <code>\\(oA</code> Â | <code>\\(~a</code> ã | <code>\\(~A</code> Ã |
| <code>\\(:a</code> ä | <code>\\(:A</code> Ä | <code>\\(ae</code> æ | <code>\\(AE</code> Æ | <code>\\(,c</code> ç |
| <code>\\(,C</code> Ç | <code>\\('e</code> é | <code>\\('E</code> É | <code>\\(^e</code> ê | <code>\\(^E</code> Ê |
| <code>\\('e</code> è | <code>\\('E</code> È | <code>\\(:e</code> ë | <code>\\(:E</code> Ë | <code>\\('i</code> í |
| <code>\\('I</code> Í | <code>\\(^i</code> î | <code>\\(^I</code> Î | <code>\\('i</code> ï | <code>\\('I</code> Ì |
| <code>\\(:i</code> ï | <code>\\(:I</code> Ì | <code>\\(~n</code> ñ | <code>\\(~N</code> Ñ | <code>\\('o</code> ó |
| <code>\\('O</code> Ó | <code>\\(^o</code> ô | <code>\\(^O</code> Ô | <code>\\('o</code> ò | <code>\\('O</code> Ò |
| <code>\\(/o</code> ø | <code>\\(/O</code> Ø | <code>\\(~o</code> õ | <code>\\(~O</code> Õ | <code>\\('o</code> ö |
| <code>\\(:o</code> Ö | <code>\\('u</code> ú | <code>\\('U</code> Ú | <code>\\(^u</code> û | <code>\\(^U</code> Û |
| <code>\\('u</code> ù | <code>\\('U</code> Û | <code>\\(:u</code> ü | <code>\\(:U</code> Ü | <code>\\('y</code> ý |
| <code>\\('Y</code> Ý | <code>\\(:y</code> ÿ | <code>\\(ss</code> ß | | |

Quando si utilizza Troff di GNU si può scrivere il sorgente Troff

utilizzando la codifica ISO 8859-1; tuttavia, mancando ancora la capacità di leggere un file in formato UTF-8, è meglio scrivere il sorgente secondo il metodo tradizionale; inoltre, ci sono altre sequenze che sono indispensabili per ottenere effetti tipografici particolari. La tabella 25.13 riassume i casi più importanti.

Tabella 25.13. Caratteri e sequenze di escape per ottenere simboli speciali che vanno oltre la codifica utilizzata.

| Sequenza | Descrizione |
|-------------------|---|
| <code>\e</code> | Barra obliqua inversa (<code>'\`</code>). |
| <code>\'</code> | Accento acuto ben distinguibile. |
| <code>\`</code> | Accento grave ben distinguibile. |
| <code>-</code> | Trattino corto. |
| <code>\(hy</code> | Trattino corto di sillabazione. |
| <code>\-</code> | Il segno meno (<code>'-'</code>). |
| <code>\(mi</code> | Il segno meno (<code>'-'</code>). |
| <code>\(em</code> | Trattino lungo. |
| <code>\(ru</code> | Trattino basso. |
| <code>\(ul</code> | Trattino di sottolineatura (più basso). |
| <code>\(rn</code> | Trattino alto. |
| <code>\(bu</code> | Pallino. |
| <code>\(sq</code> | Quadrato. |

Infine, Troff consente l'uso delle lettere greche, utilizzando delle sequenze di escape che iniziano per `\(*` seguite immediatamente da una lettera (dell'alfabeto latino-inglese) che in qualche modo può avere una corrispondenza con quella greca.

Segue la descrizione di alcuni esempi.

- `.ft B`
Il testo a partire dalla riga successiva del sorgente Troff, viene reso in neretto.
- `.ft`
Ripristina il carattere utilizzato in precedenza.
`Testo normale \fBtesto in neretto\fP testo normale.`
Nella frase, il pezzo `'testo in neretto'` viene reso in neretto. La sequenza `'\fP'` serve a ripristinare il carattere precedente.
- `.ft H`
Il testo a partire dalla riga successiva del sorgente Troff, viene reso con il carattere Helvetica normale.
- `.ft HI`
Il testo a partire dalla riga successiva del sorgente Troff, viene reso con il carattere Helvetica obliquo.
- L'opzione `\-ms` serve a definire l'uso del pacchetto di macro `«s»`.
Attraverso la sequenza `'\-'`, viene richiesto espressamente l'uso di un trattino normale.
L'opzione
`.ft CR`
`\-ms`
`.ft`
serve a definire l'uso del pacchetto di macro `«s»`.
Come nell'esempio precedente, ma il testo `'-ms'` viene reso con il carattere Courier che risulta più adatto (essendo a larghezza fissa).

```
Il file
.ft CR
mio\rufile
.ft
serve a...
```

Fa in modo che la parola `'mio_file'` appaia in Courier, utilizzando in particolare un trattino basso.

- `\(*b`
La lettera greca beta minuscola.
- `\(*W`
La lettera greca omega maiuscola.
- `\(*w`
La lettera greca omega minuscola.

25.2.4 Rientri e dimensioni varie

Il dimensionamento della pagina e del testo all'interno di questa, avviene in modo un po' strano. Per cominciare, il foglio normale di riferimento è il formato lettera (8,5 in × 11 in); all'interno di questo spazio può essere definito un margine sinistro e una larghezza della riga. I margini superiore e inferiore sono generalmente predefiniti attraverso il pacchetto di macro utilizzato.

Se non si vuole approfondire l'uso di Troff, conviene limitarsi ad accettare il più possibile le convenzioni del pacchetto di macro tradizionale, quello che viene richiamato con l'opzione `'-ms'`. Questo significa che il foglio è in formato lettera (anche se poi si stampa su un A4) e i margini superiore e inferiore sono di un pollice di altezza.

Il margine sinistro della pagina può essere modificato attraverso il comando `'po'` (*Page offset*), che viene usato normalmente prima di iniziare il testo del documento.

```
.po n [unità di misura]
```

Per definire la larghezza del testo si utilizza il comando `'ll'` (*Line length*). Anche questo può essere usato con valori di incremento o di riduzione, per mantenere un riferimento con il testo precedente.

```
.ll [+|-] n [unità di misura]
```

All'interno del testo è possibile modificare il margine con il comando `'in'` che fa riferimento al margine assoluto della pagina. Spesso, il comando `'in'` viene usato con valori di incremento o di riduzione, in modo da mantenere un riferimento con la situazione precedente del testo.

```
.in [+|-] n [unità di misura]
```

Incrementando il rientro con il comando `'in'`, si riduce conseguentemente la larghezza della riga; se invece lo si diminuisce, la larghezza della riga aumenta in relazione. Questo serve a mantenere il margine destro invariato, a seguito dell'utilizzo del comando `'in'`. Per ottenere il rientro di una sola riga, si utilizza il comando `'ti'`.

```
.ti [+|-] n [unità di misura]
```

I comandi che sono stati descritti accettano tutti delle dimensioni espresse anche in forma frazionaria, utilizzando il punto per separare la parte intera dalle cifre decimali.

Segue la descrizione di alcuni esempi.

- `.po 0`
Pone il margine sinistro della pagina al valore minimo possibile.
- `.po 1i`
Pone il margine sinistro della pagina a un pollice.
- `.in 1c`

Inizia un margine sinistro che si pone a un centimetro più a destra del margine della pagina.

```
.in +1c
```

Incrementa il margine sinistro di un centimetro.

```
.ll 15.5c
```

Definisce la larghezza del testo di 15,5 cm.

```
.in +0.5c
.ll -0.5c
Testo...
.ll +0.5c
.in -0.5c
```

Rientra il testo a sinistra di un mezzo centimetro e anche a destra della stessa dimensione, riducendo la larghezza della riga. Dopo la scrittura del testo (che così appare inscatolato), vengono ripristinate le dimensioni precedenti.

```
.ti +1c
```

La prima riga del paragrafo che segue il comando viene scritta con un rientro di un centimetro.

```
.ti -1c
```

Rientra all'indietro di un centimetro.

25.2.5 Allineamento e interruzione del testo

Di solito, utilizzando il pacchetto di macro 's', si ottiene un documento in cui il testo è allineato a sinistra e a destra (giustificato); inoltre, le righe del sorgente che appaiono in sequenza, senza spazi verticali intermedi, vengono unite assieme. Per indicare esplicitamente un'interruzione di riga si può usare il comando '.br' (*Break*) che non prevede alcun argomento. Inoltre, per richiedere espressamente di saltare una pagina, si può usare il comando '.bp' (*Break page*), nello stesso modo.

Si è accennato al fatto che normalmente il testo contenuto nel sorgente viene riunito assieme prima di definire l'impaginazione finale. Per richiedere esplicitamente questo comportamento, si utilizza il comando '.fi' (*Fill*), mentre per fare in modo che vengano rispettate le interruzioni di riga che appaiono nel sorgente, si usa il comando '.nf' (*No fill*).

L'allineamento del testo viene richiesto attraverso il comando '.ad' (*Adjust*) con un argomento composto da una lettera che permette di scegliere come allinearli.

```
.ad l|r|c|b
```

Le lettere 'l', 'r', 'c' e 'b' servono a richiedere rispettivamente l'allineamento sinistro, destro, centrato, o simultaneo (destra e sinistra).

Eventualmente, il comando '.ce' permette di ottenere la centratura di un certo numero di righe del sorgente.

```
.ce n_righe
```

Segue la descrizione di alcuni esempi.

```
Bla bla bla...
.br
qui inizia una riga nuova
```

Il testo viene interrotto esplicitamente in modo da farlo riprendere in una riga successiva.

```
.ft CR
.nf
uno     due     tre
quattro cinque sei
.fi
.ft
```

Viene riportato del testo catturato da un comando o da una schermata. Per questo si utilizza il carattere Courier e si specifica che

le interruzioni di riga devono essere rispettate. Alla fine, viene ripristinato il comportamento normale.

```
.ad r
testo a bandierina allineato a destra
.ad b
testo allineato simultaneamente a sinistra e a destra.
```

Allinea il testo a destra e successivamente lo rimette nella situazione normale di allineamento simultaneo.

```
.ce 1
testo centrato
testo normale
```

Centra solo la prima riga successiva del testo che appare nel sorgente.

25.2.6 Tabulazioni

La tabulazione orizzontale che si ottiene con il codice ASCII <HT>, viene interpretata regolarmente da Troff, che lo intende come un salto allo *stop di tabulazione* successivo. Questi stop possono essere regolati attraverso il comando '.ta', ma se non sono definiti espressamente, Troff utilizza gli stop predefiniti che dovrebbero trovarsi ogni quarto di pollice (poco più di mezzo centimetro).

```
.ta stop_1 stop_2...
```

L'argomento di '.ta' è costituito da una serie di numeri (seguiti dall'unità di misura) che esprimono la distanza degli stop di tabulazione dal margine sinistro del testo.

Generalmente, quando si usano gli stop di tabulazione per scrivere delle tabelle elementari, si fa in modo che le interruzioni di riga vengano rispettate, attraverso l'uso del comando '.nf'.

Quando si cerca di incolonnare dei numeri, può essere utile la sequenza '\0' che si traduce in uno spazio orizzontale della stessa ampiezza di una cifra numerica.

Infine, attraverso il comando '.tc' è possibile richiedere l'utilizzo di un carattere particolare per riempire lo spazio della tabulazione. '.tc' richiede un argomento composto da un solo carattere, anche una sequenza di escape.

```
.tc carattere_di_riempimento
```

Segue la descrizione di alcuni esempi.

```
.nf
.ta 3c 6c 9c
      1      2      3
      11     22     333
.fi
```

Dopo aver fatto in modo che vengano rispettate le interruzioni di riga che appaiono nel sorgente, definisce tre stop di tabulazione a 3 cm, 6 cm e 9 cm, rispettivamente. Il contenuto della tabella appare allineato a sinistra.

```
.nf
.tc \{ru
.ta 15c
Nominativo      ,
.ta 3c 7c 12c 15c
CAP      Citt\('a      Via      N.      ,
.fi

bla bla bla
```

Questo rappresenta un esempio un po' più complesso, dove si vuole predisporre un modello da compilare. Si osservi la riga in cui appare la parola 'Nominativo': lo spazio che si vede prima della virgola è ottenuto con un carattere di tabulazione orizzontale che viene riempito da caratteri '\{ru', corrispondenti a trattini bassi. Ogni volta che il modello cambia elementi, occorre ridefinire la posizione degli stop di tabulazione.

| |
|--|
| Nominativo |
| CAP _____ Città _____ Via _____ N. _____ |
| bla bla bla |

25.2.7 Comandi che causano un «break»

«

Nella logica di funzionamento di Troff, alcuni comandi causano un'interruzione nel flusso del testo, costringendo Troff a interromperlo e a riprenderlo nella riga successiva. Un esempio evidente è dato dal comando `.br`, che si usa proprio per questo: ottenere un'interruzione di riga nel documento finale. Anche l'utilizzo di altri comandi implica un'interruzione del testo, benché questo non venga richiesto esplicitamente:

| | |
|------------------|--|
| <code>.br</code> | interruzione di riga; |
| <code>.bp</code> | interruzione di pagina; |
| <code>.sp</code> | spazio verticale; |
| <code>.in</code> | margini sinistro; |
| <code>.ti</code> | rientro temporaneo; |
| <code>.nf</code> | rispetto delle interruzioni nel sorgente; |
| <code>.fi</code> | scorrimento del testo indipendente dalle interruzioni che appaiono nel sorgente; |
| <code>.ce</code> | centratura di alcune righe successive. |

Il fatto che questi comandi interrompano il flusso del testo dovrebbe apparire logico al lettore; probabilmente si potrebbe trovare strano il fatto che il comando `.ad` non faccia parte di questo gruppo. Troff consente di spogliare questi comandi della funzionalità di interruzione (o *break*), sostituendo il punto iniziale con un apostrofo. Per fare un esempio estremo, `'br` diventa un'interruzione di riga senza interruzione, in pratica non serve più. Si osservi comunque l'esempio seguente:

```
bla bla bla bla--
Bla bla bla bla
'sp 1c
bla bla bla bla--
```

Si mostra l'uso del comando `'sp 1c` che ha lo scopo di inserire uno spazio verticale di un centimetro. Avendo usato l'apostrofo al posto del punto, lo spazio viene inserito quando il testo precedente è arrivato alla fine della riga. In pratica, parte del testo che si trova sopra il comando potrebbe essere riprodotto nel documento finale **dopo** lo spazio verticale.

Chi ha difficoltà a comprendere il senso della cosa, può limitarsi a tenere a mente che è opportuno privare questi comandi della funzione di interruzione di riga quando questi vengono usati per predisporre delle intestazioni o dei piè di pagina.

25.2.8 Macro

«

Troff consente di creare i propri comandi, ovvero delle macro, che permettono di semplificare e uniformare il proprio documento. Per comprendere il senso di questo occorre presentare subito un esempio; si osservi il testo seguente:

```
.de T1
.sp
.ft B
..
```

Questo pezzo di istruzioni Troff serve a dichiarare la macro `T1` che quando utilizzata si traduce nei comandi `.sp` e `.ft B`. Si intuisce che il comando `.de` serve a iniziare la dichiarazione della macro e i due punti in orizzontale finali servono a concluderla la dichiarazione.

```
.de nome_macro
dichiarazione
dichiarazione
...
dichiarazione
..
```

Il nome della macro che si crea deve essere di due caratteri e generalmente si utilizzano le lettere maiuscole in modo da essere certi di non interferire con i comandi normali di Troff.

Si noti che il nome della macro che si dichiara non ha il punto iniziale.

Tornando all'esempio, la macro `T1` potrebbe servire per spaziare ed evidenziare un titolo di qualcosa. Usandola, occorre ricordare che modifica il tipo di carattere, dal momento che passa alla scrittura in neretto; volendo si può preparare un'altra macro per uniformare i paragrafi normali.

```
.de P1
.ft R
.ti +1m
..
```

Questa volta viene dichiarata la macro `P1` con lo scopo di ripristinare l'uso del carattere normale e di inserire un rientro temporaneo della prima riga (di un quadratone), così da ottenere un paragrafo con rientro iniziale. Quello che si vede sotto è un esempio di utilizzo di queste macro.

```
.T1
Introduzione al documento

.P1
Questo paragrafo tratta dell'inizio della fine e viceversa..

.P1
Questo paragrafo parla d'altro..

.T1
Approfondimento

.P1
\('E nato prima l'uovo o la gallina?
```

Un esempio un po' più interessante potrebbe essere quello della definizione di due macro allo scopo di semplificare la scrittura di testo circoscritto in qualche modo, per esempio per mostrare ciò che appare su un terminale o quello che si ottiene da una stampa.

```
.de TA
.ft CR
.ps 8
.in +2m
.nf
..
```

```
.de TC
.fi
.in -2m
.ps
.ft
..
```

La prima macro serve a iniziare la scrittura in Courier con un corpo leggermente più piccolo del solito, rispettando le interruzioni di riga e aggiungendo due quadratoni al margine sinistro. La seconda serve a ripristinare la situazione precedente. Si osservi l'esempio seguente in cui si mostra in che modo utilizzarle.

```
Il comando ls \-l mostra un elenco simile a quello seguente:

..TA
-rwxr-xr-x  1 root  root    2864 ott 14 06:44 arch
-rwxr-xr-x  1 root  root   62660 ago 29 01:43 ash
-rwxr-xr-x  1 root  root    4892 ago  5 21:15 basename
-rwxr-xr-x  1 root  root   353944 ott 13 01:23 bash
...
..TC

Si osservi in particolare la propriet\('a dei file...
```

Si comprende che il vantaggio di usare le macro sta nella possibilità di uniformare lo stile personale del documento e di poter modificare tale stile in modo più facile, intervenendo solo sulla definizione delle macro stesse.

Infine, è bene accennare alla possibilità di dichiarare delle macro con argomenti: all'interno della definizione di una macro, le sequenze formate da `'\$n'`, dove *n* è un numero da uno a nove, rappresentano l'*n*-esimo argomento. Si osservi l'esempio seguente:

```
.de DO
.br
data: \$1
.br
ora: \$2
.br
..
```

In questo modo, la macro `'DO'` permette di fornire due argomenti che rappresentino rispettivamente una data e un'ora.

```
bla bla bla
.DO 11/11/2011 11:11
bla...
```

Utilizzando la macro nel modo appena mostrato, si ottiene il testo seguente:

```
bla bla bla
data: 11/11/2011
ora: 11:11
bla...
```

Gli argomenti di una macro di distinguono in quanto separati da uno o più spazi. Se è necessario fornire un argomento che contiene spazi, occorre delimitarlo attraverso virgolette, come si vede nell'esempio che appare sotto.

```
.DO "11 11 2011" 11:11
```

25.2.9 Margini e Intestazioni

Se si utilizza un pacchetto di macro come `'s'`, questo si occupa da solo di dare alle pagine un'intestazione composta dal numero di pagina (a partire dalla seconda). Se si vuole fare a meno di un pacchetto di macro esterno, si può realizzare la propria intestazione ed eventualmente il proprio piè di pagina.²

La stampa di un'intestazione deve avvenire in modo regolare, ogni volta che si raggiunge la «fine» di una pagina. Troff non permette di definire esplicitamente i margini superiore e inferiore; questo lo deve fare il pacchetto di macro prescelto, oppure l'utente attraverso il controllo dato dal comando `'wh'` (*When*).

```
.wh [-]n_collocazione_verticale [unità] macro
```

Il comando `'wh'` permette di definire una «trappola» in corrispondenza di una particolare posizione verticale del testo; se il valore di tale collocazione è negativo, si intende riferito alla distanza dalla fine del foglio. Quando il testo del documento finale arriva al punto della trappola, si ottiene l'esecuzione della macro indicata come secondo argomento. Si osservi l'esempio:

```
.wh -2.5c PA
```

Quello che si vede serve a fare in modo che quando mancano meno di 2,5 cm dalla fine del foglio, venga eseguita la macro `'PA'`. Si

osservi a questo proposito che nel comando `'wh'` la macro viene indicata senza il punto consueto.

```
.de PA
'bp
..
```

L'esempio che si vede sopra è la creazione della macro `'PA'`, che ovviamente deve apparire prima di qualunque utilizzo, specialmente prima del comando `'wh'` che serve a richiamarla. La macro mostrata è la più banale possibile: si limita a eseguire un salto pagina (`'bp'`), senza imporre l'interruzione di riga. In pratica, quando scatta la trappola a 2,5 cm dalla fine del foglio, **viene completata la riga** e quindi la «carta» viene fatta avanzare fino all'inizio di una nuova pagina.

Quanto mostrato fino a questo punto serve solo a ottenere un margine inferiore di 2,5 cm e niente altro. Per inserire un margine superiore (che possa intervenire a partire dalla seconda pagina), occorre aggiungere qualcosa alla macro `'PA'`:

```
.de PA
'bp
'sp 2.5c
..
```

Come si vede, è stato aggiunto il comando `'sp 2.5c'` per ottenere lo stesso margine anche all'inizio della pagina.

È bene osservare che lavorando in questo modo, il margine superiore della prima pagina deve essere gestito direttamente nel testo, attraverso un comando `'sp'` o qualcosa di simile. Tuttavia, di solito la prima pagina viene usata come copertina, per cui non si avverte il problema del margine superiore che può funzionare automaticamente solo a partire dalla seconda.

Per preparare un'intestazione come si è abituati a vederle di solito, occorre mostrare il funzionamento del comando `'t1'`. Questo permette di definire una riga da collocare in un'intestazione, suddivisa in tre parti che si traducono in testo che viene poi allineato a sinistra, al centro e a destra.

```
.t1 'testo_a_sinistra' 'testo_al_centro' 'testo_a_destra'
```

Questo comando viene usato normalmente solo nelle intestazioni (o nei piè di pagina) e ha la particolarità di sostituire il carattere di percentuale (`'%`) con il numero delle pagine. L'esempio seguente mostra la solita macro `'PA'` un po' più raffinata.

```
.de PA
'bp
'sp 1.5c
.t1 'Introduzione a Troff' 'pagina %'
'sp 0.7c
..
```

In pratica, ogni volta che viene richiamata la macro, questa salta una pagina e dopo 1,5 cm stampa l'intestazione (nella parte centrale non c'è alcun testo), dove in particolare appare il numero della pagina all'estrema destra. Infine, dopo 0,7 cm continua il testo normale.

È bene ripetere che se si vogliono gestire direttamente i margini e le intestazioni, come negli esempi mostrati qui, è opportuno evitare di utilizzare stili esterni attraverso l'inclusione di pacchetti di macro richiamati con l'opzione `'-m'` di Troff.

25.2.10 Ambienti

Da quanto visto fino a questo punto su Troff, si può notare una certa difficoltà nel ripristinare l'impostazione precedente a una serie di comandi. In aiuto del compositore è possibile definire degli ambienti, uscendo dai quali si ripristina tutto come prima. Un ambiente viene definito con il comando `'ev'` (*Environment*), con il quale si seleziona un numero di ambiente prima di iniziare con una serie di comandi. Quando si vuole ripristinare tutto come prima, basta richiamare il comando `'ev'` senza argomenti.


```
.ev [n_ambiente]
```

Gli ambienti sono numerati a partire da zero; nella versione originale di Troff erano solo quattro (dal numero zero al numero tre), mentre nelle realizzazioni attuali possono essere molti di più. Per comprendere il funzionamento di questi dovrebbe bastare un esempio. Nella sezione precedente è stato visto come creare un'intestazione; considerando che il testo normale potrebbe essere inserito nell'ambiente zero, si potrebbe cambiare la definizione della macro di intestazione nel modo seguente:

```
.de PA
.ev 1
.bp
.sp 1.5c
.ps 8
.ft H
.tl 'Introduzione a Troff' 'pagina %'
.sp 0.7c
.ev
..
```

In questo modo viene definito un carattere Helvetica di otto punti. Alla fine, prima della conclusione della macro, viene ripristinato l'ambiente precedente.

25.2.11 Titoli

Nella sezione in cui si mostrava la preparazione di un'intestazione si è visto l'uso del comando `.tl` (*Title line*), ma è il caso di approfondire un po' la cosa. `.tl` serve per generare una sorta di titolo diviso in tre parti allineate rispettivamente a sinistra, al centro e a destra. Dal momento che per Troff non esiste una grande differenza tra le due cose, questo titolo può trovarsi sia in un'intestazione che nel testo normale. Il simbolo di delimitazione delle tre parti che lo compongono viene deciso nel momento in cui si scrivono le tre stringhe. Per esempio, nel comando

```
.tl 'sinistra|centro|destra'
```

il simbolo di delimitazione è l'apostrofo, ma potrebbe essere qualunque altra cosa, specialmente se l'apostrofo serve nel testo dell'intestazione.

```
.tl "L'altra faccia della medaglia" "pagina %"
```

Anche il simbolo usato per inserire il numero della pagina non è sempre lo stesso; quello comune è `%`, ma può essere modificato con il comando `.pc` (*Page character*).

```
.pc x
```

Per esempio, si potrebbe decidere di sostituirlo con un dollaro:

```
.pc $
.tl "Carta riciclata al 100%" "pagina $"
```

Infine, il comando `.tl` è autonomo per quel che riguarda la larghezza della riga. Se si vogliono cambiare i margini laterali, intervenendo anche con il comando `.ll`, conviene adeguare conseguentemente anche la larghezza del titolo su riga. Per questo si utilizza il comando `.lt` (*Length of title*).

```
.lt [+|-]n[unità_di_misura]
```

Nell'esempio seguente viene ridefinita la larghezza della riga del testo normale e anche quella del titolo su una riga.

```
.ll 10c
.lt 10c
```

25.2.12 Importazione di file esterni

Attraverso il comando `.so` è possibile incorporare un sorgente Troff esterno.

```
.so file
```

Alle volte viene utilizzato questo sistema per creare delle pagine di manuale con nomi differenti ma con lo stesso contenuto, evitando di utilizzare i collegamenti ai file.

Tabella 25.61. Riassunto dei comandi vitali di Troff.

| Comando | Break | Descrizione |
|---------------------------------|-------|---|
| <code>.ps [[+ -]n]</code> | | Dimensione del carattere in punti. |
| <code>\sn</code> | | Dimensione del carattere in punti. |
| <code>.vs [n [unità]]</code> | | Interlinea. |
| <code>.sp [n [unità]]</code> | break | Spazio aggiuntivo verticale. |
| <code>.ft [sigla]</code> | | Scelta di un tipo di carattere. |
| <code>\fx</code> | | Scelta di un tipo di carattere. |
| <code>.po n [unità]</code> | | Margine sinistro della pagina. |
| <code>.ll [+ -]n [unità]</code> | | Larghezza della riga. |
| <code>.in [+ -]n [unità]</code> | break | Margine sinistro del testo. |
| <code>.ti [+ -]n [unità]</code> | break | Rientro temporaneo. |
| <code>.br</code> | break | Interruzione di riga esplicita. |
| <code>.bp</code> | break | Interruzione di pagina esplicita. |
| <code>.fi</code> | break | Unione delle righe adiacenti nel sorgente. |
| <code>.nf</code> | break | Mantenimento delle interruzioni di riga del sorgente. |
| <code>.ad l r c b</code> | | Allineamento del testo. |
| <code>.ce n</code> | break | Centrata di alcune righe successive del sorgente. |
| <code>.ta nunità...</code> | | Definizione degli stop di tabulazione. |
| <code>.tc x</code> | | Definizione del carattere di riempimento delle tabulazioni. |
| <code>.de xx</code> | | Dichiarazione dell'inizio di una macro. |
| <code>..</code> | | Conclusione di una macro. |
| <code>.tl 'sx centro dx'</code> | | Titolo su una riga. |
| <code>.lt [+ -]n [unità]</code> | | Larghezza di un titolo su riga. |
| <code>.ev [n]</code> | | Selezione di un ambiente. |
| <code>.so file</code> | | Inclusione di un file esterno. |

Segue la descrizione di alcuni esempi.

```
.so presentazione
bla bla bla..
```

L'esempio mostra l'inclusione del file `presentazione` che deve trovarsi nella directory corrente nel momento in cui viene elaborato il file principale da Troff.

```
.so man1/gs.1
```

Questo è l'esempio del file `/usr/man/man1/ghostscript.1` che fa semplicemente riferimento al file `man1/gs.1`.

25.3 Precompilatori

Il linguaggio di composizione Troff consente l'uso di comandi molto più raffinati di quanto non sia stato mostrato, permettendo la rappresentazione di oggetti di vario tipo, compreso il disegno di curve. Per gestire queste funzionalità senza troppa fatica, sono stati realizzati dei programmi esterni che si occupano di analizzare preventivamente un sorgente Troff, in modo da trasformare alcuni comandi particolari in codice di basso livello adatto a Troff. Il concetto è simile

a quello del precompilatore del linguaggio C, con il quale, attraverso istruzioni apposite, si genera un sorgente specifico prima della compilazione vera e propria.

I programmi di pre-elaborazione più comuni per quanto riguarda Troff sono: Tbl, Eqn e Pic. Il primo è specializzato nella preparazione di tabelle, il secondo serve a facilitare la scrittura di equazioni e il terzo facilita il disegno di curve. In generale, un sorgente Troff che contenga simultaneamente tabelle, equazioni e disegni, andrebbe analizzato attraverso un condotto simile a quello seguente:

```
cat file_troff | tbl | eqn | pic | troff | ...
```

Qui viene mostrato solo qualche esempio dell'uso di tabelle ed equazioni; alla fine del capitolo si trovano i riferimenti per approfondire l'uso di Troff e di questi programmi aggiuntivi.

25.3.1 Tbl

«

Tbl filtra un file Troff alla ricerca di tabelle delimitate dalle macro `.TS` e `.TE`; se ne trova, trasforma la descrizione di queste in qualcosa di adatto a Troff. In modo semplificato, si può rappresentare la struttura di una tabella di Tbl nel modo seguente:

```
.TS
[ opzioni ; ]
formato_celle .
contenuto_celle
.TE
```

Le opzioni sono rappresentate da una serie di parole chiave, facoltative, che descrivono la tabella in modo complessivo, terminate alla fine da un punto e virgola. Se si utilizzano, queste parole sono separate da uno spazio e probabilmente devono apparire sulla stessa riga del sorgente.

Il formato delle celle è un elenco di simboli composti da una sola lettera che servono a indicare l'allineamento del testo contenuto al loro interno. Utilizzano più righe, una per ogni riga della tabella finale, dove in particolare l'ultima definizione riguarda tutte le righe rimanenti della tabella.

Il contenuto della tabella viene scritto separando gli elementi di ogni riga attraverso un carattere di tabulazione.

La descrizione non viene approfondita ulteriormente. Gli esempi dovrebbero rendere l'idea del funzionamento di queste tabelle, il cui uso può essere appreso con maggiore dettaglio leggendo la documentazione indicata alla fine del capitolo.

Si suppone di voler realizzare una tabella simile allo schema seguente:

| Intestazione | |
|--------------------|---------------|
| Nominativo | Telefono |
| Tizio Tizi | 0987,654321 |
| Caio Cai | 0876,543210 |
| Sempronio Semproni | 0765,43210123 |

Questa tabella si può rappresentare attraverso Tbl nel modo seguente:

```
.TS
allbox;
c s
c c
l l.
Intestazione
Nominativo Telefono
Tizio Tizi 0987,654321
Caio Cai 0876,543210
Sempronio Semproni 0765,43210123
.TE
```

Lo stesso risultato avrebbe potuto essere ottenuto sostituendo la parola chiave `'allbox'`, che serve a incasellare ogni cella, con `'box'` che crea solo una cornice esterna, richiedendo esplicitamente l'inserimento delle linee verticali e orizzontali.

```
.TS
box;
c s
c|c
l|l.
Intestazione
-
Nominativo Telefono
-
Tizio Tizi 0987,654321
-
Caio Cai 0876,543210
-
Sempronio Semproni 0765,43210123
.TE
```

Così, si può decidere di modificare la tabella nello schema seguente che alterna l'uso delle separazioni orizzontali.

| Intestazione | |
|--------------------|---------------|
| Nominativo | Telefono |
| Tizio Tizi | 0987,654321 |
| Caio Cai | 0876,543210 |
| Sempronio Semproni | 0765,43210123 |

Per ottenere questo risultato si possono utilizzare le istruzioni seguenti:

```
.TS
box;
c s
c|c
l|l.
Intestazione
=
Nominativo Telefono
-
Tizio Tizi 0987,654321
Caio Cai 0876,543210
Sempronio Semproni 0765,43210123
.TE
```

25.3.2 Eqn

«

Eqn filtra un file Troff alla ricerca di equazioni delimitate dalle macro `.EQ` e `.EN`; se ne trova, trasforma la descrizione di queste in qualcosa di adatto a Troff. In modo semplificato, si può rappresentare la struttura di un'equazione nel modo seguente:

```
.EQ
equazione
.EN
```

Anche la sintassi particolare di Eqn viene omessa e si lascia eventualmente al lettore l'onere di procurarsi la documentazione relativa, indicata alla fine del capitolo.

Si suppone di voler realizzare l'equazione dell'interesse semplice:

$$I = Ct \frac{r}{100}$$

Si può ottenere nel modo seguente:

```
.EQ
I = C t r over 100
.EN
```

Un altro esempio con valori all'esponente:

$$f(x) = x^2$$

La trasformazione attraverso la sintassi di Eqn:

```
.EQ
f(x) = x sup 2
.EN
```

25.4 Groff

Groff è la realizzazione GNU dei programmi *roff. I nomi dei programmi tradizionali sono stati mantenuti, eventualmente attraverso dei collegamenti, quindi si trovano gli eseguibili **'troff'**, **'tbl'**, **'eqn'**, **'pic'**, oltre a uno script **'nroff'** che emula il comportamento di quel programma. A fianco di questo si aggiungono in particolare: **'groff'**, un programma che facilita l'uso di **'troff'** e di ciò che serve a ottenere il formato finale prescelto; inoltre, **'gtbl'**, **'geqn'** e **'gpic'** che rappresentano semplicemente dei nomi alternativi a quelli tradizionali usati per Tbl, Eqn e Pic.

Groff si compone di una serie di programmi in grado di trasformare quanto generato da Troff nel formato finale prescelto. Si tratta principalmente di **'grotty'**, **'groddvi'** e **'groeps'**, necessari rispettivamente per ottenere un testo adatto allo schermo di un terminale, un file DVI e un file PostScript. Questi ricevono un file dallo standard input, oppure leggono quelli indicati negli argomenti e li trasformano conseguentemente. In pratica, vengono usati attraverso dei condotti come negli schemi seguenti:

```
troff -Tlatin1 [altre_opzioni] [file_troff...] | grotty > file_tty
```

```
troff -Tutf8 [altre_opzioni] [file_troff...] | grotty > file_tty
```

```
troff -Tdvi [altre_opzioni] [file_troff...] | groddvi > file_dvi
```

```
troff -Tps [altre_opzioni] [file_troff...] | groeps > file_ps
```

Groff include il precompilatore omonimo, **'groff'**, che permette di semplificare tutto questo nel modo seguente:

```
groff -Tlatin1 [altre_opzioni] [file_troff...] > file_tty
```

```
groff -Tutf8 [altre_opzioni] [file_troff...] > file_tty
```

```
groff -Tdvi [altre_opzioni] [file_troff...] > file_dvi
```

```
groff -Tps [altre_opzioni] [file_troff...] > file_ps
```

Per ottenere questo, **'groff'** accetta quasi tutte le opzioni di **'troff'**, a cui poi provvede a passarle. **'groff'** si occupa anche di richiamare la pre-elaborazione da parte di programmi come **'tbl'**, **'eqn'** e **'pic'**, semplificando quindi la scrittura di condotti che eventualmente possono diventare molto complessi.

```
groff [opzioni] [file...]
```

'groff' è il programma frontale del pacchetto GNU omonimo. Attraverso questo è possibile comandare la definizione automatica dei condotti che tradizionalmente servivano per ottenere la composizio-

ne di un sorgente Troff/Nroff. A questo proposito, molte delle opzioni di **'groff'** sono le stesse che andrebbero fornite direttamente al programma **'troff'**. Segue la descrizione di alcune di queste opzioni.

| Opzione | Descrizione |
|---------------------|---|
| -e | |
| -t | |
| -p | Filtra i file utilizzando rispettivamente: 'eqn' , 'tbl' , 'pic' o 'soelim' . |
| -s | |
| -Tsigla_dispositivo | Definisce il tipo di composizione che deve essere eseguito specificando una sigla adatta. |
| -Tps | Compone in formato PostScript. |
| -Tdvi | Compone in formato DVI. |
| -Tascii | Compone in formato testo senza lettere accentate. |
| -Tlatin1 | Compone in formato testo utilizzando la codifica ISO 8859-1. |
| -Tutf8 | Compone in formato testo utilizzando la codifica UTF-8. |
| -Thtml | Compone in HTML. |
| -Tlj4 | Compone secondo il linguaggio di stampa PCL5 (HP Laserjet e simili). |
| -a | |
| -i | |
| -nn_iniziale | |
| -oelenco_pagine | |
| -rregistronumero | |
| -mnome | |
| -Epercorso | |
| | Queste e altre opzioni hanno lo stesso significato di quelle corrispondenti usate per Troff. |

Segue la descrizione di alcuni esempi.

```
$ groff -Tps -ms mio.troff > mio.ps [Invio]
```

Elabora il file **'mio.troff'** generando il file **'mio.ps'** in formato PostScript. In particolare, fa uso del pacchetto di macro **'s'**.

```
$ groff -t -Tps -ms mio.troff > mio.ps [Invio]
```

Come nell'esempio precedente, utilizzando Tbl per la pre-elaborazione delle tabelle.

```
$ groff -Tlatin1 -ms -o4,6,8-10 mio.troff > mio.tty [Invio]
```

Elabora il file **'mio.troff'** generando il file **'mio.tty'** in un formato adatto alla visualizzazione attraverso un terminale a caratteri, accettando la codifica ISO 8859-1, selezionando le pagine 4, 6, 8, 9 e 10.

```
$ groff -Tutf8 -ms -o4,6,8-10 mio.troff > mio.tty [Invio]
```

Come nell'esempio precedente, generando un file in un formato adatto alla visualizzazione attraverso un terminale a caratteri che richiede la codifica UTF-8.

25.5 Documentazione Man

La documentazione interna tradizionale di Unix è scritta utilizzando comandi di composizione di Troff, facendo uso, in particolare, di un pacchetto di macro specifico, più o meno standardizzato tra i vari sistemi: **'an'**. In pratica, per comporre un file delle pagine di manuale di GNU/Linux o di un altro sistema Unix, occorre usare Troff con l'opzione **'-man'**.

Groff, in particolare, fornisce anche un altro pacchetto di macro che dovrebbe essere compatibile con il formato utilizzato da una vecchia versione di BSD: `'doc'`. Inoltre, è possibile risolvere questi problemi di compatibilità in modo automatico attraverso il pacchetto di macro `'andoc'`, che in pratica è richiamato con l'opzione `'-mandoc'`.

Ogni sistema Unix ha probabilmente il suo stile tipografico particolare per la redazione delle pagine di manuale e questa informazione dovrebbe essere contenuta all'interno di `man(7)` oppure `man(5)`. Le macro del pacchetto `'an'` secondo Groff sono descritte nel seguito.

| Opzione | Descrizione |
|--|--|
| <code>.TH nome n_sezione</code> ← →[<i>data origine titolo_documento</i>] | Il file sorgente di una pagina di manuale deve iniziare con la macro <code>' .TH'</code> (<i>Title header</i>), che serve a definire il titolo, l'intestazione e il piè pagina del documento. In particolare, è il caso di sottolineare il fatto che in generale, in sistemi Unix diversi da GNU/Linux, potrebbero essere previsti solo i primi due argomenti, cioè il nome della pagina di manuale e il numero di sezione. |
| <code>.SH titolo_sezione</code> | Dopo il preambolo costituito dalla macro <code>' .TH'</code> , il testo del documento è suddiviso in sezioni, introdotte dalla macro <code>' .SH'</code> (<i>Section header</i>). La prima di queste è denominata convenzionalmente <code>'NAME'</code> , o <code>'NOME'</code> nelle edizioni italiane. |
| <code>.SS titolo_sottosezione</code> | Le sezioni possono articolarsi in sottosezioni, attraverso questa macro che permette di indicarne il titolo. |
| <code>.LP</code> <code>.PP</code> | Queste due macro sono equivalenti e servono a introdurre un paragrafo. Data la loro natura, introducono automaticamente un'interruzione di riga (<i>break</i>). |
| <code>.B testo_in_neretto</code> <code>.I testo_in_corsivo</code> <code>.SM testo_in_piccolo</code> | Rende il testo posto come argomento della macro in neretto (<code>' .B'</code>), in corsivo (<code>' .I'</code>), o in piccolo (<code>' .SM'</code>). Il testo che appare nelle righe successive non è coinvolto da queste macro. |
| <code>.BI testo_in_neretto testo_in_corsivo ...</code> <code>.BR testo_in_neretto testo_in_tondo ...</code> <code>.IB testo_in_corsivo testo_in_neretto ...</code> <code>.IR testo_in_corsivo testo_in_tondo ...</code> <code>.RB testo_in_tondo testo_in_neretto ...</code> <code>.RI testo_in_tondo testo_in_corsivo ...</code> <code>.SB testo_in_piccolo testo_in_neretto ...</code> | Si tratta di macro particolari che rendono il testo fornito come argomento in modo alternato. Gli argomenti vengono uniti assieme. <code>' .BI'</code> alterna il neretto e il corsivo; <code>' .BR'</code> alterna il neretto e il tondo; <code>' .IB'</code> alterna il corsivo e il neretto; <code>' .IR'</code> alterna il corsivo e il tondo; <code>' .RB'</code> alterna il tondo e il neretto; <code>' .RI'</code> alterna il tondo e il corsivo; <code>' .SB'</code> alterna il piccolo al neretto. |
| <code>.DT</code> | Ripristina le tabulazioni normali. |
| <code>.HP</code> | Inizia un paragrafo in cui le righe successive alla prima sono rientrate. |
| <code>.IP etichetta</code> | Inizia un paragrafo di un elenco descrittivo, in cui l'etichetta è l'argomento della macro. |
| <code>.TP</code> | Inizia un paragrafo di un elenco descrittivo, in cui l'etichetta è la prima riga di testo che segue la macro nel sorgente. |
| <code>.RS</code> <code>.RE</code> | Queste due macro servono a circoscrivere un paragrafo rientrato: <code>' .RS'</code> inizia il testo rientrato, <code>' .RE'</code> termina il blocco. |

Quello che segue è l'esempio di un sorgente di una pagina di manuale scritta secondo le modalità previste per la documentazione di GNU/Linux.

```
.TH ARCH 1 "20 Dicembre 1993" "Linux 0.99" "Linux Programmer's Manual"
.SH NOME
arch \- stampa l'architettura della macchina
.SH SINTASSI
.B arch
.SH DESCRIZIONE
.B arch
è equivalente a
.B uname -m

Sugli attuali sistemi Linux,
.B arch
stampa "i386" o "i486".
.SH VEDERE ANCHE
.BR uname (1), "uname" (2)
```

Dall'esempio mostrato, si possono osservare alcune parti. All'inizio, il titolo e l'intestazione del documento contiene alcuni argomenti delimitati tra virgolette doppie, per poter includere gli spazi.

```
.TH ARCH 1 "20 Dicembre 1993" "Linux 0.99" "Linux Programmer's Manual"
```

In pratica, si tratta del documento `arch(1)`. Alla fine del sorgente mostrato, si vede l'uso della macro `' .BR'`, che è una di quelle che uniscono gli argomenti alternandone il tipo di enfaticizzazione. In questo caso, il neretto si alterna al carattere tondo normale, in modo da evidenziare le parole `'uname'` lasciando che le sezioni vengano rese attraverso il tondo normale. È importante osservare anche l'uso delle virgolette che permette di inserire uno spazio prima del secondo `'uname'`. Volendo, quella riga avrebbe potuto essere scritta nel modo seguente:

```
.BR uname (1), " " "uname" (2)
```

In questo modo, la stringa nulla verrebbe resa in neretto e la stringa contenente uno spazio verrebbe resa con un carattere tondo normale.

25.6 Riferimenti

- Brian W. Kernighan, *A TROFF Tutorial*, 1987, <http://www.kohala.com/start/troff/v7man/troffut/troffut.ps>
- Joseph F. Ossanna, Brian W. Kernighan, Gunnar Ritter, *Heirloom Documentation Tools, Nroff/Troff User's Manual*, <http://heirloom.sourceforge.net/doctools/troff.pdf>
- L. L. Cherry, M. E. Lesk, *Tbl --- A Program to Format Tables*, Bell Laboratories, 1976, <http://www.cs.bell-labs.com/10th10EdMan/tbl.pdf>
- Brian W. Kernighan, Lorinda L. Cherry, *Typesetting Mathematics --- User's Guide*, 1978, <http://www.cs.bell-labs.com/cm/cs/doc/74/eqn.ps.gz>

¹ Nel caso di Troff GNU, installato su GNU/Linux, il file si potrebbe trovare nella directory `'/usr/lib/groff/tmac/'`.

² Qui non vengono mostrati esempi per la definizione del piè di pagina, dal momento che il problema richiede uno sforzo aggiuntivo non giustificabile in questo contesto introduttivo di Troff.

