EMACS RIDOTTO ALL'OSSO

0. A chi sono destinate queste note

Queste note sono destinate agli studenti del primo modulo del laboratorio di Esperimentazioni di Fisica 1, corso di laurea in Fisica, Università di Milano Bicocca. Tuttavia gran parte del contenuto di queste note riguarda semplicemente l'uso di Emacs, indipendentemente dall'installazione, e quindi può essere utile anche ad altri. Le informazioni in queste note sono volutamente ridotte al minimo. Per chi volesse conoscere i dettagli dei comandi qui suggeriti o di altri, esiste l'help in linea: sulla barra dei comandi di Emacs c' è la voce "Help". Basta un clicche, e tanta fantasia (sbagliando s' impara; o si muore).
Nota: Caro lettore, non ti offendere se nel resto di queste note mi prenderò qualche volta la libertà di darti del tu. Lungi da me l'idea di farti sentire stupido. È solo la traduzione brutale dello "you" inglese. O, se preferisci, è solo per non intimidirti troppo. A farti sentire stupido basta già la macchina.

1. Notazioni essenziali

La maggior parte dei comandi EMACS prevede l'uso di caratteri composti con tasti funzione.

- `S-` : funzione "`SHIFT`"
- `C-` : funzione "`CONTROL`"
- `M-` : funzione "`ALT`"

Il funzionamento è analogo al "maiuscolo" sulla macchina da scrivere: per battere il carattere maiuscolo "A" si preme il tasto "`a`" tenendo premuto contemporamente il tasto "`SHIFT`". Questa operazione viene indicata con "`S-a`". Analogamente, "`C-a`" significa premere il tasto "`a`" tenendo premuto contemporaneamente il tasto "`CONTROL`", e `M-a` significa premere contemporaneamente i tasti "`ALT`" e "`a`".

Sono possibili composizioni più complesse, del tipo `M-S-a` : premere contemporaneamente i tre tasti "`ALT`", "`SHIFT`" ed "`a`". Per brevità, quest'ultima operazione viene talvolta indicata semplicemente con "`M-A`" (la "A" maiuscola richiede già il tasto "`SHIFT`"). Il massimo della complicazione è "`C-M-S-a`": si devono premere contemporaneamete i tasti "`CONTROL`", "`ALT`", "`SHIFT`" e "`a`". Più facile da dire che da fare. Aver frequentato con profitto un corso di pianoforte può essere molto utile (esercizio: battere la successione di comandi `C-x C-f` servendosi solo della mano sinistra; in altre parole, tenere premuto il tasto "`CONTROL`" col dito mignolo della mano sinistra, e battere in successione il tasto "`x`" con il dito medio ed il tasto "`f`" col dito indice della stessa mano).

Nota 1: Le notazioni assurde che uso qui non sono frutto del mio sadismo: sono quelle usate nel manuale di Emacs.

Nota 2: Molte operazioni si possono eseguire anche servendosi del mouse e dei menu che compaiono nella striscia in alto della finestra Emacs. Funzionano come tutti i menu: c' è l'elenco delle operazioni che puoi fare, ma si guardano bene dal darti qualunque spiegazione che ti metta in grado di usarli. L'uso dei menu non verrà spiegato in queste note. In caso di necessità, potrai:

(a) metterci un po' di fantasia e di gusto del rischio;

(b) passare al menu “help”, e cercare di capire mettendoci una robusta dose di pazienza; può essere utile la conoscenza del computerese;
(c) chiedere appoggio morale e consiglio a qualche amico (sufficientemente paziente) che abbia già sbagliato prima di te. Ti prego di escludere dalla lista l'autore di queste note (che ha già fatto la sua parte).

2. Procedura elementare

(a) Apertura di una sessione di editing: si batte il comando
`emacs` <*nome–file*> `&`
(il carattere “`&`” serve solo per aprire Emacs in una finestra indipendente); compare la finestra di lavoro Emacs con il contenuto del file da correggere, oppure una finestra vuota se il file non esiste.
(b) Si inserisce un nuovo testo, o si modifica l'esistente: a livello veramente elementare basta usare le frecce per posizionare il cursore, il tasto “`BACKSPACE`” o “`DELETE`” per cancellare il carattere immediatamente prima del cursore, il tasto “`RETURN`” per andare a capo. Quello che si fa con qualunque text editor di tipo “full–screen”.
(c) Si memorizza il file modificato battendo in successione `C-x C-s` (`CONTROL-x` seguito da `CONTROL-s`).
(d) Si esce da Emacs battendo in successione `C-x C-c`. La finestra Emacs sparisce. Può accadere che vi siano files aperti e modificati, ma non ancora salvati; in tal caso Emacs ti chiederà se vuoi salvarli. Basta rispondere “y” (yes… non dimenticare che siamo una colonia americana, e ringrazia il cielo che i giapponesi fanno computers che parlano in americano) o “n” (no). In realtà, Emacs è un po' petulante: se rispondi “n” ti chiederà ancora se vuoi proprio uscire senza salvare le aree modificate; per giunta, a questa domanda dovrai proprio rispondere “yes” o “no”, scritto per intero. Abbi pazienza: ti tratta da stupido, ma lo fa per il tuo bene…

Nota: Questo è sufficiente per lavorare in modo beluinamente elementare (o elementarmente beluino). Quando sarai stanco di posizionarti nel testo solo usando le frecce, di cercare un testo ad occhio (con conseguente strabismo serale), di aspettare che Emacs si decida a partire ogni volta che lo attivi, ecc…, forse ti verrà voglia di leggere anche i paragrafi seguenti. In attesa di quel momento puoi anche fermarti qui.

3. Altri comandi per gestire il testo

- Comandi di spostamento del cursore:
  - `C-e` si sposta alla fine della riga corrente;
  - `C-a` si sposta all'inizio della riga corrente;
  - `C-l` porta la riga corrente al centro della finestra (il carattere “`l`” minuscolo, non 1; se hai provato a battere un testo più lungo di una pagina avrai notato che la finestra di Emacs si sposta a scatti, e ti è difficile centrarla secondo i tuoi desideri: `C-l` serve proprio a questo);
  - `C-v` si sposta avanti di una pagina (prova anche col tasto “PgDn”);
  - `M-v` si sposta indietro di una pagina (prova anche col tasto “PgUp”).
- Comandi di cancellazione e ripristino:

`C-k` cancella dal cursore fino alla fine della riga; un secondo `C-k` cancella anche il ritorno a capo;

`C-y` reinserisce l'ultimo blocco di righe cancellate (il blocco è l'insieme delle righe che sono state cancellate premendo più volte `C-k`, in successione; qualunque tasto diverso da `C-k` interrompe la sequenza e termina il blocco).

- Taglia e cuci:
  (a) posizionare il cursore ad un estremo del testo da tagliare o copiare;
  (b) battere `C-`<*spazio*> (`CONTROL` + la barra spaziatrice);
  (c) spostare il cursore all'altro estremo del testo da copiare o tagliare;
  (d) battendo `C-w` si taglia il testo selezionato (che viene rimosso);
  (e) spostare il cursore nella posizione dove si vuole inserire il testo;
  (f) battendo `C-y` l'ultimo testo rimosso viene inserito nella posizione del cursore.

  Il medesimo testo si può duplicare quante volte si vuole.
- Ancora sul taglia e cuci:
  è uno dei (non molti) casi in cui usare il mouse è effettivamente vantaggioso. Si procede così:
  (a) porta il cursore del mouse ad un estremo della parte da tagliare o copiare, e fa un clicche col tasto sinistro del mouse.
  (b) porta il cursore del mouse all'altro estremo del testo da copiare, e fa un clicche col tasto destro. Vedrai il testo selezionato visualizzato su sfondo diverso dal normale.
  (c) se vuoi tagliare il testo fa un secondo clicche col tasto destro del mouse, senza spostare il cursore. La parte selezionata sparirà.
  (d) sposta il cursore del mouse nel punto di inserimento, e fa un clicche col tasto centrale del mouse (se il mouse ha solo due tasti premili ambedue contemporaneamente; se non succede nulla chiedi ad un esperto come si fa ad abilitare la simulazione del terzo tasto nel file `/etc/X11/XF86Config`). Vedrai il testo inserito nel punto indicato *dal cursore del mouse*.

  Nota: ho specificato puntualmente *il cursore del mouse*. Non è (solo) pignoleria: sullo schermo ci sono due cursori: quello che si riferisce alla tastiera, e quello del mouse. Per l' operazione qui sopra la posizione del cursore della tastiera è irrilevante. Ma non preoccuparti più di tanto: dopo i primi 30 sbagli imparano tutti.
- Ricerca di un testo:
  - il comando `C-s` inizia la ricerca; tutto quello che si batte dopo `C-s` specifica il testo da cercare; la ricerca finisce solo quando si batte un tasto di comando (ad esempio, uno spostamento con una freccia). Al primo utilizzo il comportamento della ricerca sembra un po' bizzarro. Ti consiglio di provare una prima volta osservando il comportamento del cursore ad ogni tasto che batti: vedrai che la posizione viene aggiornata continuamente. Ad esempio, se dopo `C-s` batti "b" il cursore si posiziona

sulla prima "b" che trova; se aggiungi "o" il cursore si sposta ancora fino alla prima sequenza "bo"; se aggiungi "h" il cursore si sposta su "boh", e così via. Se cerchi una sequenza di caratteri che non esiste Emacs si mette a fischiare per protesta. Non dimenticare di concludere la ricerca con un tasto di comando.

  - se si conclude la ricerca con un altro `C-s`, Emacs abbandona la sequenza trovata e cerca la successiva. Quando arriva alla fine del testo protesta per una volta, ma basta ripetere `C-s` per farlo ricominciare dall'inizio.
  - un doppio `C-s` significa che si vuole cercare di nuovo l'ultima sequenza di caratteri specificata.
  - il tasto `C-r` funziona come `C-s`, salvo il fatto che la ricerca viene effettuata all'indietro.

- Quando vorresti dimenticare ...:

  `C-S-_` (`CONTROL-SHIFT`–<*underline*>; il carattere <*underline*> si trova di solito sopra il segno -) cancella l'effetto dell'ultimo comando eseguito. Battendo `C-S-_` più volte in successione si possono annullare diversi comandi. Provare per credere. Su qualche sistema la stessa funzione viene svolta anche dal comando `C-/`, che richiede qualche acrobazia in meno per le dita.

  `C-g` annulla un comando iniziato e non ancora terminato (ad esempio, se batti un `C-s` per sbaglio la ricerca viene annullata da un `C-g`. Qualche volta Emacs non capisce bene: di `C-g` bisogna dargliene due. In ogni caso, uno di troppo non fa danni.

4. Uso di diversi buffers

Emacs è in grado di gestire contemporaneamente diversi files. È comodo tenere una sessione di Emacs permanentemente aperta, ed aprire via via i files che servono. Ogni file viene memorizzato separatamente in un'area di memoria che viene denominata "buffer". Al buffer viene assegnato un nome che coincide col nome del file. In basso alla finestra si vede una riga di stato (la penultima riga) che inizia solitamente con una scritta del tipo

```
-----Emacs:  ciccio.mio
```

oppure

```
--**-Emacs:  ciccio.mio
```

In questo caso, "ciccio.mio" è il nome del buffer, e quindi anche del file, su cui si sta lavorando. I due asterischi significano che il file ha subito modifiche dopo la lettura da disco o dopo l'ultimo salvataggio. L'assenza dei due asterischi indica che il contenuto del buffer è identico a quello del file. Per interpretare il resto della riga basta in qualche caso un po' di fantasia (ad esempio, L113 significa che il cursore si trova sulla linea 113 del testo); in altri casi la spiegazione sarebbe un po' più complessa. Se vuoi saperne di più, divertiti con l'help o con il manuale.

Questo elenco contiene le operazioni elementari sui buffers:

`C-x C-f` (`CONTROL-x` seguito da `CONTROL-f`) crea un nuovo buffer; nella riga più bassa della finestra compare la richiesta del nome del file da aprire. Tipicamente può accadere che il directory di lavoro sia già indicato, come se fosse già stato battuto.

Il nuovo buffer assume il nome del file. Se il file non esiste, alla prima operazione di scrittura (con `C-x C-s`) ne viene creato uno nuovo.

`C-x C-s` salva il contenuto del buffer corrente nel file corrispondente; il buffer non viene eliminato, e può essere modificato ulteriormente. I due asterischi spariscono (il contenuto del buffer è identico a quello del file), ma ricompaiono alla prima modifica.

`C-x C-w` salva il buffer in un file di nome diverso da quello originale; la richiesta del nuovo nome compare nell'ultima riga. A partire da questo momento il buffer cambia nome, e viene agganciato al nuovo file (tutte le operazioni di salvataggio successive finiranno sul nuovo file).

`C-x i` copia il contenuto di un file nel buffer corrente (diverso da `C-x C-f` perché non crea un nuovo buffer).

`C-x k` cancella (kill) un buffer. Se il buffer contiene dati modificati e non ancora salvati Emacs ti chiede se vuoi che vengano salvati prima di sopprimere il buffer; rispondi con "y" o "n".

`C-x b` passa su un altro buffer esistente. Se non sai quali buffers hai già creato leggi il prossimo comando.

`C-x C-b` visualizza l'elenco dei buffers esistenti in una mezza finestra separata. Per portare il cursore nella seconda metà della finestra basta un "click" col mouse; per tornare ad una finestra sola portare il cursore nella parte di finestra che si vuol mantenere e battere la sequenza `C-x 1`.

Nota: per passare da un buffer all'altro è più comodo servirsi del menu "Buffers" e del mouse. Clicca tu che clicco io . . . alla fine qualcosa succede.

5. Uso di finestre multiple

La finestra corrente può essere divisa in due con una delle sequenze

`C-x 2` divide la finestra in senso orizzontale;

`C-x 3` divide la finestra in senso verticale;

`C-x 1` elimina tutte le finestre tranne quella che contiene il cursore.

Ciascuna delle due parti ha una riga di stato, e può contenere un suo buffer. Se ambedue le parti contengono lo stesso buffer, ogni modifica apportata su una delle due viene automaticamente applicata anche all'altra.

L'uso di finestre multiple è comodo quando si debba lavorare in contemporanea su due buffers diversi o su parti diverse dello stesso buffer. Ad esempio si possono effettuare facilmente confronti, trasferimenti o copie di testo da un buffer all'altro. Il passaggio da una finestra all'altra si effettua con un semplice "click" col mouse.

6. Comandi in forma esplicita

Tutti i comandi Emacs hanno una forma esplicita, in linguaggio simile all'inglese (beh, . . . all'americano . . .) corrente. In effetti, i tasti funzione sono solo delle forme abbreviate per i comandi (ritenuti da qualcuno) più comuni.

Nella maggior parte dei casi un comando è costituito dal prefisso `M-x` seguito da una specifica. Appena si batte `M-x` Emacs chiede il resto del comando nell'ultima linea della finestra. Ogni comando viene terminato dal tasto “`RETURN`”. Riporto solo due esempi di comandi che mi sembrano particolarmente utili. A te il divertimento di scoprirne altri.

- `M-x auto-fill-mode` : abilita la funzione di “a capo” automatico quando la riga corrente diventa troppo lunga. Comodo per battere testi senza doversi preoccupare di andare esplicitamente a capo. (Un paragrafo disallineato in seguito a correzioni si può riallineare posizionando il cursore su una riga qualunque del paragrafo interessato e battendo il comando `M-q`).
- `M-x goto-line` : salta alla linea `xxx`; batti il tasto “`RETURN`” immediatamente dopo `goto-line`, ed Emacs ti chiederà il numero di linea.

7. Per saperne di più

L’elenco qui sopra non esaurisce i comandi di Emacs, ovviamente. Una versione relativamente recente del manuale ne contiene circa 500, ma credo che nel frattempo siano aumentati.

Se nella tua vita hai altri scopi, oltre a quello di saper usare Emacs, i comandi elencati qui sopra dovrebbero coprire la maggior parte delle tue esigenze. Per operazioni particolari ti basterà probabilmente chiedere a qualcuno più curioso di te.

Se qualche volta ti capita di avere qualche curiosità, prova a giocare con l’help, o cerca in rete un manuale Emacs e prova a leggerlo (è un buon esercizio di computerese, grosso modo del quarto grado; per il sesto grado, cerca di imparare il LISP). In particolare, può essere interessante scoprire

- come definire della macro istruzioni;
- come personalizzare Emacs definendo nuovi tasti funzione adattati alle tue necessità (o idiosincrasie);
- come sfruttare le capacità di Emacs di lavorare su testi di tipo diverso (ad esempio, TeX, o linguaggi) rendendo automatiche certe operazioni di impaginazione.

Se sei un informatico (e nonostante questo sei arrivato a leggere fin qui), studiati il manuale a memoria, e sappi che Emacs può diventare un ambiente globale di lavoro nel quale potrai: spedire e ricevere posta (lettere d’amore comprese); scrivere, compilare, correggere, eseguire e mettere a punto programmi in (quasi) qualunque linguaggio conosciuto; spiegare ad Emacs come fare le stesse cose in un linguaggio sconosciuto anche a te; prepararti un caffè con una dose predefinita di zucchero (ma non fare lo schizzinoso sulla qualità del caffè — stile americano); prenotare i biglietti dell’aereo; rinunciare alle tue vacanze in cambio di una seduta psichiatrica con Eliza; e … tante altre cose che avrai il piacere di scoprire tu stesso. Ti chiedo solo un favore: lascia perdere i virus…