

I Metacaratteri della Shell Unix

La shell Unix riconosce alcuni caratteri speciali, chiamati **metacaratteri**, che possono comparire nei comandi.

Quando l'utente invia un comando, la shell lo scandisce alla ricerca di eventuali metacaratteri, che processa in modo speciale.

Una volta processati tutti i metacaratteri, viene eseguito il comando.

Esempio:

```
user> ls *.java
```

```
Albero.java          div.java             ProvaAlbero.java
AreaTriangolo.java  EasyIn.java         ProvaAlbero1.java
AreaTriangolo1.java IntQueue.java
```

Il **metacarattere** `*` all'interno di un pathname è un'**abbreviazione** per un nome di file. Il pathname `*.java` viene espanso dalla shell con tutti i nomi di file che terminano con l'estensione `.java`. Il comando `ls` fornisce quindi la lista di tutti e soli i file con tale estensione.

Abbreviazione del Pathname

I seguenti metacaratteri, chiamati **wildcard** sono usati per **abbreviare** il nome di un file in un pathname:

Metacarattere	Significato
*	stringa di 0 o più caratteri
?	singolo carattere
[]	singolo carattere tra quelli elencati
{ }	stringa tra quelle elencate

Esempi:

```
user> cp /JAVA/Area*.java /JAVA_backup
```

copia tutti i files il cui nome inizia con la stringa Area e termina con l'estensione .java nella directory JAVA_backup.

```
user> ls /dev/tty?
```

```
/dev/ttya /dev/ttyb
```

... esempi

```
user> ls /dev/tty?[234]
```

```
/dev/ttyp2 /dev/ttyp4 /dev/ttyq3 /dev/ttyr2 /dev/ttyr4  
/dev/ttyp3 /dev/ttyq2 /dev/ttyq4 /dev/ttyr3
```

```
user> ls /dev/tty?[2-4]
```

```
/dev/ttyp2 /dev/ttyp4 /dev/ttyq3 /dev/ttyr2 /dev/ttyr4  
/dev/ttyp3 /dev/ttyq2 /dev/ttyq4 /dev/ttyr3
```

```
user> mkdir /user/studenti/rossi/{bin,doc,lib}
```

crea le directory bin, doc, lib .

Il “quoting”

Il meccanismo del **quoting** è utilizzato per inibire l'effetto dei metacaratteri. I metacaratteri a cui è applicato il quoting perdono il loro significato speciale e la shell li tratta come caratteri ordinari.

Ci sono tre meccanismi di quoting:

- il metacarattere di **escape** \ inibisce l'effetto speciale del metacarattere che lo segue:

```
user> cp file file\?  
user> ls file*  
file      file?
```

- tutti i metacaratteri presenti in una stringa racchiusa tra **singoli apici** perdono l'effetto speciale:

```
user> cat 'file*?'  
...
```

- i metacaratteri per l'abbreviazione del pathname presenti in una stringa racchiusa tra **doppi apici** perdono l'effetto speciale (ma non tutti i metacaratteri della shell):

```
user> cat "file*?"  
...
```

Ridirezione dell'I/O

Di default i comandi Unix prendono l'input da **tastiera** (**standard input**) e mandano l'**output** ed eventuali **messaggi di errore** su video (**standard output, error**).

L'input/output in Unix può essere **rediretto** da/verso **file**, utilizzando opportuni metacaratteri:

Metacarattere Significato

>	ridirezione dell'output
>>	ridirezione dell'output (append)
<	ridirezione dell'input
<<	ridirezione dell'input dalla linea di comando ("here document")
2>	ridirezione dei messaggi di errore (bash Linux)

Esempi:

```
user> ls LABSO > temp
```

```
user> more temp
```

```
lezione1.aux lezione1.log lezione1.tex lezione2.dvi lezione2.tex
```

```
lezione1.dvi lezione1.ps lezione2.aux lezione2.log lezione2.tex
```

... esempi

```
user> echo ciao a tutti >file      # ridirezione dell'output
```

```
user> more file
```

```
ciao a tutti
```

```
user> echo ciao a tutti >>file     # ridirezione dell'output (append)
```

```
user> more file
```

```
ciao a tutti
```

```
ciao a tutti
```

Il comando `wc` (**word counter**) fornisce numero di linee, parole, caratteri di un file:

```
user> wc <progetto.txt
```

```
21 42 77
```

```
user> wc <<delim      # here document
```

```
?  queste linee formano il contenuto
```

```
?  del testo
```

```
?  delim
```

```
2   7   44
```

```
user> man -s2 passwd      # ridirezione dei messaggi di errore
```

```
No entry for passwd in section(s) 2 of the manual.
```

```
user> man -s2 passwd 2>temp
```

Pipe

Il metacarattere | (**pipe**) serve per comporre n comandi “in cascata” in modo che l’output di ciascuno sia fornito in input al successivo. L’output dell’ultimo comando è l’output della pipeline.

La sequenza di comandi

```
user> ls /usr/bin > temp
```

```
user> wc -w temp
```

```
459
```

ha lo stesso effetto della pipeline:

```
user> ls /usr/bin | wc -w
```

```
459
```

I comandi `ls` e `wc` sono eseguiti in parallelo: l’output di `ls` è letto da `wc` mano a mano che viene prodotto.

Per mandare in stampa la lista dei files in `/usr/bin`:

```
user> ls /usr/bin | lpr
```

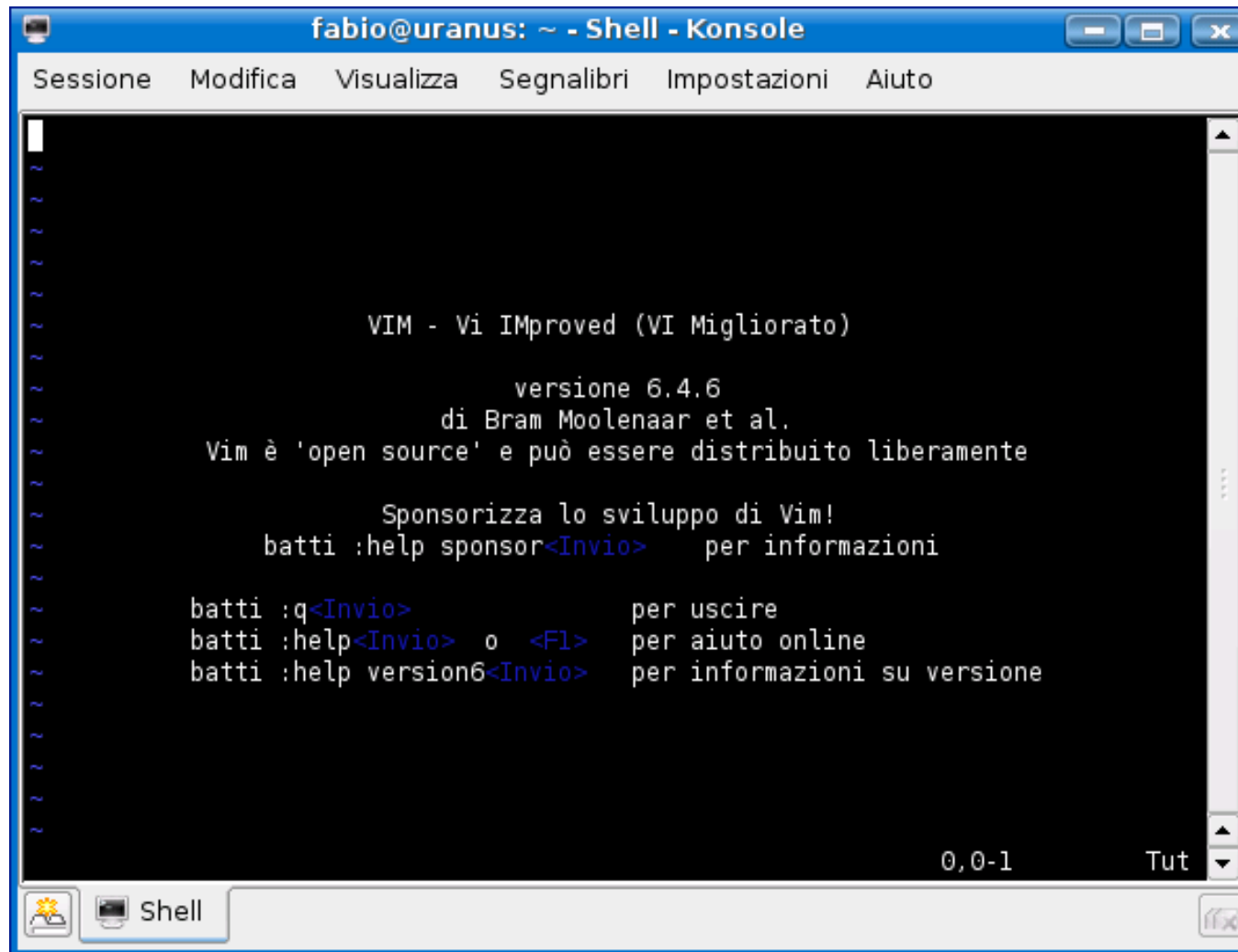
Per visualizzare l’output di `ls` pagina per pagina

```
user> ls | more
```

I'Editor vi

- Negli ambienti Unix esistono molti editor di testo diversi; tuttavia `vi` è l'unico che siamo sicuri di trovare in qualsiasi variante di Unix.
- `vi` (**visual editor**) è stato scritto per essere utilizzabile con qualsiasi tipo di terminale.
- `$ vi filename`
invoca `vi` aprendo il file `filename` (se non esiste, viene creato).
- `vi` ha tre **modalità**:
 1. **edit** mode (all'avvio di `vi` si è in questa modalità),
 2. **insert** mode,
 3. **command** mode.

Interfaccia di vi



The image shows a terminal window titled "fabio@uranus: ~ - Shell - Konsole". The window contains the following text:

```
VIM - Vi IMproved (VI Migliorato)

      versione 6.4.6
      di Bram Moolenaar et al.
Vim è 'open source' e può essere distribuito liberamente

      Sponsorizza lo sviluppo di Vim!
      batti :help sponsor<Invio>   per informazioni

      batti :q<Invio>               per uscire
      batti :help<Invio> o <F1>    per aiuto online
      batti :help version6<Invio> per informazioni su versione
```

At the bottom right of the terminal, it shows "0,0-1" and "Tut". The window has a menu bar with "Sessione", "Modifica", "Visualizza", "Segnalibri", "Impostazioni", and "Aiuto".

Edit mode (I)

La modalità di edit è usata principalmente per muovere il cursore nel punto di interesse all'interno del file di testo che si sta editando.

Comando	Effetto
k, j, h, l (od i tasti cursore)	muove il cursore su, giù, a sinistra ed a destra
Ctrl-f, Ctrl-b	muove il cursore avanti/indietro di una pagina
H, M, L	muove il cursore alla prima riga, all'ultima od a quella nel mezzo dello schermo
w	muove il cursore all'inizio della parola successiva
e	muove il cursore alla fine della parola successiva
b	muove il cursore all'inizio della parola precedente
0	muove il cursore all'inizio della linea corrente
^	muove il cursore nella posizione del primo carattere della linea che non sia un <code>whitespace</code>
\$	muove il cursore alla fine della linea corrente
/string	cerca nel file la stringa <i>string</i>
?string	cerca "all'indietro" nel file la stringa <i>string</i>
n	cerca l'occorrenza della stringa successiva (in avanti o all'indietro)

Edit mode (II)

Comando	Effetto
<i>nrc</i>	rimpiazza <i>n</i> caratteri con <i>c</i> a partire dalla posizione del cursore
<i>nx</i>	cancella <i>n</i> caratteri dalla posizione del cursore
<i>ndw</i>	cancella <i>n</i> parole dalla posizione del cursore
<i>ndb</i>	cancella <i>n</i> parole prima del cursore
<i>ndd</i>	cancella <i>n</i> linee a partire da quella del cursore
d\$	cancella tutti i caratteri dalla posizione del cursore fino alla fine della linea
d0	cancella tutti i caratteri dalla posizione del cursore fino all'inizio della linea
J	unisce la linea corrente alla successiva
p	incolla il testo copiato/cancellato a destra del cursore
P	incolla il testo copiato/cancellato a sinistra del cursore
yy	copia la riga corrente in memoria
<i>nyy</i>	copia <i>n</i> righe in memoria a partire dalla posizione del cursore
u	annulla l'ultimo comando
.	ripete l'ultimo comando
ZZ	termina l'esecuzione di vi, salvando le modifiche

Insert mode

Siccome l'edit mode utilizza un gran numero di tasti alfanumerici, per inserire del testo in un file si rende necessaria un'altra modalità: l'**insert** mode.

Comando	Effetto
i	inserisce del testo alla sinistra del cursore
a	inserisce del testo alla destra del cursore
I	inserisce del testo all'inizio della linea corrente
A	inserisce del testo alla fine della linea corrente
o	inserisce una nuova linea sotto la posizione del cursore
O	inserisce una nuova linea sopra la posizione del cursore

Per uscire dalla modalità di inserimento basta premere il tasto `Esc` o `Ctrl-[` nei terminali senza tasto `Esc`.

Command mode

Tutti i comandi del **command** mode iniziano con i due punti (:); dopo aver inserito tale carattere il cursore si sposta nell'ultima riga dello schermo dove compaiono i caratteri del comando successivamente digitati. La pressione del tasto invio provoca l'esecuzione del comando.

Comando	Effetto
:q	termina vi se non vi sono delle modifiche non salvate
:q!	termina vi perdendo le eventuali modifiche non salvate
:w	salva il file originale
:wq	salva il file originale e termina vi (stesso effetto di ZZ)
:w <i>file</i>	salva il contenuto nel file <i>file</i>
:r <i>file</i>	legge il contenuto del file <i>file</i> inserendolo dopo la posizione del cursore
:e <i>file</i>	edita il file <i>file</i> , sostituendo il contenuto corrente
:f <i>file</i>	cambia il nome del contenuto corrente in <i>file</i>
:f	stampa il nome e lo stato del testo corrente
:n	sposta il cursore alla linea <i>n</i>
:/ <i>str</i> /	sposta il cursore alla prossima linea contenente <i>str</i>
:s/ <i>str1</i> / <i>str2</i> /	sostituisce la prima occorrenza di <i>str1</i> sulla linea con <i>str2</i>
:set <i>option</i>	definisce un'opzione, e.g. :set <i>number</i> aggiunge i numeri di linea

L'editor Emacs/XEmacs

Emacs, E(ditor) Mac(ro)s, è un applicativo che non fa parte di Unix; è stato scritto da R. Stallman nel 1975. Digitando al prompt

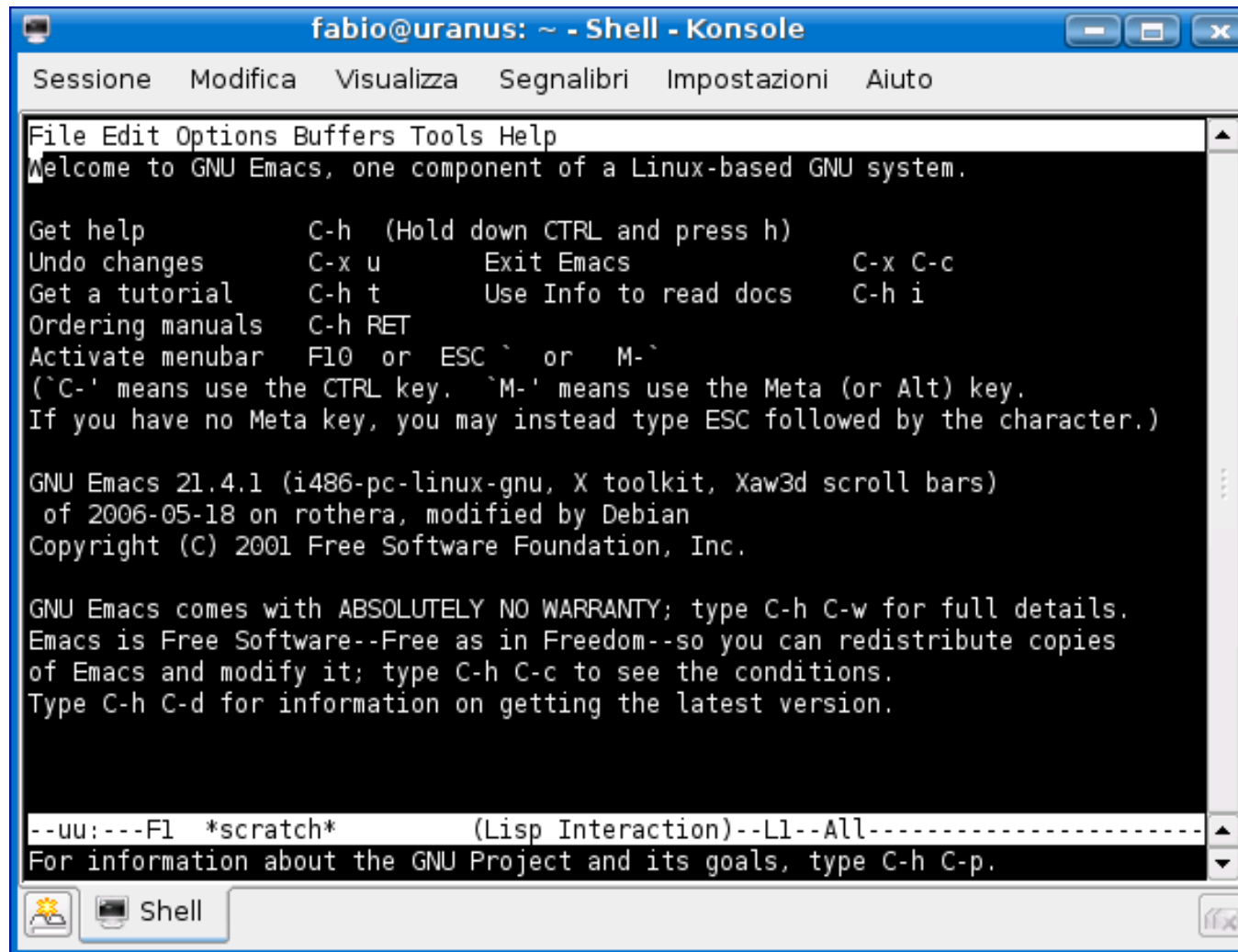
```
$ emacs <file>
```

si apre una finestra in cui viene visualizzato il contenuto del file <file>, che può essere editato normalmente. Tale contenuto è memorizzato in un buffer (struttura dati interna di Emacs).

Ci sono due modi per accedere ai comandi di editing:

1. barra dei menu,
2. **caratteri di controllo.**

Interfaccia di emacs



The image shows a terminal window titled "fabio@uranus: ~ - Shell - Konsole". The window contains the GNU Emacs help screen. At the top, there is a menu bar with "File Edit Options Buffers Tools Help". Below the menu bar, the text reads: "Welcome to GNU Emacs, one component of a Linux-based GNU system." This is followed by a list of commands and their shortcuts: "Get help C-h (Hold down CTRL and press h)", "Undo changes C-x u Exit Emacs C-x C-c", "Get a tutorial C-h t Use Info to read docs C-h i", "Ordering manuals C-h RET", and "Activate menubar F10 or ESC ` or M-`". A note explains that '^C-' means the CTRL key and '^M-' means the Meta (or Alt) key. Below this, the version information is displayed: "GNU Emacs 21.4.1 (i486-pc-linux-gnu, X toolkit, Xaw3d scroll bars) of 2006-05-18 on rothera, modified by Debian Copyright (C) 2001 Free Software Foundation, Inc." A disclaimer follows: "GNU Emacs comes with ABSOLUTELY NO WARRANTY; type C-h C-w for full details. Emacs is Free Software--Free as in Freedom--so you can redistribute copies of Emacs and modify it; type C-h C-c to see the conditions. Type C-h C-d for information on getting the latest version." At the bottom of the terminal, the prompt "--uu:--F1 *scratch* (Lisp Interaction)--L1--All-----" is visible, along with the instruction "For information about the GNU Project and its goals, type C-h C-p."

```
fabio@uranus: ~ - Shell - Konsole
Session Modifica Visualizza Segnalibri Impostazioni Aiuto
File Edit Options Buffers Tools Help
Welcome to GNU Emacs, one component of a Linux-based GNU system.

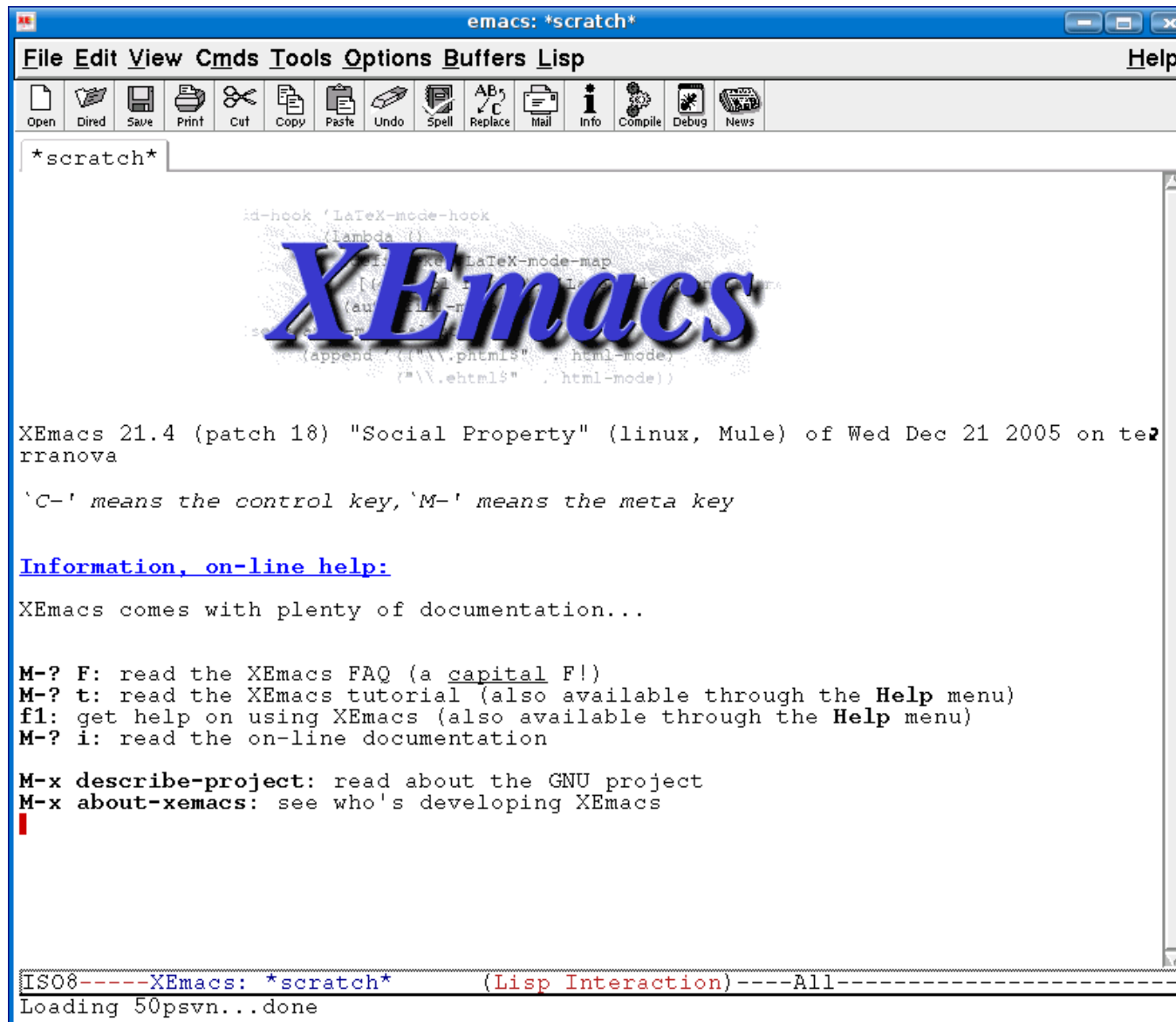
Get help          C-h (Hold down CTRL and press h)
Undo changes      C-x u      Exit Emacs          C-x C-c
Get a tutorial    C-h t      Use Info to read docs  C-h i
Ordering manuals C-h RET
Activate menubar F10 or ESC ` or M-`
(`C-' means use the CTRL key. `M-' means use the Meta (or Alt) key.
If you have no Meta key, you may instead type ESC followed by the character.)

GNU Emacs 21.4.1 (i486-pc-linux-gnu, X toolkit, Xaw3d scroll bars)
  of 2006-05-18 on rothera, modified by Debian
Copyright (C) 2001 Free Software Foundation, Inc.

GNU Emacs comes with ABSOLUTELY NO WARRANTY; type C-h C-w for full details.
Emacs is Free Software--Free as in Freedom--so you can redistribute copies
of Emacs and modify it; type C-h C-c to see the conditions.
Type C-h C-d for information on getting the latest version.

--uu:--F1 *scratch* (Lisp Interaction)--L1--All-----
For information about the GNU Project and its goals, type C-h C-p.
```

Interfaccia di xemacs



Emacs: caratteri di controllo

Ci sono due categorie di caratteri di controllo:

1. sequenze che iniziano con `Ctrl` (abbrev. `C`), e.g., `C-h t` (esegue il tutorial)
2. sequenze che iniziano con meta (tasto `Alt` o `Esc`, abbrev. `M`), e.g., il prefisso `M-x`.

Esistono dei **prefissi** standard di Emacs, ovvero delle sequenze di caratteri, che, se completate opportunamente, costituiscono un comando:

`C-c` `C-x` `C-h` `C-x 4`

Esempio: la finestra principale di Emacs può essere suddivisa in due finestre con la sequenza `C-x 2`. Per ingrandire la finestra su cui si sta lavorando e chiudere l'altra si usa la sequenza `C-x 1`.

Emacs: la mode line

La **mode line** si trova sotto la finestra e fornisce informazioni su:

- lo stato del buffer contenente il file che si sta editando,
- il nome del file,
- la percentuale di file visibile (i.e., che sta nella finestra),
- il **modo**:
 - Fundamental (default)
 - text (testo inglese)
 - Lisp mode
 - C mode
 - ...

Esistono:

- major mode (e.g. text)
- minor mode (e.g. Auto Fill: va a capo automaticamente se una riga è troppo lunga)

Emacs: il minibuffer ed alcuni comandi utili

Sotto la mode line si trova il **minibuffer**, che è una parte dell'interfaccia di Emacs che si occupa di visualizzare e di permettere all'utente di editare/completare i comandi.

Comando	Effetto
M-x text-mode	cambia il major mode in text (modalità testo inglese)
C-h m	informazioni sul modo corrente
C-x C-f <i>file</i>	apre il file <i>file</i>
C-x C-s	salva il buffer corrente
C-<Spacebar>	imposta il mark (inizio di una regione di testo su cui eseguire un comando in seguito)
M-w	copia la regione di testo compresa fra il mark e la posizione del cursore
C-w	taglia la regione di testo compresa fra il mark e la posizione del cursore
C-y	incolla la regione di testo copiata/tagliata in precedenza
C-x u	undo
C-x C-c	termina Emacs

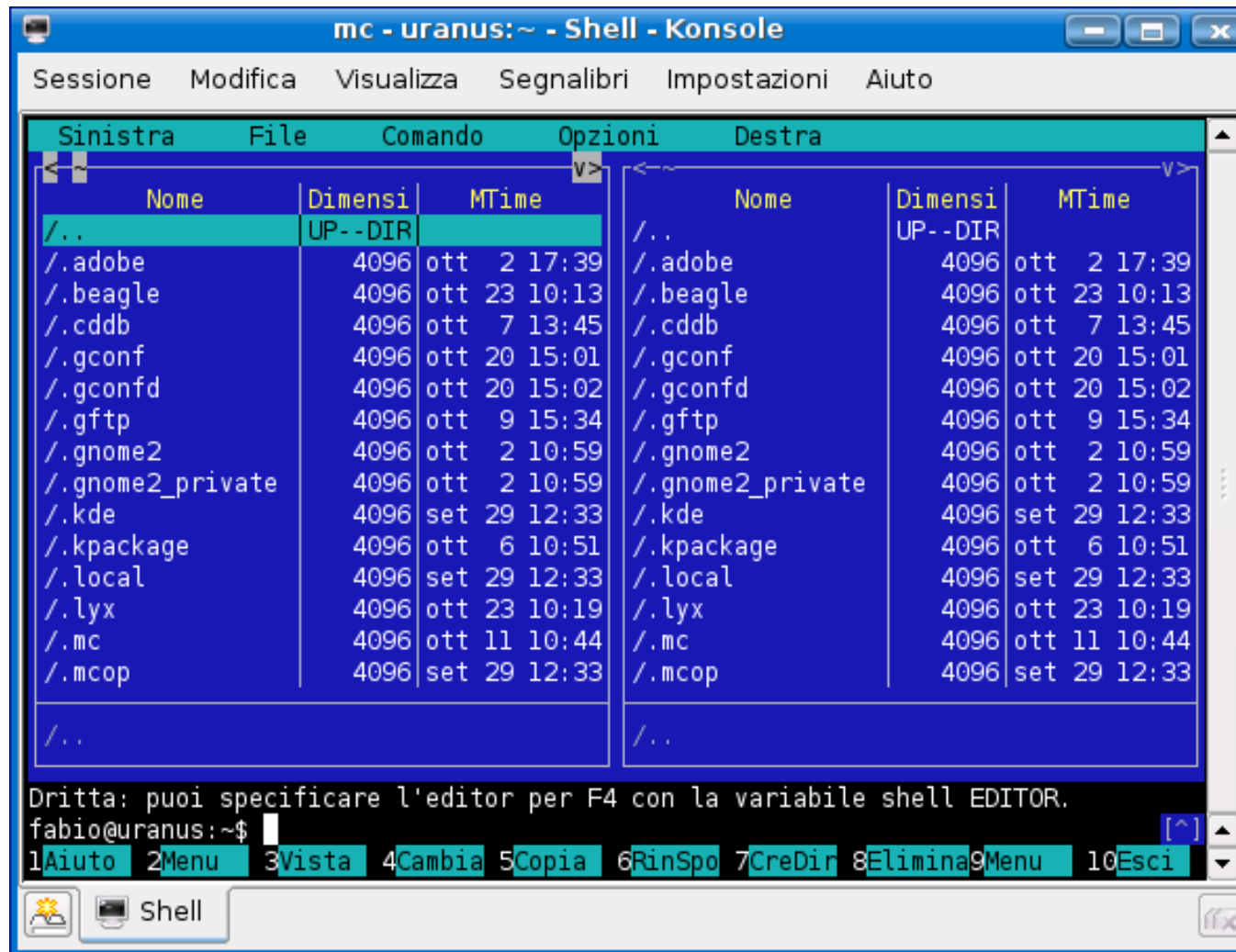
L'alternativa `mc` (Midnight Commander)

Midnight Commander, che si esegue con `mc`, è un browser per l'esplorazione del filesystem da console, che permette anche di visualizzare e modificare i file.

Pur non utilizzando X, fornisce una interfaccia più user-friendly rispetto ad altri editor e permette di:

- navigare tra file e cartelle utilizzando le frecce e il tasto `Invio`;
- visualizzare un file con `F3`;
- modificare un file con `F4`;
- salvare un file con `F2` (in edit mode);
- uscire dall'editor e da `mc` con `F10`.

Interfaccia di mc



Esercizi

- Scrivete un unico comando (pipeline) per
 - copiare la directory `dir1` (con tutto il suo contenuto) nella directory `dir2`;
 - fornire il numero di file (e directory) a cui avete accesso, contenuti ricorsivamente nella directory `/home` (si può utilizzare `ls -R?` e con il comando `find?`);
 - fornire la lista dei file della home directory il cui nome è una stringa di 3 caratteri seguita da una cifra.
- Qual è la differenza tra i seguenti comandi?
`ls`
`ls | cat`
`ls | more`
- Quale effetto producono i seguenti comandi?
 - `uniq < file`, dove `file` è il nome di un file;
 - `who | wc -l ;`

– `ps -e | wc -l .`

- Aprire un file di testo con almeno un editor a scelta tra quelli descritti ed eseguire le seguenti operazioni:
 - inserire una nuova linea di testo in fondo al file;
 - copiare le ultime 4 linee del file all’inizio del file;
 - sostituire tutte le occorrenze della stringa *are* con il carattere -;
 - salvare le modifiche.
- Familiarizzare con almeno uno dei tre editor per avere una buona padronanza delle loro funzioni.