Introduction

to the

GNU Emacs Display Editor

November 1992

This document is an introduction to the GNU Emacs display editor for **timessqr** users.

Revision/Update Information: Version 1.0

Copyright ©1992 George McClintock III

This document is released under the terms of the GNU General Public License as published by the Free Software Foundation, Inc., 675 Mass Ave., Cambridge, MA 02139, USA. Accordingly:

❶ Permission is granted to copy and distribute this document for no commercial gain, provided that the copyright notice and this permission notice are preserved on all copies.

❷ Permission is granted to copy and distribute modified versions of this document for no commercial gain, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

❸ This document is still evolving, thus the information in this document is subject to change without notice. The author of this document cannot be held responsible for any errors that may appear in this document.

For more information about the GNU General Public License, see the file ``COPYING,'' located in **timessqr**'s /usr/users/emacs/etc directory, or write to the Free Software Foundation.

Comments and suggestions for improvements may be forwarded to **mcc@timessqr.gc.cuny.edu**.

DISCLAIMER: The software described in this document is provided ``**as is**''. No guarantee is made by the author as to the suitability, reliability, security, usefulness, or performance of this software.

This document is **not authorized** by the Computer Center, and should not be construed as a commitment by the Graduate School and University Center of the City University of New York, which assumes no responsibility for any errors that may appear in this document.

Table 1 presents trademarks of the Digital Equipment Corporation:

Table 1 DEC Trademarks

DEC	DECnet	ULTRIX
VAX	VAXcluster	VMS
DECsystem	DECwindows	VT

Contents

PREFACE v

CHAPTER 1 THE GNU EMACS DISPLAY EDITOR 1–1

1.1 INTRODUCTION 1–1
1.1.1 Emacs Conventions 1–1

1.2 GETTING STARTED 1–1

CHAPTER 2 EMACS COMMANDS 2–1

2.1 THE .EMACS FILE 2–1

2.2 EDIT COMMANDS 2–2

2.3 SEARCH AND REPLACE 2–3

2.4 CUT AND PASTE 2–4

CHAPTER 3 EMACS MAIL AND RMAIL 3–1

3.1 MAIL 3–1

3.2 RMAIL 3–2
3.2.1 Invoking RMAIL 3–2
3.2.2 Listing RMAIL Messages 3–2
3.2.3 Reading RMAIL Messages 3–3
3.2.4 Deleting and "Undeleting" RMAIL Messages 3–3
3.2.5 Saving RMAIL Messages 3–3
3.2.6 Forwarding and Replying to Messages 3–3
3.2.7 Saving Copies of Mail 3–3

Contents

CHAPTER 4 GNUS 4–1

CHAPTER 5 EMACS HELP 5–1

5.1 EMACS DOCUMENTATION 5–1

FIGURES

1–1 A GNU Emacs Buffer 1–2

4–1 A GNUS Buffer 4–3

TABLES

1 DEC Trademarks ii

1–1 Emacs Command Conventions 1–1

2–1 Cursor Commands 2–2

2–2 File Commands 2–2

2–3 Buffer Commands 2–3

2–4 Formatting Commands 2–3

2–5 Search Commands 2–3

2–6 Kill and Yank Commands 2–5

3–1 Mail Commands 3–1

3–2 RMAIL Commands 3–3

4–1 GNUS Newsgroup Commands 4–4

4–2 GNUS Subject Commands 4–4

4–3 GNUS Article Commands 4–5

4–4 GNUS Post Commands 4–6

5–1 Help Command Key Sequences 5–1

Preface

GNU Emacs is a full-screen, ‘‘self-documenting, customizable, extensible real-time display’’ editor first developed at the Massachusetts Institute of Technology Artificial Intelligence Laboratory by Richard Stallman and Guy Steele. Distributed by the Free Software Foundation (FSF), ‘‘GNU Emacs is *free* ... everyone is free to use it and free to redistribute it on a free basis.’’† This document presents summaries of those commands needed most by the new user.‡

Intended Audience

This document is in production for the benefit of the faculty, students and staff of the Graduate School and University Center of The City University of New York.

Document Production Information

This document was produced with VAX DOCUMENT, an electronic publishing system developed by the Digital Equipment Corporation. VAX DOCUMENT is installed on the Graduate School **VAX** (**cunyvms1.gc.cuny.edu**). Requests for information should be forwarded to **mcc@cunyvms1.gc.cuny.edu**.

† Richard Stallman, *GNU Emacs Manual*, 3. For more information about GNU Emacs, write to the Free Software Foundation, 1000 Mass Ave. Cambridge, MA 02138.

‡ Future versions of this document will contain a more detailed discussion of the GNU Emacs display editor.

1 The GNU Emacs Display Editor

1.1 Introduction

Unlike other text editors, Emacs offers users a total computing environment. Countless operations can be performed in multiple windows: files can be edited, copied, renamed and deleted; text can be formatted for text processors; directories can be listed; programs compiled; UNIX shell commands executed. Macros can be created to facilitate the execution of repetitive commands; abbreviations can be written to reduce repetitive editing tasks. Emacs also provides interfaces for electronic mail and UseNet newsreaders.

1.1.1 Emacs Conventions

Emacs employs a variety of command key sequences composed of the Control and ''Meta'' or Escape keys, plus another key or keys. A number of conventions for these command key sequences have been standardized in Emacs documentation. Table 1–1 presents Emacs command key sequence conventions.

Table 1–1 Emacs Command Conventions

Key	Function
C–	Press the Control key.
DEL	Press the Delete or Backspace key.
M–	Press the Esc or ''Meta'' key.
RET	Press the Return or Enter key.
SPC	Press the SPACEBAR.
TAB	Press the Tab key.

Note: If your Escape key does not function properly (or if you do not have an Escape key), which sometimes happens when using a modem to connect to CUNYGS or timessqr, use C–[(Control left-bracket).

1.2 Getting Started

To access Emacs, type **emacs** at the command prompt and press **RET**. A blank Emacs **scratch* buffer* window will open on your screen. Buffers are temporary storage areas for text. All editing is done in buffers, then saved to files. The Emacs window is composed of four functional areas:

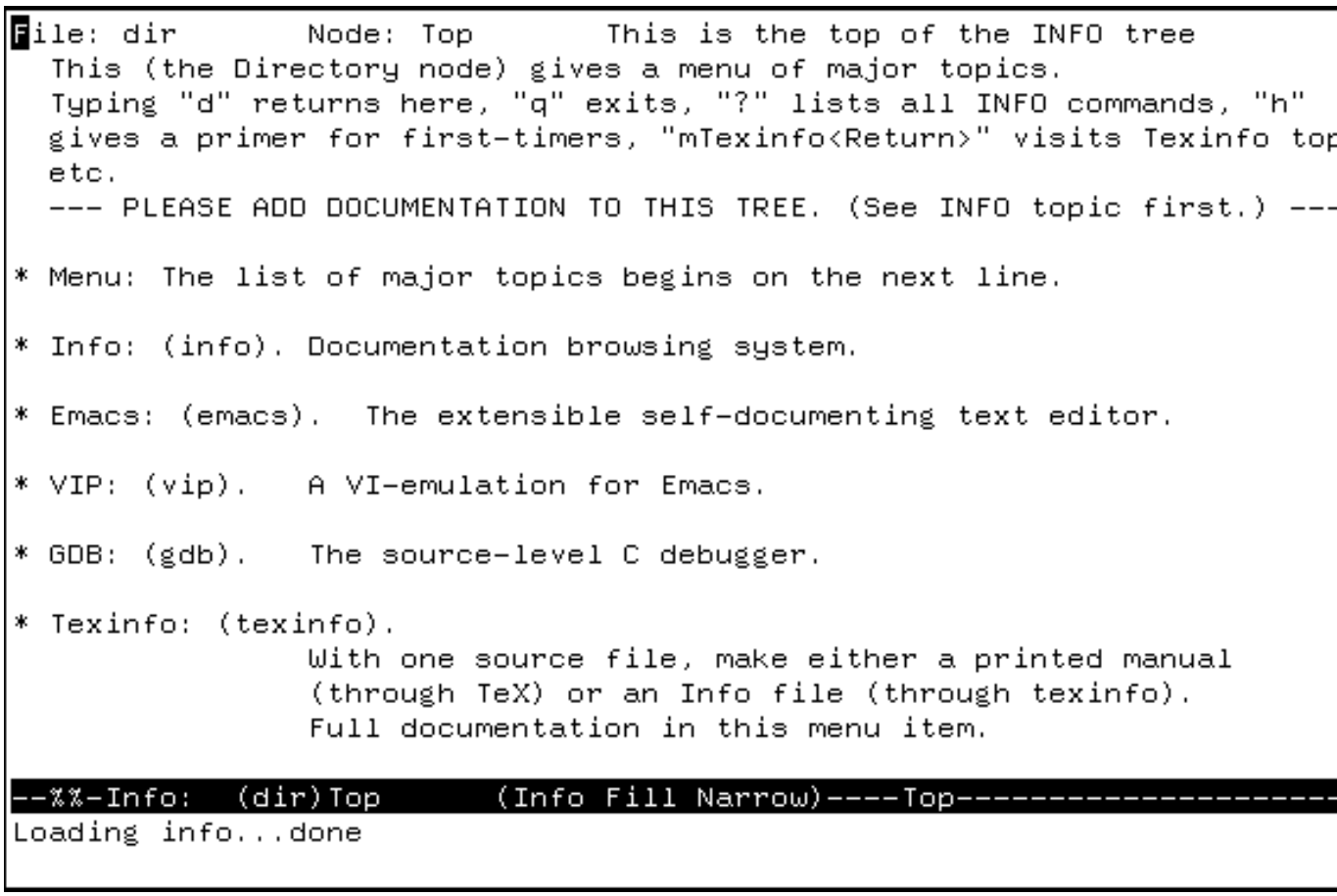
❶ The ''buffer'' is the area in which you edit text.

❷ The ''point,'' represented by the ''cursor,'' marks your position in the buffer.

❸ The ‘‘mode line,’’ located near the bottom of the screen, displays information about your Emacs session, including the buffer name, the name of major and minor modes associated with the buffer, the position of the cursor in the buffer, and the modification status.

❹ The ‘‘minibuffer,’’ also known as the ‘‘echo area,’’ located beneath the mode line, prompts you for commands and reports status information.

Figure 1–1 A GNU Emacs Buffer



2 Emacs Commands

The Emacs editor is written in a dialect of the LISP programming language called, not surprisingly, ELISP. Emacs commands are LISP symbols named after functions defined to perform specific tasks. Command key sequences are ''bound'' to these symbols. Thus to find and read a file in an Emacs buffer, you can type

```
M-x find-file
```

or you can type

```
C-x C-f
```

because the command key sequence **C–x C–f** is bound to the **find-file** command. While key bindings can be changed to suit the needs of a particular user or operating system, the function that performs the command remains the same.

To issue an Emacs command using the Control key, press **C–** and the following key *simultaneously*. For example, to issue the ''find-file'' command (**C–x C–f**), press **C–** and **x f** at the same time.

To issue an Emacs command using the Escape key, press **M–**, *release it*, then press the following key.

''Command line completion'' is one of the great advantages of GNU Emacs. For example, after typing the first few letters of a filename in response to the **C–x C–f** command (find-file), press TAB. Emacs will complete the filename, create a ''completion list'' if the letters indicate more than one file, or complain if there is ''No match.''

2.1 The .emacs File

Although Emacs will run properly without an initialization file (often called a ''dotfile''), it is always better to have a *.emacs* file in your home directory. The Network Operations group has provided an excellent *.emacs* file. To copy this file into your home directory (represented by a tilde), use the following command *as written*:

```
cp /usr/skel/.emacs ~ RET
```

Note: It beyond the scope of this document to explain the significance and the syntax of the *.emacs* file; nevertheless, from time to time I will offer a line of code which, if entered in your .emacs file, will make Emacs easier to use.

2.2 Edit Commands

Table 2–1 lists command key sequences used to move the cursor in buffers.

Table 2–1 Cursor Commands

Sequence	Command	Cursor Movement
C–f	forward-char	Forward one character
C–b	backward-char	Back one character
C–p	previous-line	Up one line
C–n	next-line	Down one line
C–a	beginning-of-line	To beginning of line
C–e	end-of-line	To end of line
C–v	scroll-up	Scroll forward one screen
M–v	scroll-down	Scroll backward one screen
M–f	forward-word	Forward one word
M–b	backward-word	Back one word
M–a	backward-sentence	Back one sentence
M–e	forward-sentence	Forward one sentence
M–(	backward-paragraph	Back one paragraph
M–)	forward-paragraph	Forward one paragraph
M–<	beginning-of-buffer	Back to beginning of buffer
M–>	end-of-buffer	Forward to end of buffer

Table 2–2 presents command key sequences used to manipulate files in buffers.

Table 2–2 File Commands

Sequence	Command	Action
C–l	recenter	Redraw the screen display
C–x C–f	find-file	Display file in buffer
C–x i	insert-file	Insert another file at the cursor point
C–x C–w	write-file	Write contents of buffer to file
C–x C–s	save-buffer	Save buffer contents to file
C–x s	save-some-buffers	Save buffer contents to file
C–x C–c	save-buffers-kill-emacs	Quit Emacs

Note: Depending on your terminal emulation, the standard key binding for the ‘‘save-buffer’’ command (C–x C–s) may not function

properly, because the control key sequence <CTRL/s> is used to freeze the screen. Not to worry. To save the contents of a buffer to a file, use the ''save-some-buffers'' command: first type C–x followed by an s, as listed in the table above. Emacs will ask you to confirm the save in the minibuffer. Type y and the file will be saved to disk.

Table 2–3 presents command key sequences used to manipulate buffers.

Table 2–3 Buffer Commands

Sequence	Command	Action
C–x b	switch-to-buffer	Move to specified buffer
C–x o	other-window	Move cursor clockwise to another window
C–x 4 b	switch-to-buffer-other-window	Move to buffer in another window
C–x C–b	list-buffers	Display buffer list
C–x k	kill-buffer	Delete specified buffer
C–x 0	delete-window	Delete current window
C–x 1	delete-other-windows	Delete all but current window

Table 2–4 presents command key sequences used to format text.

Table 2–4 Formatting Commands

Sequence	Command	Action
M–q	fill-paragraph	Format paragraph
M–g	fill-region	Format region
C–u M–q		Justify paragraph

2.3 Search and Replace

Emacs offers a variety of methods to search and replace text, including simple searches of words and phrases, ''incremental'' searches, which begin with the first character of the search text, ''regular expression'' or ''pattern matching'' searches, and ''query-replace'' searches, which request confirmation before replacing the found text with new text specified by the user.

Table 2–5 presents command key sequences used to search for character strings in buffers.

Table 2–5 Search Commands

Sequence	Command	Action
C–s Esc C–w	word-search-forward	Search forward for word

Table 2–5 (Cont.) Search Commands

Sequence	Command	Action
C–r Esc C–w	word-search-backward	Search backward for word
C–r	isearch-backward	Search backward to beginning of buffer
C–s	isearch-forward	Search forward to end of buffer
M–%	query-replace	Search forward and wait for confirmation of replacement

As stated above, **C–s** is used to freeze the screen. The most commonly used search command is ''isearch-forward'', bound to **C–s**. To avoid this problem, use the command name when specifying a forward search, as follows:

```
M-x isearch-forward
```

If typing ''**M–x isearch-forward**'' seems tedious, there is a more efficient way to perform a forward incremental search: create a key binding for the command. This is done by adding a line of ELISP code in your *.emacs* file. For example, if you would like to bind **M–s** to the ''isearch-forward'' command, insert the following line *as written* in your *.emacs* file:

```
(define-key global-map "\M-s" 'isearch-forward)
```

2.4 Cut and Paste

To ''cut and paste'' in Emacs parlance is to ''kill and yank''. Text can be deleted by character, word, line, region, and buffer. (A ''region'' of text is a section ''marked'' or selected for various operations, such as deletion.) Deleted words, lines, regions, and buffers are placed in a ''kill ring,'' where they can be recovered with a ''yank'' command.

To mark text for deletion, place the cursor at the beginning of the text you wish to delete and enter the **C–SPC** or **C–@** command. The message, ''Mark set,'' will appear in the minibuffer. Move the cursor to the end of the text to be deleted and enter the **C–w** command. The text region will disappear into the kill ring, where it will stay until the kill ring is full, or until you end your Emacs session. If you wish to yank the deleted text elsewhere in the current buffer, move the cursor to the desired point and enter the **C–y** command. To yank the text in another file, open the file with the **C–x f** command and yank the text with **C–y**.

By default, the kill ring contains 30 deletions. The ''yank-pop'' command (**M–y**) is used to retrieve those deletions from the kill ring. **M–y** replaces the most recent yank with the yank that precedes it in the kill ring. The ''yank-pop'' command can be repeated in order to move through the kill ring. **M–y** is helpful when trying to choose between adjectives; for example, if you have deleted ''dark and stormy'' from the phrase, ''It was a dark and stormy night,'' and replaced it with ''hot and steamy,'' then

replaced that with ''cool and breezy,'' you can use the ''yank-pop'' command to repeat each adjectival phrase in sequence until you find the one that best expresses your idea.

Note: Marked regions are invisible; other than the ''Mark set'' message in the minibuffer, there is no indication that a region has been marked. If you would like to verify the marked region, use the C–x C–x command, ''exchange-point-and-mark.'' The cursor will move to the beginning of the region. Repeat the C–x C–x command and the cursor will move back to the end of the region.

Table 2–6 presents command key sequences used to mark, kill, and yank text.

Table 2–6 Kill and Yank Commands

Sequence	Command	Action
C–x u	advertised-undo	Undo last edit command
C–g	keyboard-quit	Cancel current command
DEL	delete-backward-char	Delete previous character
C–d	delete-char	Delete the character under the cursor
M–d	kill-word	Delete the word under the cursor
C–k	kill-line	Delete from the cursor to line's end
C–w	kill-region	Delete marked region
M–w	copy-region-as-kill	Copy marked region into kill ring
C–y	yank	Paste deleted text
M–y	yank-pop	Paste most recent deletion
C–SPC	set-mark-command	Mark beginning or end of region
C–@	set-mark-command	Mark beginning or end of region
M–h	mark-paragraph	Mark a paragraph
C–x h	mark-whole-buffer	Mark entire buffer
C–x C–x	exchange-point-and-mark	Move cursor to begining or end of region

3 Emacs Mail and RMAIL

Emacs offers users very powerful and convenient electronic mail interfaces. The Mail program is used to write and send mail messages; RMAIL is used to read and store mail messages.

Emacs Mail recognizes aliases defined in the *.mailrc* file.

3.1 Mail

To invoke Emacs Mail, type **C–x m** and press RET. The Mail window will appear in your Emacs buffer containing the **To:** and **Subject:** fields. The cursor will be placed after the **To:** field. Type the address of the recipient. Multiple addresses can also be entered at the **To:** field, provided that a comma (,) separates each address. Using the down arrow key or the **C–n** command, move the cursor to the **Subject:** field. Enter a subject if you so desire, then move the cursor to the message area:

```
To: boris@badinoff.kremlin.edu
Subject: Capitalism
--text follows this line--
   .
   .
   .
```

Edit your message with standard Emacs commands. When you have completed your message, enter the **C–c C–c** command to send it. Emacs Mail will confirm the send with the message ''Sending...done'' in the minibuffer.

The **C–c C–f C–c** command is used to create a **CC:** field for addresses to which copies of the message are to be sent. If several copies are to be sent, separate each address with a comma (,). If the addresses do not fit on one line, repeat the **C–c C–f C–c** to create another **CC:** field.

If you have a *.signature* file in your home directory that you would like to include in your message, enter the **C–c C–w** command.

Table 3–1 presents command key sequences used to edit and send mail messages.

Table 3–1 Mail Commands

Sequence	Command	Action
C–x m	mail	Invoke Emacs Mail
C–x 4 m	mail-other-window	Invoke Mail in another window
C–c C–f C–t	mail-to	Create and move to the To: field

Table 3–1 (Cont.) Mail Commands

Sequence	Command	Action
C–c C–f C–c	mail-cc	Create and move to the CC: field for sending a copy
C–c C–f C–s	mail-subject	Create and move to the Subject: field
C–c C–w	mail-signature	Insert the .signature file
C–x i *filename*	insert-file	Insert a file to be mailed
C–c C–c	mail-send-and-exit	Send the message and exit Mail
C–c C–s	mail-send	Send the message without exiting Mail

3.2 RMAIL

The RMAIL program is used to read and manage mail messages. When mail arrives, it is stored in a system mailbox. When RMAIL is invoked, all incoming messages are moved from the system mailbox to an inbox in the user's home directory called, not surprisingly, *RMAIL*.

3.2.1 Invoking RMAIL

To invoke RMAIL, type **M–x rmail** and press RET. RMAIL will respond with the first unread message, or the most recent read message, in your inbox.

Most RMAIL commands consist of a single letter, as listed in the table below.

3.2.2 Listing RMAIL Messages

If you have received several messages, you can list them with the **h** command (short for ''header''). RMAIL will open two windows in the buffer: the ''RMAIL-summary'' window, containing the header list; and the RMAIL window, containing the last read message or the first unread message. As you move up and down the list with the arrow keys or the **p** and **n** commands, RMAIL will display the indicated message in the RMAIL window.

3.2.3 Reading RMAIL Messages

The spacebar (SPC) is used to scroll forward in a message; the delete key (DEL) is used to scroll backward. The **n** command is used to move to the next message; the **p** command is used to move to the previous message. Note: The **n** and **p** commands skip messages marked for deletion. To move forward or backward to marked messages, use the **M–n** and **M–p** commands.

3.2.4 Deleting and ''Undeleting'' RMAIL Messages

Messages are marked for deletion with the **d** command, and subsequently deleted when you exit RMAIL. The **u** command is used to ''undelete'' messages marked for deletion.

3.2.5 Saving RMAIL Messages

Messages that are not deleted remain in the *RMAIL* file unless they are saved to another RMAIL format file with the **o** command, or to an ASCII text file with the **C–o** command. The difference between these two file formats is significant if you intend to move your mail files to another computer or to read your mail with a different Mail User Agent (MUA). Use the **o** command if you intend to read your mail exclusively in RMAIL.

3.2.6 Forwarding and Replying to Messages

Mail is forwarded with the **f** command. Use the **r** command to reply to a mail message. These commands place an Emacs Mail buffer on top of the RMAIL buffer, allowing you to edit the message to be forwarded or, if you are composing a reply, to insert the original message with the **C–c C–y**.

3.2.7 Saving Copies of Mail

Although the easiest way to maintain a record of the messages you send is to send yourself a copy, the most efficient method is to have the copy sent to a file in your home directory. For this you must set the ''mail-archive-file-name'' variable in your *.emacs* file. Using the Emacs editor, append the following line of ELISP code to your *.emacs* file:

```
(setq mail-archive-file-name "~/outbox")
```

Note that any filename can be specified, provided that the syntax is correct.

Table 3–2 presents command key sequences used to read, save, reply to and forward mail messages.

Table 3–2 RMAIL Commands

Key Command	Action
M–x rmail	Begin RMAIL

Table 3–2 (Cont.) RMAIL Commands

Key Command	Action
SPC	Scroll message forward
DEL	Scroll message backward
.	Move to the top of the message
j	Jump to first message
n	Move to next message
p	Move to previous message
>	Move to last message
g	Check for new messages
d	Mark message for deletion and move forward
C–d	Mark message for deletion and move backward
M–n	Move to next message; disregard deletion mark
M–p	Move to previous message; disregard deletion mark
u	Undelete message marked for deletion
x	Delete all marked messages
e	Delete all marked messages
m	Invoke Emacs Mail in RMAIL
r	Reply to current message
C–c C–y	Include current message in reply
f	Forward current message
o *filename*	Save current message in RMAIL format
C–o *filename*	Save current message in UNIX format
M–s	Search for regular expression
h	Show list of messages by header

4 GNUS

GNUS, a subsystem of Emacs, is the ultimate UseNet newsreader. Like Emacs, GNUS offers command line completion and completion lists. Like RMAIL, GNUS displays two windows: one containing a list of subject headers; the other containing the current article.

Before invoking GNUS, you must enter a few lines of ELISP code into your *.emacs* file. The Network Operations group has in fact created a *.emacs* file containing this code. If you have not modified your *.emacs* file, you will find the NetOps *.emacs* file in **timessqr**'s /u/questions directory. If you have modified your *.emacs* file, append the following code to it, *as written*:

```
(autoload 'gnus "gnus" "Read Netnews." t)
(autoload 'gnus-post-news "gnus" "Post Netnews." t)
(setq gnus-nntp-service nil)
(setq nnspool-inews-program "/usr/local/bin/inews")
(setq nnspool-active-file "/usr/users/news/active")
(setq gnus-your-domain "gc.cuny.edu")
(setq gnus-your-organization "Graduate School and University Center, C.U.N.Y, New York")
(setq gnus-novice-user t)  ;; prompts when posting an article
```

Most GNUS commands consist of a single letter. Often the same letter command is used in each of the three GNUS buffers: Newsgroup, where newsgroups are listed; Subject, where subject headers are listed; and Article, where the article is displayed. For example, in the Newsgroup buffer, the spacebar (SPC) is used to display a selected newsgroup's subject header list; in the Subject buffer, SPC is used to display an article from the subject header list; and in the Article buffer, SPC is used to scroll forward in an article. The **q** command is used to quit both the Subject and Newsgroup buffers.

To invoke GNUS, type **M–x gnus** and press RET. You will see a series of brief messages in the minibuffer, ending with

```
NNTP server:
```

Press RET. If you have never used GNUS before, GNUS will automatically subscribe you to all available newsgroups. Finally the Newsgroup buffer will appear:

```
   9497: rec.arts.drwho
    428: comp.unix.large
   3260: comp.unix.internals
    739: comp.lang.objective-c
   3705: comp.fonts
    181: rec.games.mud.misc
   5052: comp.binaries.ibm.pc.archives
    395: comp.research.japan
   1330: talk.rumors
   3637: sci.virtual-worlds
  29894: rec.bicycles
   8617: comp.unix.ultrix
  21935: comp.unix.sysv386
  17564: comp.unix.questions
    .
    .
    .
```

Use the **j** command to ''jump'' to a newsgroup; you will be prompted for a newsgroup in the minibuffer. Type the newsgroup's name, press RET, and the cursor will find its way to the selected newsgroup. Press SPC. The total number of articles in the newsgroup will appear in the minibuffer, and you will be asked:

```
How many articles from comp.unix.ultrix (default 8663):
```

If you press return, GNUS will attempt to list all of the articles in the newsgroup. *Pick a reasonable number of articles.* Do not just press RET, because GNUS will force you to wait while it spends time looking for more articles than you have time to read. Eventually two windows will appear: the top window containing the Subject buffer's list of subject headers; and the bottom window containing the first of your selected articles. Note the capital ''D'' in the left column of the Subject buffer: each article you read is marked for deletion. When you quit GNUS, marked articles are deleted from the Subject header list. (GNUS maintains a record of your newsgroups and available articles in the *.newsrc* file; this file is created automatically in your home directory the first time you invoke GNUS.)

Table 4–1 lists GNUS Newsgroup buffer commands.

Figure 4–1 A GNUS Buffer

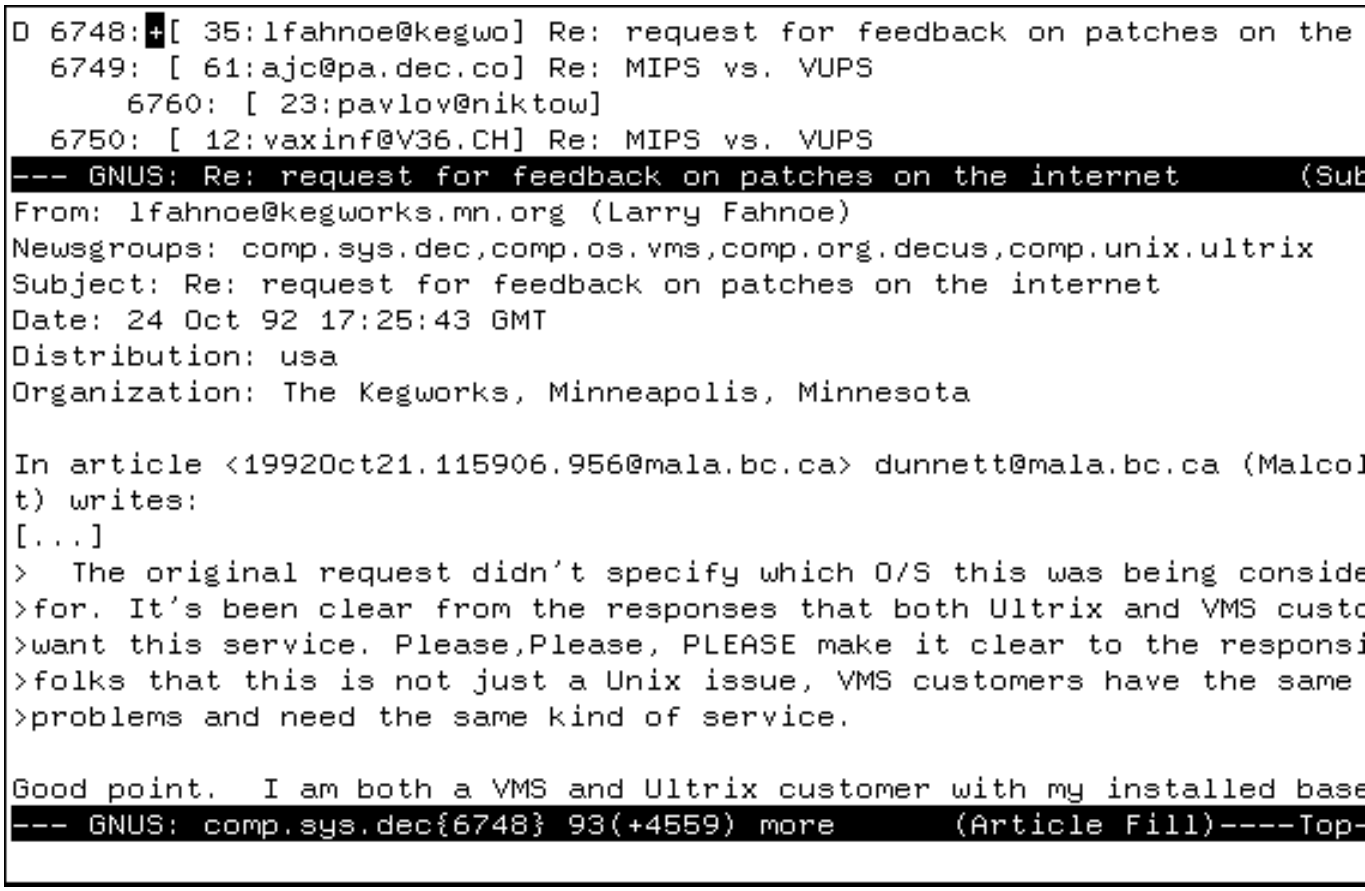


Table 4–1 GNUS Newsgroup Commands

Key	Command	Action
n	gnus-Group-next-unread-group	Move to next newsgroup of unread articles
p	gnus-Group-next-unread-group	Move to previous newsgroup of unread articles
N	gnus-Group-next-group	Move to next newsgroup
C–n	gnus-Group-next-group	Move to next newsgroup
P	gnus-Group-prev-group	Move to previous newsgroup
C–p	gnus-Group-prev-group	Move to previous newsgroup
j	gnus-Group-jump-to-group	Move to a specified newsgroup
/	isearch-forward	Incremental search forward
<	beginning-of-buffer	Move to beginning of Newsgroup buffer
>	end-of-buffer	Move to end of Newsgroup buffer
SPC	gnus-Group-read-group	Select newsgroup from list

Table 4–2 lists GNUS Subject buffer commands.

Table 4–2 GNUS Subject Commands

Key	Command	Action
C–n	gnus-Subject-next-subject	Move to next header
C–p	gnus-Subject-prev-subject	Move to previous header
M–n	gnus-Subject-next-unread-subject	Move to next unmarked header
M–p	gnus-Subject-prev-unread-subject	Move to previous unmarked header
j *number* RET	gnus-Subject-goto-subject	Move to header specified by header number

Table 4–2 (Cont.) GNUS Subject Commands

Key	Command	Action
/	isearch-forward	Incremental search of headers
SPC	gnus-Subject-next-page	Select article
g	gnus-Subject-show-article	Select article
n	gnus-Subject-next-unread-article	Read next unmarked article
p	gnus-Subject-prev-unread-article	Read previous unmarked article
N	gnus-Subject-next-article	Read next article
P	gnus-Subject-prev-article	Read previous article
M–C–n	gnus-Subject-next-same-subject	Read next article with the same subject (follow a ``thread'')
M–C–p	gnus-Subject-prev-same-subject	Read previous article with same subject (follow a ``thread'')
.	gnus-Subject-first-unread-article	Read first unread article
l	gnus-Subject-goto-last-article	Read the last selected article
J *number* RET	gnus-Subject-goto-article	Read article specified by article number
C–c C–s n	gnus-Subject-sort-by-number	Sort headers by number
C–c C–s s	gnus-Subject-sort-by-subject	Sort headers by subject
C–c C–s d	gnus-Subject-sort-by-date	Sort headers by date
C–c C–s a	gnus-Subject-sort-by-author	Sort headers by auther

Table 4–3 lists GNUS Article buffer commands.

Table 4–3 GNUS Article Commands

Key	Command	Action
SPC	gnus-Subject-next-page	Scroll down article
DEL	gnus-Subject-prev-page	Scroll up article
<	gnus-Subject-beginning-of-article	Move to top of article

Table 4–3 (Cont.) GNUS Article Commands

Key	Command	Action
>	gnus-Subject-end-of-article	Move to end of article
v	gnus-Subject-show-all-headers	Show all article headers
u	gnus-Subject-mark-as-unread-forward	Mark article as unread and move to next subject
M–u	gnus-Subject-clear-mark-forward	Clear marks and move to next subject
o	gnus-Subject-save-article	Save article in RMAIL format.
C–o	gnus-Subject-save-in-rmail	Save article in UNIX mailbox format

When saving articles, GNUS will prompt you for a filename. You can save several articles to the same file.

Table 4–4 presents commands used to post articles to newsgroups.

Table 4–4 GNUS Post Commands

Key	Command	Action
f	gnus-Subject-post-reply	Post followup to current article
F	gnus-Subject-post-reply-with-original	Include current article in reply
a	gnus-Subject-post-news	Post a new article
M–x postnews		Post a new article
C	gnus-Subject-cancel-article	Cancel your posted article
r	gnus-Subject-mail-reply	Reply to article's author in mail
R	gnus-Subject-mail-reply-with-original	Include current article in reply to author
m	gnus-Subject-mail-other-window	Edit a mail message in another window

To quit GNUS, move the cursor to the Subject buffer and type **q**. When the Newsgroup buffer reappears, type **q** again. GNUS will ask you if you really want to quit. Type **y**. GNUS will update your *.newsrc* file and quit.

5 Emacs Help

Emacs offers extensive online help, including an excellent tutorial.

Table 5–1 Help Command Key Sequences

Sequence	Command	Action
C–h	help-command	Begin help command
C–h i	info	Display INFO menu
C–h ?	help-for-help	Describe help command
C–h t	help-with-tutorial	Begin interactive tutorial
M–x doctor		Call online psychoanalyst

5.1 Emacs Documentation

A number of fine manuals about GNU Emacs are commercially available, some of which are listed here:

1. Richard Stallman's *GNU Emacs Manual* (Free Software Foundation, 1986)
2. *Learning GNU Emacs* (O'Reilly & Associates, 1992)
3. *GNU Emacs: UNIX Text Editing and Programming* (Hewlett Packard, 1992)