Getting Started with the Emacs Screen Editor

An introduction to the Emacs screen editor, which is available on Unix systems.

AUTHOR Information Systems Services
University of Leeds

DATE March 2000

EDITION 2.3

BEG 6

50p

Contents

Format Conventions

In this document the following format conventions are used:

Commands that you must type in are shown in **`bold Courier`** font.	**`emacs`**
Input which must be replaced by your details is given in *italics*.	**`emacs`** *`filename`*
Keys that you press are enclosed in angle brackets.	<Enter>

Feedback

If you notice any mistakes in this document please contact the Information Officer. Email should be sent to the address info-officer@leeds.ac.uk

Copyright

Print Record

This document was printed on 11-Apr-00.

1 Introduction

Aim of this Document

This document gives an introduction to the use of the Emacs screen editor. The document is intended for new users of Unix systems who wish to use the Emacs editor.

Unix Editors

A wide range of tools are available on Unix systems which can be used to create and edit files. These tools include:

ed	A basic line editor.
ex	A more powerful line editor.
vi	A screen editor, which makes use of the ex editor.
Emacs	A powerful screen editor.
cut, paste, awk, etc.	Examples of some of the Unix tools which can be used to carry out editing functions.

This document describes the use of the Emacs editor.

GNU Emacs

GNU Emacs is available on all Unix systems supported by the ISS. GNU Emacs (note that GNU is a recursive definition which stands for GNU's Not Unix) is a text editor and **not** a word processor; it is not normally recommended for producing documents such as essays or theses.

The advantages of GNU Emacs are:

- It is a powerful, highly functional editor.
- It is extensible. It is written mainly in Lisp and new features can be developed in Lisp. However this is not a trivial task.
- It provides a complete working environment. You can use GNU Emacs for reading and sending mail, manipulating your files, compiling programs, etc.
- It is portable.

The disadvantage with powerful systems such as GNU Emacs is the ease of use. Commands in GNU Emacs are associated with complex key stokes which can take time to get used to.

2 Running Emacs

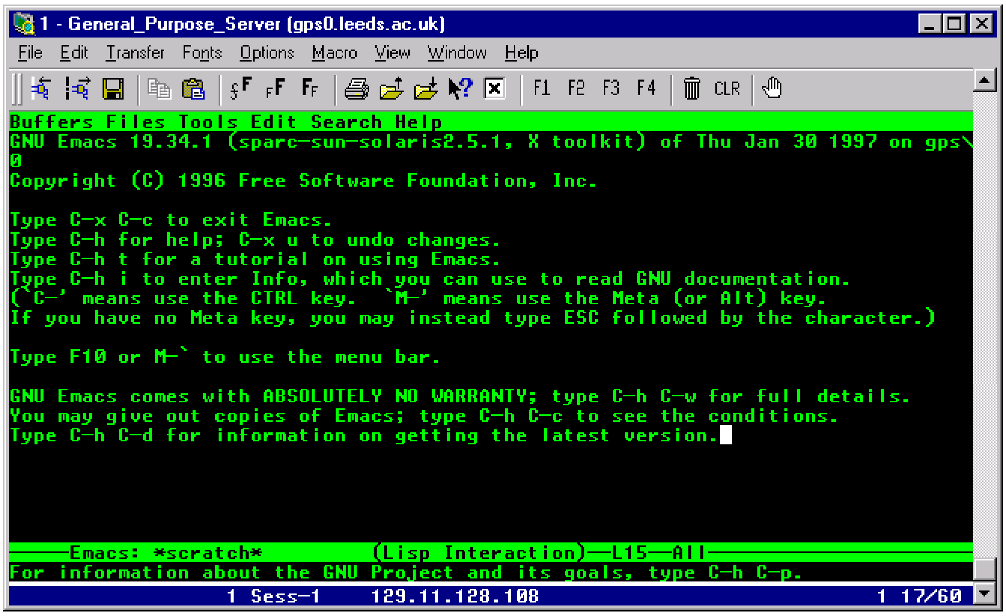
The "emacs" Command

To run Emacs you should give the command:

```
% emacs [filename]
```

Note that a filename can optionally be supplied.

Once you have given the command the following screen will be displayed.



1 - General_Purpose_Server (gps0.leeds.ac.uk)
File Edit Transfer Fonts Options Macro View Window Help
F1 F2 F3 F4 CLR

```
Buffers Files Tools Edit Search Help
GNU Emacs 19.34.1 (sparc-sun-solaris2.5.1, X toolkit) of Thu Jan 30 1997 on gps\
0
Copyright (C) 1996 Free Software Foundation, Inc.

Type C-x C-c to exit Emacs.
Type C-h for help; C-x u to undo changes.
Type C-h t for a tutorial on using Emacs.
Type C-h i to enter Info, which you can use to read GNU documentation.
(`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.)

Type F10 or M-` to use the menu bar.

GNU Emacs comes with ABSOLUTELY NO WARRANTY; type C-h C-w for full details.
You may give out copies of Emacs; type C-h C-c to see the conditions.
Type C-h C-d for information on getting the latest version.



----Emacs: *scratch*            (Lisp Interaction)--L15--All-----------------
For information about the GNU Project and its goals, type C-h C-p.
                 1 Sess-1     129.11.128.108                          1 17/60
```

Figure 1 The Initial Emacs Screen

The initial screen shows menus for commands at the top, plus a page of initial help information in the main window. This screen will blank the moment you start typing.

The things to note:

C-x means **<Ctrl+X>** **M-x** means **<Esc+X>**

F10 cannot be used to access the menus *on PCs* as it is also used to access the menus of the Telnet program itself. **<Alt-'>** works but is non-obvious to use, which essentially means that from a PC you cannot use the Emacs menus.

Problems? If the screen shown in Figure 1 is not displayed, you may have failed to specify the correct terminal type. The command:

```
% setenv TERM terminal_type
```

should be used to specify the terminal type. For example:

```
% setenv TERM vt100
```

will set the correct terminal type for a vt100 terminal.

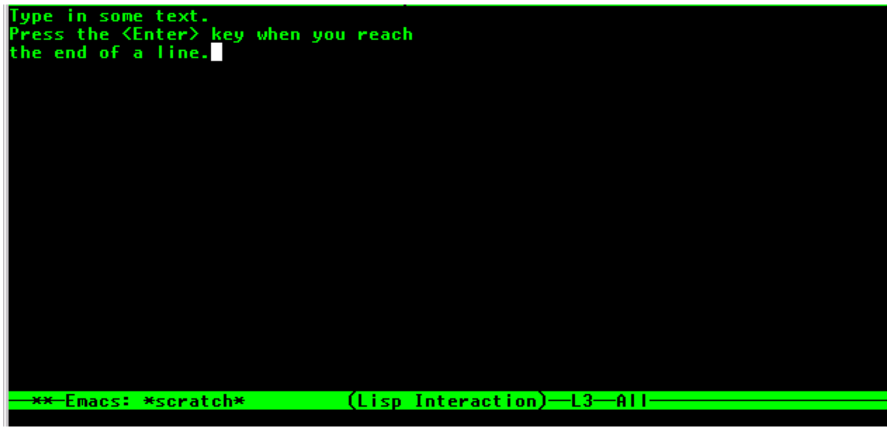
Using A PC As A Terminal

If you are using the Telnet software to access a Unix system from a PC you should set the terminal type to *vt100* as described above.

3 Creating A File

Entering Text

To To enter text, you simply start typing. Please note that you should normally press the <Enter> key when you reach the end of a line, as you would if you were using a typewriter.



```
Type in some text.
Press the <Enter> key when you reach
the end of a line.











--**-Emacs: *scratch*          (Lisp Interaction)--L3--All-------------------
```

Figure 2 Entering Text

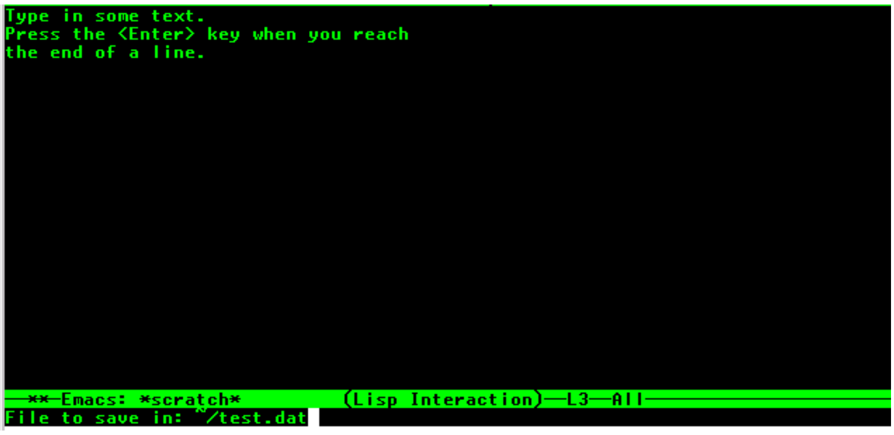
Saving Your File

You should save your file at frequent intervals. The command used to save a file is:

```
<Ctrl X> <Ctrl S>
```

This means that you should press the Control key (which is marked Ctrl on most keyboards) and then, keeping the Ctrl key pressed down, press the X key. Then press the Ctrl key and the S key in a similar manner.

The first time you give this command, Emacs will ask you to give the name of the file. You should enter the file name as shown below and press <Enter>.



```
Type in some text.
Press the <Enter> key when you reach
the end of a line.











--**-Emacs: *scratch*          (Lisp Interaction)--L3--All-------------------
File to save in: ~/test.dat
```

Figure 3 Saving Your File

Quitting Emacs

When you have finished using Emacs, you should give the command:

```
<Ctrl X> <Ctrl C>
```

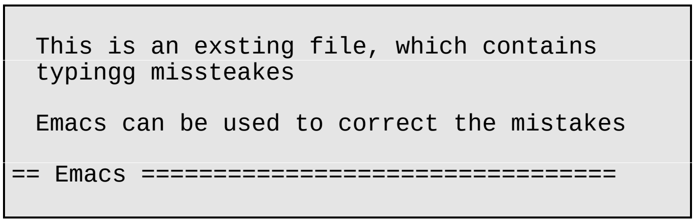
You can save the file again, then be returned to Unix operating system.

4 Editing An Existing File

To edit an existing file, give the command:

```
% emacs filename   e.g.   % emacs test.dat
```

Emacs loads with the file displayed. The example file below will show how existing text can be edited with emacs.



```
  This is an exsting file, which contains
  typingg missteakes

  Emacs can be used to correct the mistakes

== Emacs ================================
```

Figure 4 Editing An Existing File

Inserting New Text

To insert new text you should simply move the cursor to the required position in the file. Use the keyboard arrow keys for this (see below if the arrow keys don't work). Then type in the new text.

Deleting Text

The commands given in the table below can be used to delete text.

`<Ctrl D>`	Delete the character under the cursor.
`<Del>`	Delete the character to the left of the cursor (Note this is different from PCs where it deletes the character on the right).
`<Esc> D`	Delete from the cursor position to the end of word.
`<Esc> <Del>`	Delete from the cursor position to the start of word.
`<Ctrl K>`	Delete from the cursor position to the end of line.

Table 1 Commands Used To Delete Text

If you accidentally delete text, the command `<Ctrl X> U` can be used to undo the deletion. Repeated use of this command undoes previous deletes, so to recover the last deletion type `<Ctrl X> U` and to restore the deletion before that type `<Ctrl X> U` again. This command not only works on deletions but on all changes since you last saved.

Moving Around The File

There are many commands available which can be used to move around the file. The most widely used commands are listed in Table 1.

In the list of commands, you should note that many Emacs commands contain a letter which gives an indication of its function. For example, to move the cursor forward by one character, you should press `<Ctrl F>`.

Top of file	`<Esc> <`
Bottom of file	`<Esc> >`
Next screen	`<Ctrl V>`
Previous screen	`<Esc> V`
Next line	`<Ctrl N>` or ↓
Previous line	`<Ctrl P>` or ↑
Forward character	`<Ctrl F>` or →
Back one character	`<Ctrl B>` or ←
Start of line	`<Ctrl A>`
End of line	`<Ctrl E>`
Forward word	`<Esc> F`
Backward word	`<Esc> B`

Table 2 Moving Around A File

Searching For Strings

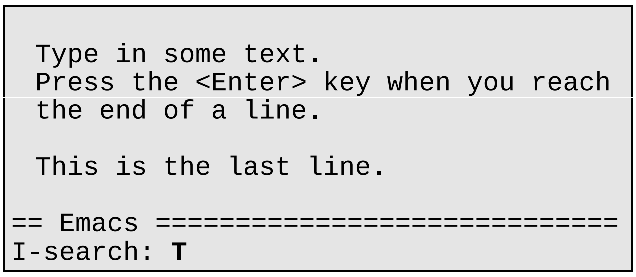
It is often more convenient to move around a file by searching for a word or sequence of characters. The following commands provide this feature:

`<Ctrl S>`	Incremental search forward
`<Ctrl R>`	Incremental search backward

The term *incremental* means that the search begins as you type.

An example of the use of the command will be given using the text shown in Figure 5.

The search command works in the following way:



```
 Type in some text.
 Press the <Enter> key when you reach
 the end of a line.

 This is the last line.

== Emacs ============================
I-search: T
```

Figure 5 The Incremental Search Command

- Typing `<Ctrl S>` followed by "t" takes you to the first instance of "t" in the file, which is the first letter of the word "Type". This illustrates the fact that the search is case insensitive, so "t" matches "T" or "t".
- Then typing "h" takes you to the next instance of "th", which is located in the word "the".
- Typing "i" will take you to the letter "i" in the word "This".

The `<Del>` key may be used to delete the last character entered in the search string and as that character is deleted the cursor returns to the previous string matched by the smaller pattern.

Both forward and backward incremental searches work in this fashion, the difference is that `<Ctrl S>` works towards the end of the file and `<Ctrl R>` works towards the beginning of the file. At any point during the search, the search direction can be changed by entering either `<Ctrl S>` or `<Ctrl R>` to indicate the new search direction.

The abort command (`<Ctrl G>`) can be typed to completely cancel the search; this returns the cursor to its initial position.

The search is stopped by typing `<Esc>`. This sets the cursor at the start of the current instance of the search string.

If you want to type the whole sequence of characters first before starting the search, press `<Esc>` as the first character in the search string, the type the characters you want to find followed by `<Enter>`. The search starts when you press `<Enter>`.

Summary of Search Commands

`<Ctrl S>`	Step forward to next match.
`<Ctrl R>`	Step backward to next match.
`<Del>`	Delete last character of search string and move to initial match of that pattern.
`<Esc>`	Stop search here or if this is the first character of the search string use conventional search.
`<Ctrl G>`	Cancel search and return to initial position.

Table 3 Search Commands

Replacing Strings

Search and replace operations can be combined using the command:

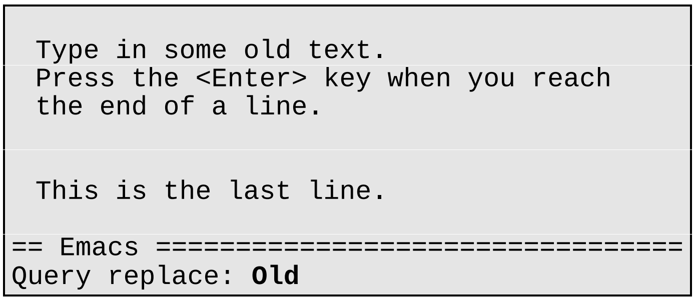
```
<Esc> %
```

which allows you to locate and selectively replace one string with another.

When you give this command the prompt:

```
Query replace:
```

is displayed at the bottom of the screen. Type in the text you wish to replace and the press <Enter>. An example of the screen display is shown below.



```
 Type in some old text.
 Press the <Enter> key when you reach
 the end of a line.


 This is the last line.

== Emacs =================================
Query replace: Old
```

Figure 6 Replacing Text

Once you have typed in the text you wish to replace and pressed the <Enter> key the prompt:

```
Query replace OLD_TEXT with:
```

will be displayed, where *OLD_TEXT* is replaced by the text you wish to replace. You should type in the new text and then press the <Enter> key.

The message:

```
Query replace OLD_TEXT with NEW_TEXT:
```

will be displayed and the cursor will then move to the first occurrence of the text to be replaced. You should then type a command to confirm that the text is to be replaced. The commands which can be given at this point are:

<Ctrl H> or ?	Display a help message.
<Spc> or y	Replace the current instance of the search string and move on.
<Del> or n	Do not replace the current instance and move on.
. (Full stop)	Replace the current instance and exit.
<Esc> or q	Exit.
<Ctrl G>	Abort command (same as exit).
!	Replace all remaining matches.

Typing "y" or " " will perform the replacement and search for the next instance.

Note The search is case insensitive, as with the incremental search, so strings such as *temp* and *Temp* would also be matched by the previous example. Once you get used to it, case-insensitive searching becomes an asset.

Moving Blocks Of Text

You can move or copy blocks of text around the file. To do this you need to set a *mark* - this is the start (or end) of the block you wish to move. You must then move to the end (or start) of the block, and give a command to either delete the block (also known as region), or copy it into a special buffer, known as the *kill ring*. You then move the cursor to the new position and paste the contents of the kill ring.

The commands to carry out these operations are:

`<Ctrl> <space>`
`<Ctrl @>`
	Set the mark.

`<Alt> <Esc>`	Set the mark. Use on PCs running Telnet. **Note** this command only works on ISS systems.
`<Ctrl W>`	Delete region
`<Esc> W`	Copy region
`<Ctrl Y>`	Paste kill ring

Table 4 Commands To Move Or Paste Regions

Note In certain circumstances if you are using a PC and the configuration files for the terminal emulation software have not been set up correctly, the command to set the mark will not work. If this happens give the command:

```
<Esc> X set-mark-command
```

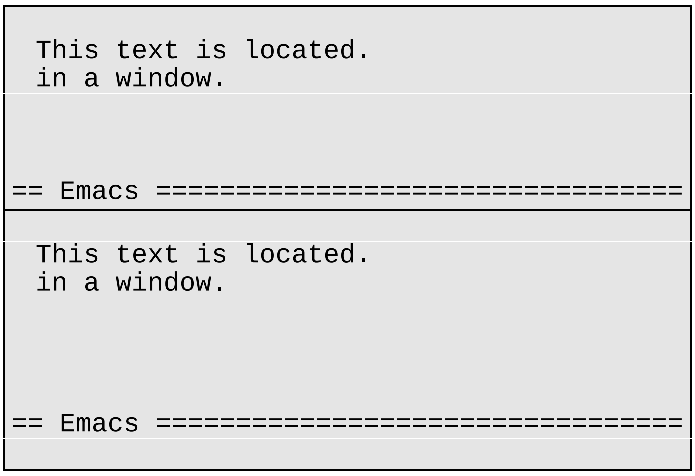
5 Advanced Features Of Emacs

Emacs is a very powerful screen editor, which contains a range of sophisticated features. The use of multiple windows is described in this section.

Handling Windows

Emacs supports multiple windows, even on dumb terminals. Multiple windows are useful for editing more than one file at a time. Each window is allocated an area of the display and is made up of two parts; the edit area and a status line, just like the basic Emacs layout. The contents of a window are known as a buffer; they may be a text file or one of the special buffer types associated with the help system or handling the collection of buffers.

Using windows is quite simple, first you open a new window by typing the command `<Ctrl X> 2`, which splits the screen horizontally into two windows. At this point both windows are associated with the same buffer, so the contents of the window and the status lines are identical – in fact, changes made to the contents of one window are also shown in the other window.



```
  This text is located.
  in a window.



== Emacs =================================

  This text is located.
  in a window.



== Emacs =================================
```

Figure 7 Multiple Windows

The second step is to access another file, this is done using the command:

`<Ctrl X> <Ctrl F>` *`filename`*

This command prompts you for a filename to edit by offering the current working directory followed by a "/" as in:

`Find file: ~/`**`test.f`**

Typing the name of the file to edit followed by `<Enter>` loads the file into a buffer for editing and associates that buffer with the current window. You can now begin editing the file.

The command "`<Ctrl X> O`", (the letter "O" not zero), moves the cursor between windows in a cyclic fashion, i.e. if you have more than two windows the cursor visits each in turn. Here is a summary of some the window-related commands which are useful to know:

`<Ctrl X> 0`	Delete the current window.
`<Ctrl X> 1`	Delete all other windows and expand the current window to the full display.
`<Ctrl X> 2`	Split the current window horizontally in two.
`<Ctrl X> O`	Move cursor into other window.

Commands that delete windows do not delete the associated buffer, the command:

```
<Ctrl X> <Ctrl B>
```

lists the allocated buffers. A buffer from this list may be associated with the current window using the command:

```
<Ctrl X> B buffer_name
```

You are prompted to select a buffer, the default being the last visited buffer which is selected by typing `<Enter>`.

Emacs Modes

A number of Emacs *modes* are available which can be used to carry out specialised editing functions for specific types of files. *Programming modes* are available for Fortran, C, Pascal and Ada source files. If you invoke Emacs on a file containing the appropriate suffix (e.g. `test.f`) the relevant mode will be invoked automatically.

Further details on Emacs modes is given in the Emacs help system, or in Emacs documentation.

Appendix 1 Getting Help

The Emacs Help System

GNU Emacs provides a very comprehensive help system, which is accessed by typing:

```
<Ctrl H> I
```

The contents of the current window are then replaced by the "directory" of the help system. The help information is organised in a hierarchical fashion. The command:

m *help_item_name*

moves down the hierarchy and the command `u` moves up the hierarchy. In the help system the following keys have special meanings:

`d`	Return to the help system directory.
`u`	Return up one level.
`n`	Move to the next item in the menu.
`p`	Move to the previous item in the menu.
`m`	Select a menu item.
`h` or ?	Get help on commands.
`q`	Quit the help system.

Standard Emacs commands such as `<Ctrl V>`, `<Esc> V`, `<Ctrl S>` and `<Ctrl R>` may be used to move around within a help item.

Note that the help system permits items to be linked to multiple points in the hierarchy; sometimes the `u` command takes you to a different place than you might expect.

Conventions Used In The Help System

The Emacs help system uses the convention that when the Control key is to be used in conjunction with another key, the key is preceded by `C-`. For example, use of the Control key and the `S` key would be described as `C-S`. In this document it is described as `<Ctrl S>`.

The Emacs help system also uses the convention that the `<Esc>` key is referred to as the *Meta* key. A reference to the use of the `<Esc> X` keys would be described as `M-X`.

Learning Emacs

Emacs contains a useful self-teach tutorial program which can be used to learn many of the facilities provided by the program. To run the tutorial program, give the command:

```
<Ctrl H> T
```

The tutorial program will then run. You should then follow the instructions displayed on the screen.