

Appendix A2

NEURON's built-in editor

Many people are already comfortable with their own text editor and will find it most convenient to load text files into hoc with the `xopen()` or `load_file()` command. However, NEURON has a built-in editor that is closely related, if not identical, to MicroEMACS (<http://uemacs.tripod.com/>), and which we will simply call "emacs." In this era of richly menued, GUI-based editing software, NEURON's tiny emacs editor is definitely showing its age, and noone would ever confuse it with the much more powerful EMACS that has Lisp-like syntax [Cameron,1991 #668]. Nonetheless, it is quite capable and has the advantage of having same "look and feel" on all platforms.

Like EMACS, emacs is a command-driven editor with modes and multiple buffers. What does this mean? Being "command-driven" means that there are no menus or buttons to click on with a mouse. Instead, special keystrokes, like the "keyboard shortcuts" in other editors, are used as commands to the editor. Having "modes" implies that some of these commands change how emacs responds to what you type, like the way many editors can be switched between "insert" and "replace" mode. The notion of "buffers" may seem strange, but if you have ever used any kind of editor or word processing software, then you have almost certainly worked with buffers even if you didn't realize it. When you open a file to edit it, a copy of the file's contents are placed in a buffer in the computer's memory, and you edit the contents of the buffer. When you save your work, the buffer is written back to the file.

In the following pages we describe the commands that are available in emacs. The keystrokes for these commands are represented with this shorthand notation:

notation	means
<code>^A</code>	press and hold the "control" (Ctrl) key while typing a or A
<code>^A B</code>	first type ^A, then type b or B
<code>Esc</code>	press and release the "escape" (Esc) key
<code>Esc-A</code>	press and release the "escape" (Esc) key, then type a or A
<code>Esc n A</code>	press and release the "escape" (Esc) key, type a number, then type a or A
<code>space</code>	press and release the space bar
<code>Tab</code>	press and release the tab key

Starting and stopping

Switching from hoc to emacs

Type the command `em` at the `oc>` prompt.

Returning from emacs to hoc

Type `^C`. The current edit buffer will be read into `hoc` and executed. If the `hoc` interpreter encounters a syntax error, NEURON will return to emacs with the cursor at the line where the parser failed. Note that `^C` only executes a buffer. It does not save any unsaved buffer. On exit from `hoc` you will be prompted for whether or not to save each unsaved buffer.

Killing the current command

`^G` stops the current command. Sorry, emacs has no "undo."

Moving the cursor

<code>^F</code>	Forward 1 character.*
<code>^B</code>	Backward 1 character.*
<code>ESC-F</code>	Forward 1 word.
<code>ESC-B</code>	Backward 1 word.
<code>^E</code>	To end of line.
<code>^A</code>	To start of line.
<code>^N</code>	Next line.
<code>^P</code>	Previous line.
<code>ESC-N</code>	To end of next paragraph.
<code>ESC-P</code>	To start of previous paragraph.
<code>^V</code>	Scroll down one screen.*
<code>ESC-V</code>	Scroll up one screen.*
<code>ESC-></code>	To end of buffer.*
<code>ESC-<</code>	To start of buffer.*

*--Under MSWindows the arrow keys, Page Down, Page Up, End, and Home keys may work.

Modes

There are five modes that can be used individually, in any combination, or not at all.

- `^X M` Add mode. At the prompt type the name of the mode to be added.
- `^X ^M` Delete mode. At the prompt type the name of the mode to be removed.

Mode name	Description
<code>over</code>	New typing replaces (overwrites) instead of inserting characters.
<code>cmode</code>	Automatic indenting of C programs.
<code>wrap</code>	Automatic word wrap (inserts return at right margin).
<code>exact</code>	Search using exact case matching.
<code>view</code>	View buffer without changing it (ignores commands to change buffer).

Deleting and inserting

- `^H` Delete previous character.
- `^D` Delete next character.
- `Esc-^H` Delete previous word.
- `Esc-D` Delete next word.
- `^K` Delete to end of line.
- `^O` Insert line.

Under MSWindows, the Insert key inserts a space, and the Delete key acts like `^H`.

Blocks of text: marking, cutting, and pasting

- `Esc-space` Set mark at cursor position. After a mark has been set, the "region" is the text between the mark and the current position of the cursor. The "region" changes dynamically as the cursor is moved around.
- `^X ^X` Exchange mark and cursor.
- `^W` Delete (cut) region between mark and cursor. The deleted text is kept in the "kill buffer," replacing whatever the kill buffer may have already contained.
- `Esc-W` Copy region between mark and cursor to the kill buffer.
- `^Y` Copy (paste) text from kill buffer to cursor location.

Searching and replacing

<code>^S</code>	Search forward. At the prompt enter the search string followed by Esc. Case sensitive if buffer is in "exact" mode.
<code>^R</code>	Search backward.
<code>Esc-R</code>	Start from present location and replace every instance of the first string (the "search string") with the second string.
<code>Esc-Q</code>	Like <code>Esc-R</code> but asks for user's approval. User may reply with <code>y</code> or press the space bar to replace and continue <code>!</code> to replace the rest <code>n</code> to reject this replacement and continue to the next occurrence of the search string <code>^G</code> to stop <code>.</code> to stop and return to the starting point <code>?</code> for a list of options.

Text formatting and other tricks

<code>Esc-C</code>	Capitalize word.
<code>Esc-U</code>	Change word to upper case.
<code>^X ^U</code>	Change region to upper case.
<code>Esc-L</code>	Change word to lower case.
<code>^X ^L</code>	Change region to lower case.
<code>^T</code>	Transpose character at cursor with prior character.
<code>^Q</code>	"Quote" next character (allows entry of control codes into text).
<code>^X ^U</code>	Change region to upper case.
<code>Esc n ^X F</code>	Set right margin to column <code>n</code> .
<code>Esc-J</code>	Justify the paragraph so it fits between the margins.
<code>Esc n Tab</code>	Set right margin to column <code>n</code> .
<code>^X =</code>	Show line number, character count, and size of buffer.
<code>^X !</code>	Send a command to the OS.

Buffers and file I/O

When emacs first starts, it comes up with only one buffer, and that buffer is empty until you enter some text or type a command that loads a file. You can also have more than one buffer, each containing its own text, and you can copy text between them. To help keep things from getting mixed up, each buffer has a name. The first buffer is automatically called `main`.

It is a good idea to break large programs into many files and edit them in separate buffers. Buffers larger than 100 lines may take a noticeable time to interpret.

<code>^X B</code>	Switch to another buffer. You will be prompted for the name of a buffer. If the name you type doesn't exist, a new buffer will be created and given that name.
<code>^X K</code>	Delete a non-displayed buffer.
<code>^X ^B</code>	Show a directory of the buffers in a window.
<code>^X X</code>	Switch to the next buffer in the directory.
<code>ESC-^X 1</code>	Clear the buffer directory window.
<code>^X ^F</code>	Create a new buffer and load a file into it. You will be prompted for the name of the file. If the file doesn't already exist, the buffer will still be created with that name but it will be empty.
<code>^X ^R</code>	Read a file into the current buffer. Pre-existing contents of the current buffer are overwritten.
<code>^X ^I</code>	Insert a file into the current buffer at the location of the cursor.
<code>^X ^V</code>	Read a file into the current buffer and set the buffer to view mode.
<code>^X ^S</code>	Write current buffer to disk, using the name of the buffer as the name of the file. Overwrites pre-existing file with same name.
<code>^X ^W</code>	Write current buffer to disk. You will be prompted for a file name. This name will also become the name of the current buffer.
<code>^C</code>	Exit from emacs to hoc. The current buffer is passed to the hoc interpreter. For more details see Returning from emacs to hoc under Starting and stopping above.
<code>^X ^C</code>	Exit from both emacs and NEURON. You will be asked if you want to save buffers.

Windows

The emacs editor appears inside the terminal window from which NEURON was started. At first this window contains only one area for editing text, but you can split this into several different areas. These areas are called "windows" in standard EMACS and MicroEMACS terminology [Cameron,1991 #668]. Each window may show a different part of the same buffer, or part of a different one.

This use of the word "windows" was well-established years before the advent of microcomputer operating systems with windowed GUIs, in an era when the thought of copyrighting this word would have seemed ludicrous.

<code>^X 2</code>	Split the current window in two.
<code>^X 1</code>	Remove all but the current window.

<code>^X N</code>	Put the cursor in the next window.
<code>^X P</code>	Put the cursor in the previous window.
<code>^X ^</code>	Enlarge the current window.
<code>Esc-^V</code>	Scroll the other window down.
<code>Esc-^Z</code>	Scroll the other window up.

Macros and repeating commands

<code>Esc n -</code>	Repeats the next command <i>n</i> times.
<code>^U n</code>	Repeats the next command <i>n</i> times.
<code>^U</code>	Repeats the next command 4 times. Typing <code>^U</code> several times in a row repeats the next command by a power of 4, e.g. <code>^U ^U ^U ^N</code> moves the cursor down 64 lines.
<code>^X (</code>	Begin macro.
<code>^X)</code>	End macro.
<code>^X E</code>	Execute macro. All keystrokes entered from the beginning to the end of the macro are interpreted as emacs commands.

References

Cameron, D. and Rosenblatt, B. Learning GNU Emacs. O'Reilly & Associates, Inc., Sebastopol, CA. 1991.

Web site retrieved Oct. 27, 2004. MicroEMACS: Binaries, executables, documentation, source code. <http://uemacs.tripod.com/>.

Appendix A2 Index

E

emacs

blocks of text

copying to the kill buffer 3

cutting to the kill buffer 3

marking 3

pasting from the kill buffer 3

buffers 1, 4

cursor movement 2

entering 2

exiting 2

files 4

modes 1, 3

repeating commands

kill current command 2

macros 6

repeating commands 6

search and replace 4

send command to OS 4

text

deleting 3

formatting 4

inserting 3

windows 5