Emacs Buffers

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>	
Emacs Buffers	Emacs information and edited files are all held inside Emacs buffers. This table lists the commands you can use to list and manage buffers. PEL provides the pel-pkg-for-buffer customization group to control some aspect of Emacs buffers. Th user options are: • pel-use-uniquify : activates uniquify to that buffer names show the distinguishing directory after the file name, like this: fname dir • pel-use-ascii-table : activates the ascii-table external package. See Help/Info for the key binding. • pel-use-nhexl-mode : activates the nhexl-mode external package used to display and manipulate the content of the current buffer in hexadecimal. PEL also provides a Hydra that manipulates Emacs windows and buffers. See the Windows table for its description.			
Open this PDF file. See also: <u>∑ Help/Info</u>	<f11> b <f1></f1></f11>	(pel-help-pdf &optional OPEN-WEB-PAGE)	Open the local copy of the <u>Sauffers</u> PDF file unless a command prefix (like C-u) was used. In that case it opens the Github-hosted file instead.	
Customize PEL Buffer Support See also: ∑ Customize	<f11> b <f2></f2></f11>	(pel-customize-pel &optional OTHER-WINDOW)	Customize PEL Bookmark support: open PEL buffer support specific group. • If OTHER-WINDOW is non-nil (use C-u), display in other window.	
Customize Emacs & external package buffer support See also: Customize	<f11> b <f3></f3></f11>	(pel-customize-library &optional OTHER-WINDOW)	Customize Emacs and external packages related to buffer. This includes the following customize groups: Buffer-menu, ibuffer, minibuffer, nhexl. When a prefix argument (like C-u) opens the buffer inside another window. • Group belonging to files that have not yet been loaded are normally not accessible in Emacs and via the customize-group command. PEL, however, attempts to locate the file that defines a non-loaded customization group and will prompt you for loading the file if it finds it.	
Manage Buffers	The following comman	ds support buffer management.		
Open Buffer Menu	<c-f10></c-f10>	(buffer-menu-open)	Start key navigation of the buffer menu. Lists the buffers by major-mode when several buffers of the same major-mode are opened. This is the keyboard interface to <c-down-mouse-1></c-down-mouse-1>	
Toggle read-only status of buffer	• C-x C-q • <f11> b r</f11>	(read-only-mode &optional ARG)	When the buffer is in read-only mode the mode line shows '%%' on the left side, in the 'ch' area of "cs:ch-fr buf pos line (major minor)". The manual states: "For a read-only buffer, it shows '%*' if the buffer is modified, and '%%' otherwise." See also: the View Mode activating commands toward the end of this table. A buffer in View Mode cannot be modified. The View Mode may be used to ensure that no modifications are made to a buffer (visiting a file or not).	
Switch to next buffer	• C-x <right> • C-x C-<right> • <f11> b n</f11></right></right>	(next-buffer)	Switch to the next buffer displayed in the current window.	
Switch to previous buffer	• C-x <left> • C-x C-<left> • <f11> b p</f11></left></left>	(previous-buffer)	Switch to the previous buffer displayed in the current window.	
Show name of previous buffer in window	<f11> b P</f11>	(pel-show-window-previous-buffer)	Show the name of previous buffer used in the current window.	
Switch to previous buffer in window	<f11> b 1</f11>	(pel-switch-to-last-used – buffer)	Switch buffer in current window to the buffer previously seen in this window. Used twice returns to the same buffer.	
Switch to buffer	С-х в	(switch-to-buffer BUFFER-OR-NAME &optional NORECORD FORCE-SAME- WINDOW)	Switch window to display the previous, or another buffer (entered at prompt). Switch window to display the previous, or another buffer (entered at prompt). The invisible buffers have a name that start with a space. To see them type space and tab and a list of those buffers will appear before the list of visible buffers.	
List all buffers	С-ж С-в	(list-buffers & optional ARG) (ibuffer & optional OTHER-WINDOW-P NAME QUALIFIERS NOSELECT SHRINK FILTER-GROUPS FORMATS)	Display a list of existing buffers in a buffer named "*Buffer List*", the buffer displays information about all buffers and enters the <i>Buffer Menu Mode</i> . See the keystrokes for the Buffer Menu Mode below. The PEL package the 'ibuffer' function instead, which provides more functionality, working like dired, allowing to sort by name, size, mode, filtering by mode (hit return on the mode of a buffer). Type <f1> m to get the list of possible actions that can be done on the listed buffers.</f1>	
Clone buffer	<f11> b c</f11>	(clone-buffer &optional NEWNAME DISPLAY-FLAG)	Create and return a twin copy of the current buffer. • Unlike an indirect buffer, the new buffer can be edited independently of the old one (if it is not read-only). NEWNAME is the name of the new buffer. It may be modified by adding or incrementing <n> at the end as necessary to create a unique buffer name. • For example if buffer *Help* is opened it opens another one named *Help*<2> (or *Help*<3> if *Help*<2> already exists, etc)</n>	
Rename a buffer	<f11> b R</f11>	(rename-buffer NEWNAME &optional UNIQUE)	If UNIQUE argument is non-nil via C-u M-x rename-buffer, the name is auto generated to be unique.	
Rename buffer - use unique name	<f11> b U</f11>	(rename-uniquely)	Rename the current buffer by adding ' <number>' to the end. • Use this if you want multiple *Buffer* or *Info* buffers for example. • Example: StackExchange: How can I have multiple help buffer with different content</number>	
Kill current buffer See also: <u>▼ Windows</u>	• <f11> b k •</f11>	(kill-current-buffer)	 Kill (close) the current buffer. Does not prompt if there is no change in the buffer. PEL also provides a window management Hydra with ability to kill the current buffer. See <u>Windows</u> for more info. 	
Kill buffer	C-x k	(kill-buffer &optional BUFFER-OR-NAME)	Kill (close) the current buffer. • Always prompt to identify a buffer, current is identified. Press enter to kill the buffer.	
Kill current buffer and close window See also: Windows	• C-x 4 0 • <f7> k</f7>	(kill-buffer-and-window)	Kill the current buffer and delete the selected window. • PEL also provides a window management Hydra with ability to kill the current buffer and close windows in separate operations. See ▼ Windows for more info.	
Kill some buffer		(kill-some-buffers &optional LIST)	Kill some buffers. Asks the user whether to kill each one of them.	
Delete all windows of a specific buffer		(delete-windows-on &optional BUFFER-OR-NAME FRAME)	Deletes all windows showing BUFFER-OR-NAME, by calling 'delete-window' on those windows.	
Accumulating Text	Emacs provides the fo	llowing commands to insert text in buffer fr	om various sources.	
Append region to specified buffer	<f11> b M-a</f11>	(append-to-buffer BUFFER START END)	Append to specified BUFFER the text of the region. The text is inserted into that buffer before its point. BUFFER can be a buffer or the name of a buffer; this function will create BUFFER if it doesn't already exist.	
Prepend region to specified buffer	<f11> b M-p</f11>	(prepend-to-buffer BUFFER START END)	Prepend to specified BUFFER the text of the region. The text is inserted into that buffer after its point. BUFFER can be a buffer or the name of a buffer; this function will create BUFFER if it doesn't already exist.	

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>	
Copy region to specified buffer (replacing old content)	<f11> b C-c</f11>	(copy-to-buffer BUFFER START END)	Copy to specified BUFFER the text of the region. The text is inserted into that buffer, replacing existing text there. BUFFER can be a buffer or the name of a buffer; this function will create BUFFER if it doesn't already exist.	
Insert content of specified buffer at point	<f11> b i</f11>	(insert-buffer BUFFER)	Insert after point the contents of BUFFER. • Puts mark after the inserted text. • BUFFER may be a buffer or a buffer name.	
Append region's text to specified file	<f11> b f</f11>	(append-to-file START END FILENAME)	Append the contents of the region to the end of file FILENAME. • This does character code conversion and applies annotations like 'write-region' does.	
Indirect Buffers	As described in Emacs Indirect Buffer section, "an indirect buffer shares the text of some other buffer, called the base buffer of the indirect buffer. In some ways it is a buffer analogue of a symbolic link between files." The section also states: "One way to utilize indirect buffers is to display multiple views of an outline" (such as Org-Mode files). The following commands are available to manage indirect buffers.			
Create indirect buffer explicitly	<f11> b I m</f11>	(make-indirect-buffer BASE-BUFFER NAME &optional CLONE)	Create and return an indirect buffer for buffer BASE-BUFFER, named NAME. BASE-BUFFER should be a live buffer, or the name of an existing buffer. NAME should be a string which is not the name of an existing buffer. Optional argument CLONE non-nil means preserve BASE-BUFFER's state, such as major and minor modes, in the indirect buffer. CLONE nil means the indirect buffer's state is reset to default values.	
Create indirect buffer of current buffer	<f11> b I c</f11>	(clone-indirect-buffer NEWNAME DISPLAY-FLAG &optional NORECORD)	Create an indirect buffer that is a twin copy of the current buffer. • Give the indirect buffer name NEWNAME. Interactively, read NEWNAME from the minibuffer when invoked with a prefix arg. If NEWNAME is nil or if not called with a prefix arg, NEWNAME defaults to the current buffer's name. The name is modified by adding a ' <n>' suffix to it or by incrementing the N in an existing suffix. Trying to clone a buffer whose major mode symbol has a non-nil 'no-clone-indirect' property results in an error. • DISPLAY-FLAG non-nil means show the new buffer with 'pop-to-buffer'. This is always done when called interactively. • Optional third arg NORECORD non-nil means do not put this buffer at the front of the list of recently selected ones.</n>	
Create indirect buffer of current buffer in another window	• C-x 4 c • <f11> b I w</f11>	(clone-indirect-buffer-other-window NEWNAME DISPLAY-FLAG &optional NORECORD)	Like 'clone-indirect-buffer' but display in another window.	
Edit Binary file with hexl	Emacs provides the built-in hext mode to edit files in hexadecimal mode. • To use it you must: • use the hext-find-file to open the file in binary mode, or • use the hext-mode command to convert an already opened buffer. To exit this mode and go back to the original mode type C-c C-c			
Open a file in hexl-mode	<f11> f M-x</f11>	(hexl-find-file FILENAME)	Edit file FILENAME as a binary file in hex dump format. • Switch to a buffer visiting file FILENAME, creating one if none exists, and edit the file in the land.	
See also: <u>∑ File-mngt</u>			'hexl-mode'. Toggle the hexl mode: a mode for editing binary files in hex dump format.	
			behavior, but not all; also, you can exit Hexl mode and return to the previous mode using 'hexl-mode-exit'. This function automatically converts a buffer into the hexl format using the function 'hexlify-buffer'. Each line in the buffer has an "address" (displayed in hexadecimal) representing the offset into the file that the characters on this line are at and 16 characters from the file (displayed as hexadecimal values grouped every 'hexl-bits' bits, and as their ASCII values). If any of the characters (displayed as ASCII characters) are unprintable (control or meta characters) they will be replaced by periods.	
Insert a byte in decimal	C-M-d	(hexl-insert-decimal-char ARG)	Insert a character given by its decimal code ARG times at point.	
Insert a byte in octal	С-М-о	(hexl-insert-octal-char ARG)	Insert a character given by its octal code ARG times at point.	
Insert a byte in hex	C-M-x	(hexl-insert-hex-char ARG)	Insert a character given by its hexadecimal code ARG times at point.	
Goto 512-byte page start	С-М-а	(hexI-beginning-of-512b-page)	Go to beginning of 512 byte boundary.	
Goto to 512-byte page end	С-М-е	(hexl-end-of-512b-page)	Go to end of 512 byte boundary.	
Goto 1K end	C-x]	(hexl-end-of-1k-page)	Go to end of 1KB boundary.	
Goto 1K beginning	С-ж [(hexl-beginning-of-1k-page)	Go to beginning of 1KB boundary.	
Goto address entered in hexadecimal	M-g	(hexi-goto-hex-address HEX- ADDRESS)	Go to Hexl mode address (hex string) HEX-ADDRESS. • Signal error if HEX-ADDRESS is out of range.	
Goto to address entered in decimal	M-j	(hexi-goto-address ADDRESS)	Go to hexl-mode (decimal) address ADDRESS. • Signal error if ADDRESS is out of range.	
Exit hexl mode	C-c C-c	(hexl-mode-exit &optional ARG)	Exit Hexl mode, returning to previous mode. • With arg, don't unhexlify buffer.	
Hexadecimal Editing with nhexI	The <a href="https://www.new.new.new.new.new.new.new.new.new.</td></tr><tr><td>Toggle buffer between normal and hex display</td><td><f11> b x</td><td>(nhexl-mode &optional ARG)</td><td>Toggle minor mode to edit files via hex-dump format. Requires the nhexl-mode package nhexl-mode			
Activate Hex nibble editing mode	<f11> b X</f11>	(nhexl-nibble-edit-mode &optional ARG)	Minor mode to edit the hex nibbles in 'nhexl-mode'. A Note: only works after nhexl-mode has been activated once. Requires the nhexl-mode package activated when pel-use-nhexl user option is t.	
Buffer View Mode	Several commands (view-buffer, etc, see at top of this table) activate the View Mode for a buffer where the buffer is essentially read-only and special commands are available.			
View buffer - no modification allowed	<f11> b v</f11>	(view-buffer BUFFER &optional EXIT-ACTION)	View BUFFER in View mode, returning to previous buffer when done. • Emacs commands editing the buffer contents are not available; instead, a special set of commands (mostly letters and punctuation) are defined for moving around in the buffer. • Space scrolls forward, Delete scrolls backward. • For a list of all View commands, type H or h while viewing. See the View Mode command list below.	

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>
		Show this message.	
D -	_	provide prefix arguments. negative prefix argument.	
<		move to the beginning of buffer.	
>		move to the end of buffer. scroll so that buffer end is at last line of window.	
o S		scroll forward "page size" lines. With prefix scroll forward	rd prefix lines.
		scroll backward "page size" lines. With prefix scroll back	ward prefix lines.
z W		like SPC but with prefix sets "page size" to prefix. like DEL but with prefix sets "page size" to prefix.	
d			alf page size" to prefix lines and scrolls forward that much.
u			half page size" to prefix lines and scrolls backward that much.
У		scroll forward one line. With prefix scroll forward prefix li scroll backward one line. With prefix scroll backward pre	
F	?	revert-buffer if necessary and scroll forward. Use this to	
= %		prints the current line number. goes prefix argument (default 100) percent into buffer.	
g		goes to line given by prefix argument (default first line).	
		set the mark.	
6 x		exchanges point and mark. return to mark and pops mark ring. Mark ring is pushed	at start of every successful search and when jump to line occurs.
		The mark is set on jump to buffer start or end.	
m,		save current position in character register. go to position saved in character register.	
s		do forward incremental search.	
r		do reverse incremental search.	went need that the basis of the second
/		searches forward for regular expression, starting after cu ! means search for a line with no match for regexp.	rrent page. ! and @ have a special meaning at the beginning of the regexp:
		@ means start search at beginning (end for backward	
\\n		searches backward for regular expression, starting beforesearches forward for last regular expression.	e current page.
p		searches lotward for last regular expression.	
q	I	quit View mode, restoring this window and buffer to prev	ious state. q is the normal way to leave view mode.
e	_	exit View mode but stay in current buffer. Use this if you	started viewing a buffer (file) and find out you want to edit it.
E	e	This command restores the previous read-only status exit View mode, and make the current buffer editable ever	
Q		quit View mode, restoring all windows to previous state.	,
C		quit View mode and maybe switch buffers, but don't kill t quit View mode, kill current buffer and go back to other b	
	ĺ	quit view mode, kill current buller and go back to other b	uner.
		and C depends on how view-mode was entered.	rame, or M-x dired-view-file (M-x view-file, M-x view-file-other-window, M-x
		rame, or the Dired mode v command), then ${f q}$ will try to kil	
•			-buffer-other-window, M-x view-buffer-other frame, M-x view-file, M-x view-file-
		r M-x view-file-other-frame, then c, q and C will return to	
Buffer Menu Mode	he list of buffers is	s shown inside its own buffer, *Buffer List* when (list-buff	er) is executed. This buffer support the following commands.
	The full list of Its	whindings is sucilable via the CETS in low	
	<u> </u>	by bindings is available via the <f1> m key. L uses (ibuffer) for the C-x C-b key binding, so the list o</f1>	f commands and key bindings that are available differ.
	? : Get I		: Immediately
	g : Upda	ate buffer list	: immediately
- Compicte the list		buffer in list uffer in list	: immediately : immediately
		buffer in list	: immediately
		ious buffer in list ious buffer in list	: immediately : immediately
	C-d : mark	k buffer for deletion	: deleted when pressing x
		k buffer for deletion k buffer for deletion	: deleted when pressing x : deleted when pressing x
		e buffer	: saved when pressing x
		re to previous line, remove all marks on buffer nove a specific mark from all buffers	: immediately if just after marking : immediately if just after marking
		ark all marks on buffer	: immediately
		cute marked commands (delete buffers marked for deletic k buffer as un-modifiable	n) : immediately : immediately
•	% : togg	le read-only	: immediately
		lay emacs in full emacs screen play this buffer & next in horizontal window	: immediately : immediately
	o : repla	ace other (next) window with this buffer	: immediately
		k buffer to be displayed in windows lay buffers marked with in as many windows as required	: when pressing v : immediately
	·	buffer list	: immediately
	S' - Save the mark		
(4)	A' - View the mark	ed buffers in the selected frame.	
	H' - View the mark	(ed buffers in another frame	
'H	H' - View the mark V' - Revert the mar	rked buffers.	
'H 'V 'T	V' - Revert the mai T' - Toggle read-or	rked buffers. nly state of marked buffers.	
'	V' - Revert the mai T' - Toggle read-or L' - Toggle lock sta D' - Kill the marked	rked buffers. nly state of marked buffers. ate of marked buffers. d buffers.	
1-1 1-1 1-1 1-1 1-1 1-1 1-1	V' - Revert the mai T' - Toggle read-or L' - Toggle lock sta D' - Kill the marked M-s a C-s' - Do ind	rked buffers. nly state of marked buffers. ate of marked buffers. d buffers. cremental search in the marked buffers.	
ነ- የ ነገ ነር ነር ነ ነ	V' - Revert the man T' - Toggle read-or L' - Toggle lock sta D' - Kill the marked M-s a C-s' - Do ind M-s a C-M-s' - Ise r' - Replace by reg	rked buffers. nly state of marked buffers. ate of marked buffers. d buffers. cremental search in the marked buffers. sarch for regexp in the marked buffers. gexp in each of the marked buffers.	
14 V 17 12 16 18 18 18 18 18 18 18 18 18 18 18 18 18	V' - Revert the maint' - Toggle read-or L' - Toggle lock sta D' - Kill the marked M-s a C-s' - Do ind M-s a C-M-s' - Ise r' - Replace by reg Q' - Query replace	rked buffers. nly state of marked buffers. ate of marked buffers. d buffers. cremental search in the marked buffers. earch for regexp in the marked buffers.	
'.' 'V 'T 'L 'C 'N 'Y 'Y 'Y 'Y 'Y	V' - Revert the mai T' - Toggle read-or L' - Toggle lock sta D' - Kill the marked M-s a C-s' - Do inf M-s a C-M-s' - Ise r' - Replace by reg Q' - Query replace l' - As above, with P' - Print the mark	rked buffers. nly state of marked buffers. at each of marked buffers. d buffers. cremental search in the marked buffers. earch for regexp in the marked buffers. each of the marked buffers. e in each of the marked buffers. a regular expression.	ation (accur)
', ', ', ', ', ', ', ', ', ', ', ', ', '	V' - Revert the main T' - Toggle read-or L' - Toggle lock state D' - Kill the marked M-s a C-s' - Do inn M-s a C-M-s' - Ise r' - Replace by reg Q' - Query replace Q' - Query replace T' - As above, with P' - Print the mark O' - List lines in all	rked buffers. nly state of marked buffers. ate of marked buffers. d buffers. cremental search in the marked buffers. earch for regexp in the marked buffers. exp in each of the marked buffers. e in each of the marked buffers. a regular expression.	ction 'occur').
'.' 'V 'T '! 'L '' 'N 'N 'Y '' '' '' '' '' '' '' '' 'Y 'Y 'Y 'Y	V' - Revert the man T' - Toggle read-or L' - Toggle lock sta D' - Kill the marked M-s a C-s' - Do ind M-s a C-M-s' - Ise r' - Replace by reg Q' - Query replace l' - As above, with P' - Print the mark O' - List lines in all X' - Pipe the conte N' - Replace the or	rked buffers. nly state of marked buffers. ate of marked buffers. d buffers. cremental search in the marked buffers. earch for regexp in the marked buffers. earch for regexp in the marked buffers. earch of the marked buffers earch earch of the marked buffers to a shell command. contents of the marked buffers with the output of a shell command.	
ነ- የ የ የ የ የ የ የ የ የ የ የ የ የ የ የ የ የ የ የ	V' - Revert the main transparent to the marked that a C-s' - Do info M-s a C-M-s' - Ise a C-M-s' - As above, with P' - Print the mark O' - List lines in all X' - Pipe the conte M' - Replace the cot! - Run a shell con E' - Evaluate a forr	rked buffers. nly state of marked buffers. ate of marked buffers. d buffers. cremental search in the marked buffers. earch for regexp in the marked buffers. earch for regexp in the marked buffers. earch of the marked buffers. a regular expression. ded buffers. I marked buffers which match a given regexp (like the functors of the marked buffers to a shell command. contents of the marked buffers with the output of a shell command with the buffer's file as an argument. m in each of the marked buffers. This is a very flexible co	
'	V' - Revert the main T' - Toggle read-or L' - Toggle lock state D' - Kill the marked M-s a C-s' - Do inn M-s a C-M-s' - Ise r' - Replace by reg Q' - Query replace I' - As above, with P' - Print the mark O' - List lines in all X' - Pipe the conte N' - Replace the city - Replace the city - Replace the city - Evaluate a forrusing (read-only-mind)	rked buffers. nly state of marked buffers. ate of marked buffers. d buffers. cremental search in the marked buffers. earch for regexp in the marked buffers. earch for regexp in the marked buffers. earch of the marked buffers. ear negular expression. ear d buffers. I marked buffers which match a given regexp (like the functors of the marked buffers to a shell command. contents of the marked buffers with the output of a shell command with the buffer's file as an argument. m in each of the marked buffers. This is a very flexible colode 1) as the input form.	ommand.
\frac{1}{1} \\ \frac{1} \\ \fr	V' - Revert the main T' - Toggle read-or L' - Toggle lock state D' - Kill the marked M-s a C-s' - Do ind M-s a C-M-s' - Ise r' - Replace by reg Q' - Query replace l' - As above, with P' - Print the mark O' - List lines in all X' - Pipe the conte N' - Replace the coll P' - Run a shell conte l' - Run a shell conte in girling (read-only-min) W' - As above, but k' - Remove the m	rked buffers. nly state of marked buffers. ate of marked buffers. d buffers. cremental search in the marked buffers. earch for regexp in the marked buffers. earch for regexp in the marked buffers. earch of the marked buffers. a regular expression. ded buffers. I marked buffers which match a given regexp (like the functors of the marked buffers to a shell command. contents of the marked buffers with the output of a shell command with the buffer's file as an argument. m in each of the marked buffers. This is a very flexible co	ommand. For example, if you want to make all of the marked buffers read-only, try

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>
IBuffer Mode command (2)	** c' - Change the mark 'u' - Unmark the buffer 'DEL' - Unmark the pre 'M-DEL' - Unmark buff 'U' - Unmark all marke '* M' - Mark all "unsave '* m' - Mark all modifie '* s' - Mark all buffers v '* e' - Mark all buffers v '* r' - Mark all read-on! '* /' - Mark buffers in 'I' - Mark buffers older 'd' - Mark buffers older 'd' - Mark buffers by '% m' - Mark buffers by '% m' - Mark buffers by '% m' - Mark buffers by '% f' - Mark buffers by 'Mark buffers by 'M	y marked buffers, and mark all unmarked is used on marked buffers. at point. vious buffer. ers marked with MARK. d buffers. major mode. d' buffers. This means that the buffer is mid buffers, regardless of whether they have whose name begins and ends with '''. which have an associated file, but that file of buffers. irelp-mode', 'apropos-mode', etc. than 'ibuffer-old-time'. point for deletion. their name, using a regexp. their filename, using a regexp. their content, using a regexp.	odified, and has an associated file. an associated file.
IBuffer Mode command (3)	'/ RET' - Add a filter by a '/ M' - Add a filter by d '/ m' - Add a filter by bu '/ n' - Add a filter by bu '/ c' - Add a filter by bu '/ b' - Add a filter by bu 'M-x ibuffer-filter-by-di '/ f' - Add a filter by file '/ '- Add a filter by file '/ '- Add a filter by must '- Add a filter by bu '/ c' - Switch to previou '/ s' - Save the current '/ r' - Switch to previou '/ a' - Add saved filters '/ &' - Replace the top	major mode now in use. erived mode. iffer name. iffer content. isename. rectory' - Add a filter by directory name. name. extension. idified buffers. arbitrary Lisp predicate. iffer size. ecial buffers. iffers visiting files. filters with a name. sly saved filters. to current filters. two filters with their logical AND. wo filters with their logical OR. filter. itense of the top filter. topmost filter.	chosen by completion.
IBuffer Mode command (4)	Filter group command '/ g' - Create filter group '/ P' - Remove top filte 'TAB' - Move to the net 'M-p' - Move to the pretion of the current '/ R' - Save the current '/ R' - Restore previous '/ X' - Delete previous system	p from filters. r group. xt filter group. evious filter group. filter groups. groups with a name. sly saved groups.	
IBuffer Mode command (5)	Sorting commands: ',' - Rotate between the 's i' - Reverse the curre 's a' - Sort the buffers to strict of the strict of t	exicographically. by the file name. by last viewing time. by size.	
IBuffer Mode command (6)	" - Change the current 'SPC' - Move point to to 'C-p' - Move point to to 'h' - This help. 'e' - View the differenct 'RET' - View the buffer 'o' - As above, but in a	t display format. the next line. ne previous line. es between this buffer and its associated f on this line. nother window. but don't select the new window.	whether buffers that match 'ibuffer-maybe-show-predicates' should be displayed. ile.