

GNU Readline

From Wikipedia, the free encyclopedia

GNU Readline is a software library that provides line-editing and history capabilities for interactive programs with a command-line interface, such as Bash. It is currently maintained by Chet Ramey as part of the GNU Project.

It allows users to move the text cursor, search the command history, control a kill ring (a more flexible version of a copy/paste clipboard) and use tab completion on a text terminal.

Contents

- 1 Keyboard shortcuts
- 2 Choice of the GPL as GNU Readline's license
 - 2.1 Implications of GNU Readline's GPL license
- 3 Sample code
- 4 Notes and references
- 5 External links

GNU Readline

Original author(s)	Brian Fox
Developer(s)	Chet Ramey
Initial release	1989
Stable release	6.3 / February 26, 2014
Written in	C
Operating system	Various
Type	Library
License	GNU General Public License
Website	Official website (http://cnswww.cns.cwru.edu/php/chet/readline/rltop.html)

Keyboard shortcuts

Readline key bindings are taken from the text editor Emacs, but can be customized. As a cross-platform library, readline allows applications on various systems to exhibit identical line-editing behavior.

On some systems, Esc must be used instead of Alt, because the Alt shortcut conflicts with another shortcut. For example, pressing Alt + f in Xfce's terminal emulator window does not move the cursor forward one word, but activates "File" in the menu of the terminal window, unless that is disabled in the emulator's settings.

- Tab ↵ : Autocompletes from the cursor position.
- Ctrl + a : Moves the cursor to the line start (equivalent to the key Home).
- Ctrl + b : Moves the cursor back one character (equivalent to the key ←).
- Ctrl + c : Sends the signal SIGINT to the current task, which aborts and closes

it.

- **Ctrl + d**
 - Sends an EOF marker, which (unless disabled by an option) closes the current shell (equivalent to the command `exit`). (Only if there is no text on the current line)
 - If there is text on the current line, deletes the current character (then equivalent to the key **Delete**).
- **Ctrl + e** : (end) moves the cursor to the line end (equivalent to the key **End**).
- **Ctrl + f** : Moves the cursor forward one character (equivalent to the key **→**).
- **Ctrl + g** : Abort the research and restore the original line.
- **Ctrl + h** : Deletes the previous character (same as backspace).
- **Ctrl + i** : Equivalent to the tab key.
- **Ctrl + j** : Equivalent to the enter key.
- **Ctrl + k** : Clears the line content after the cursor and copies it into the clipboard.
- **Ctrl + l** : Clears the screen content (equivalent to the command `clear`).
- **Ctrl + n** : (next) recalls the next command (equivalent to the key **↓**).
- **Ctrl + o** : Executes the found command from history, and fetch the next line relative to the current line from the history for editing.
- **Ctrl + p** : (previous) recalls the prior command (equivalent to the key **↑**).
- **Ctrl + r** : (reverse search) recalls the last command including the specified character(s). A second **Ctrl + r** recalls the next anterior command that corresponds to the search
- **Ctrl + s** : Go back to the next more recent command of the research (beware to not execute it from a terminal because this command also launches its XOFF). If you changed that XOFF setting, use **Ctrl + q** to return.
- **Ctrl + t** : Transpose the previous two characters.
- **Ctrl + u** : Clears the line content before the cursor and copies it into the clipboard.
- **Ctrl + v** : If the next input is also a control sequence, type it literally (e. g. *
Ctrl + v **Ctrl + h** types "**^H**", a literal backspace.)
- **Ctrl + w** : Clears the word before the cursor and copies it into the clipboard.
- **Ctrl + x** **Ctrl + e** : Edits the current line in the `$EDITOR` program, or vi if undefined.
- **Ctrl + x** **Ctrl + r** : Read in the contents of the `inputrc` file, and incorporate any

bindings or variable assignments found there.

- `Ctrl + x Ctrl + u` : Incremental undo, separately remembered for each line.
- `Ctrl + x Ctrl + v` : Display version information about the current instance of Bash.
- `Ctrl + x Ctrl + x` : Alternates the cursor with its old position. (C-x, because x has a crossing shape).
- `Ctrl + y` : (yank) adds the clipboard content from the cursor position.
- `Ctrl + z` : Sends the signal SIGTSTP to the current task, which suspends it. To execute it in background one can enter `bg`. To bring it back from background or suspension `fg ['process name or job id']` (foreground) can be issued.
- `Ctrl + _` : Incremental undo, separately remembered for each line.
- `Alt + b` : (backward) moves the cursor backward one word.
- `Alt + c` : Capitalizes the character under the cursor and moves to the end of the word.
- `Alt + d` : Cuts the word after the cursor.
- `Alt + f` : (forward) moves the cursor forward one word.
- `Alt + l` : Lowers the case of every character from the cursor's position to the end of the current word.
- `Alt + r` : Cancels the changes and puts back the line as it was in the history.
- `Alt + u` : Capitalizes every character from the cursor's position to the end of the current word.
- `Alt + .` : Insert the last argument to the previous command (the last word of the previous history entry).

Choice of the GPL as GNU Readline's license

GNU Readline is notable for being a free software library which is licensed under the GNU General Public License (GPL) instead of the GNU Lesser General Public License (LGPL). Free software libraries are often licensed under the LGPL, for example, the GNU C Library, GNU gettext and FLTK.

A developer of an application who chooses to link to an LGPL licensed library when building a new application is required to have the LGPL licensed library which it uses remain under the LGPL when distributing the combined resulting application. The part of the combined application excluding the LGPL licensed library can remain under the original license.^[1] This is in contrast to a developer choosing to use a GPL licensed library to create a new application, in which case the *entire* combined resulting application is required to be licensed under the GPL when distributed, to comply with section 5 of the GPL.^{[2][3]}

Implications of GNU Readline's GPL license

An important example of an application changing its licensing to comply with the copyleft conditions of GNU Readline is CLISP, an implementation of Common Lisp.

Originally released in 1987, it changed to the GPL license in 1992,^[4] after an email exchange between one of CLISP's original authors, Bruno Haible, and Richard Stallman, in which Stallman argued^[5] that the linking of readline in CLISP meant that Haible was required to re-license CLISP under the GPL if he wished to distribute the implementation of CLISP which used readline.^[6]

Alternative command line editing libraries which are permissively licensed can be used by software projects which want to implement command line editing functionality, but wish to remain under a permissive license. For example, the Glasgow Haskell Compiler uses Haskeline (which is licensed under the 3 clause BSD license).

Sample code

The following code is in C and must be compiled with a `-lreadline` compiler flag:

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <readline/readline.h>
#include <readline/history.h>

int main()
{
    char* input, shell_prompt[100];

    // Configure readline to auto-complete paths when the tab key is hit.
    rl_bind_key('\t', rl_complete);

    for(;;) {
        // Create prompt string from user name and current working directory.
        snprintf(shell_prompt, sizeof(shell_prompt), "%s:%s $ ", getenv("USER"), getcwd(NULL, 1024));

        // Display prompt and read input (NB: input must be freed after use)...
        input = readline(shell_prompt);

        // Check for EOF.
        if (!input)
            break;

        // Add input to history.
        add_history(input);

        // Do stuff...

        // Free input.
        free(input);
    }
    return 0;
}
```

Notes and references

- "GNU Lesser General Public License". *The GNU Lesser General Public License v3.0 - GNU Project*. Free Software Foundation. 2007. Retrieved 2011-09-03.
- "GNU General Public License". *The GNU General Public License v3.0 - GNU Project*. Free

Software Foundation. 2007. Retrieved 2011-09-03.

3. "Frequently Asked Questions about the GNU licenses". *Frequently Asked Questions about the GNU Licenses - GNU Project*. Free Software Foundation. 2010. Retrieved 2011-09-03.
4. "CLISP copyright notice". CLISP repository. 1992. Retrieved 2011-09-03.
5. "Why CLISP is under GPL". CLISP repository. 1992. Retrieved 2011-09-03.
6. "License - why GNU GPL?". *Frequently Asked Questions (With Answers) about CLISP*. CLISP team. Retrieved 2011-09-03.

External links

- GNU readline homepage (<https://cnswww.cns.cwru.edu/php/chet/readline/rltop.html>)

Retrieved from "https://en.wikipedia.org/w/index.php?title=GNU_Readline&oldid=724188284"

Categories: GNU Project software | Free software programmed in C
| Text user interface libraries

- This page was last modified on 7 June 2016, at 17:50.
- Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.