About the list

An average terminal session accepts a range of control keys. This is a list for a system running [Bash](http://wiki.linuxquestions.org/wiki/Bash), the (by far) most commonscenario for a linux user. The list assumes no one has changed the default behavior.

The key is the key to press, ^A meaning C-a or Ctrl+A. A key followed by a '!' means the key is very important to know, one followed by '+' means it's not important as such, but lifechangingly useful. One should try out the various other commands to see if they are worth knowing or not. (Editors: please do not add !/+ to your own pet favorite, it cheapens the system)

The function is what the key does. Multiple uses are separated by slashes (/).

The User is the part of the system that is responsible for handling the key. The terminal and the shell have a symbiotic relationship, often both will be required to complete an action. The User is in any case the part one can modify to remap the key. If the key has multiple functions, users will be separated by slashes as well.

Different terminals can have different settings for certain keys, Backspace especially (some uses ^? for example).

The list

|  |  |  |
| --- | --- | --- |
| Key (and importance) | Function | User |
| ^A | GNU [Screen](http://wiki.linuxquestions.org/wiki/Screen) multifunction key/Go to the beginning of the line | Screen / Bash |
| ^B | One char backwards (like left-arrow) | Bash |
| ^C ! | Interrupt a running program (SIGINT) | Terminal |
| ^D ! | Ends input for interactive programs. The program will act as if you [piped](http://wiki.linuxquestions.org/wiki/Pipe) a file to it that now has reached the end, and will usually terminate itself. For this reason it's also a simple way of logging out (an alternative to exit). | Terminal (though the current program will do what it please when it encounters the event) |
| ^E | Go to the end of the line | Bash |
| ^F | One char forward (like right-arrow) | Bash |
| ^G | Abort the current editing command (for composite command such as M-5 Up-Arrow (M-5 meaning Meta-5 or Alt+5) which would repeat the Up-Arrow action five times. Pressing ^G after M-5 would abort it, meaning a subsequent Up-Arrow key would just function as normal and just act once) | Bash |
| ^H | Erase the last character | Terminal (this is the Backspace key) |
| ^I | Move the cursor to the next column divisible by eight (for alignment)/autocomplete commands and file names (typing | Terminal / Bash (This is the Tab key) |
| ^J | Move the cursor to the next line\*/ Execute the current command line | Terminal / Bash (this is the Return key) |
| ^K | Remove the parts of the command line after the cursor | Bash |
| ^L + | Redraw the screen | Bash (and most [curses](http://wiki.linuxquestions.org/wiki/Curses) applicationsor textmode programs and games that draw graphics using characters and symbols) |
| ^M | Move the cursor to the next line\* / Execute the current command line | Terminal / Bash (this is the Enter key) |
| ^N | Move to next (more recent) history entry (same as down-arrow) | Bash |
| ^O | Execute the current command and start editing the next history entry (for redoing multiple commands that need small edits) | Bash |
| ^P | Move to previous (earlier) history entry (same as up-arrow) | Bash |
| ^Q | Allow a terminal to continue display after being paused with ^S | Terminal |
| ^R + | Search history backwards, allows one to search the history by keyword rather than hit ^P or Up-arrow to locate an old command. | Bash |
| ^S | Temporarily stop the terminal from displaying more information. It will continue by pressing ^Q | Terminal |
| ^T | Swap positions of the current and previous chars, and advance one char forward on the command line | Bash |
| ^U | Delete from the current point backwards, complement to ^K | Bash |
| ^V | Insert the next character literally, one can use ^V^S to type a ^S character rather than stop the terminal. | Terminal |
| ^W | Delete one word backwards | Terminal |
| ^X | Bash multifunction key. One command is ^X^E which will invoke the users editor on the current command and execute it when they exit the editor (very useful for long command lines) | Bash |
| ^Y | Copy previously killed text (^W, ^K, ^U, ...) onto the current command line (functioning in effect as cut&paste) | Bash |
| ^Z ! | Suspend the current job and get back to the shell | Terminal |
| ^[ | Mostly a composing character meaning Meta/Alt. ^[5 will repeat the next action five times. | Bash (this is the Escape key) |
| ^/ | Abort a running program (SIGQUIT) | Terminal |
| ^] | Search char forward, pressing ^] V will place the cursor on the next V on the current command line | Bash |
| ^\_ | Undo, restore the command line to what it was before the last edit (since \_ is shift-minus on US layouts, this command is actually Ctrl-Shift-Minus) | Bash |

 \* Strictly, and under certain terminal modes (stty -onlcr), ^J (newline) will move the the next line at the same position, while ^M (carriage return) will move to the beginning of the same line.

Changing shortcuts

**When to change keys**

It is possible to change most of the shortcuts to suit one's need. Changing terminal keys such as ^C, ^D and ^Z is not recommend however, this will cause much confusion and grief when one uses other terminals or let others use one's own (imagine not finding ^C in an emergency!). It is in fact used as a prank on those who leave their terminals unattended.

Bash keys such as ^K/^U can be customized, since advanced editing commands are a mere convenience, not vital to the operation of the system. There are many, many other bash commands, some which do not even have key combinations bound do them by default. The [manual](http://wiki.linuxquestions.org/wiki/Man) mentions them all.

**How to change keys**

Bash keys are changed by editing ~/.inputrc or using bind (not to be confused with the name server). Terminal keys can be changed with [stty](http://wiki.linuxquestions.org/wiki/Stty). The [manual](http://wiki.linuxquestions.org/wiki/Man) for bash and stty will give directions and possible i