daverdave.com

Apple / Mac | BASH | Disk / IO / testing and info | Drupal | HTTP/HTML | LAMP | Linux Notes | MySQL | NetApp | Networking | Perl | Puppet | RHEL DOCS | Recipes | Redhat Virtualization KVM ZEN QEMU | SVN cheat sheet | Scripts (awk/sed) | Syntax for scripting languages | TOMCAT | VMware | Veritas | Windows | Zenoss | vi / vim

BASH shortcuts | BASH- test operators | bash arrays | bash history | bash progress indicator | bash script to get memory statistics from Linux process | bash: converting time to epoch and back | color codes | how to find files newer than a certain date in linux | how to rename a list of files with to a new name in bash | options (getopt and getopts) | prompt escape characters | script to add number in a file | script to check time stamp of a file | script to monitor CPU and print iostat output | script to reverse lines | script to see if a directory has changed using stat | simple bash counter | special bash variables | subroutine in bash | sweet bash script for replacing grants | using cut and awk in bash | using while, read, echo to print out a file without spawning another process

Search

Search this site:

[   ] [Search]

contents

▶ Apple / Mac
▼ BASH
   ◦ BASH shortcuts
   ◦ BASH- test operators
   ◦ bash arrays
   ◦ bash history
   ◦ bash progress indicator
   ◦ bash script to get memory statistics from Linux process
   ◦ bash: converting time to epoch and back
   ◦ color codes
   ◦ how to find files newer than a certain date in linux
   ◦ how to rename a list of files with to a new name in bash
   ◦ options (getopt and getopts)
   ◦ prompt escape characters

Home

bash shortcuts

print key bindings: "bind -P"

Ctrl-a Jump to BOL

Ctrl-e Jump to EOL

Ctrl-w Delete word left from cursor

Ctrl-u Delete from BOL to cursor

Ctrl-k Delete from cursor to EOL

Alt-d Delete forward word

Ctrl-(dash) Undo

Alt-t Swap current word with previous word

http://ruslanspivak.com/2010/09/20/bash-key-bindings/

http://www.bigsmoke.us/readline/shortcuts

Bash, which is the default shell in Linux contains a whole lot of key bindings which makes it really easy to use . The most commonly used shortcuts are listed below :

_____CTRL Key Bound_____
Ctrl + a - Jump to the start of the line
Ctrl + b - Move back a char
Ctrl + c - Terminate the command
Ctrl + d - Delete from under the cursor

Ctrl + e - Jump to the end of the line
Ctrl + f - Move forward a char
Ctrl + k - Delete to EOL
Ctrl + l - Clear the screen
Ctrl + r - Search the history backwards
Ctrl + R - Search the history backwards with multi occurrence
Ctrl + u - Delete backward from cursor
Ctrl + xx - Move between EOL and current cursor position
Ctrl + x @ - Show possible hostname completions
Ctrl + z - Suspend/ Stop the command
_____ALT Key Bound_____
Alt + < - Move to the first line in the history
Alt + > - Move to the last line in the history
Alt + ? - Show current completion list
Alt + * - Insert all possible completions
Alt + / - Attempt to complete filename
Alt + . - Yank last argument to previous command
Alt + b - Move backward
Alt + c - Capitalize the word
Alt + d - Delete word
Alt + f - Move forward
Alt + l - Make word lowercase
Alt + n - Search the history forwards non-incremental
Alt + p - Search the history backwards non-incremental
Alt + r - Recall command
Alt + t - Move words around
Alt + u - Make word uppercase
Alt + back-space - Delete backward from cursor

----------------More Special Keybindings-------------------

Here "2T" means Press TAB twice

$ 2T - All available commands(common)
$ (string)2T - All available commands starting with (string)
$ /2T - Entire directory structure including Hidden one
$ 2T - Only Sub Dirs inside including Hidden one
$ *2T - Only Sub Dirs inside without Hidden one
$ ~2T - All Present Users on system from "/etc/passwd"
$ $2T - All Sys variables
$ @2T - Entries from "/etc/hosts"
$ =2T - Output like ls or dir

Bash Shortcuts Quick Reference

Ctrl-a	Move to the start of the line.
Ctrl-e	Move to the end of the line.
Ctrl-b	Move back one character.
Alt-b	Move back one word.

Navigation

- **Contact**
- **Forums**
- **Search**

Ctrl-f	Move forward one character.
Alt-f	Move forward one word.
Alt-] *x*	Where x is any character, moves the cursor forward to the next occurance of x.
Alt-Ctrl-] *x*	Where x is any character, moves the cursor backwards to the previous occurance of x.
Ctrl-u	Delete from the cursor to the beginning of the line.
Ctrl-k	Delete from the cursor to the end of the line.
Ctrl-w	Delete from the cursor to the start of the word.
Esc-Del	Delete previous word (may not work, instead try Esc followed by Backspace)
Ctrl-y	Pastes text from the clipboard.
Ctrl-l	Clear the screen leaving the current line at the top of the screen.
Ctrl-x Ctrl-u	Undo the last changes. Ctrl-_ does the same
Alt-r	Undo all changes to the line.
Alt-Ctrl-e	Expand command line.
Ctrl-r	Incremental reverse search of history.
Alt-p	Non-incremental reverse search of history.
!!	Execute last command in history
!abc	Execute last command in history beginning with abc
!abc:p	Print last command in history beginning with abc
!*n*	Execute *n*th command in history
!$	Last argument of last command
!^	First argument of last command
^abc^xyz	Replace first occurance of abc with xyz in last command and execute it

http://www.lugod.org/mailinglists/archives/vox-tech/2003-06/msg00182.html

----

```
hi all,

it took me about 3 years of searching to find the answer to this.

i can't remember the guy's name who wrote this, but i've found the

results to be very, very satisfactory.  i'm quite pleased with these

bindings.
```

basically, you're in emacs mode, so ^w, ^e, ^a, ^u work, as do up/down
arrow keys.

once you press escape, you're in vi command mode. however, the arrow
keys actually work! another annoyance with vi mode was that ^l didn't
work either without pressing escape first. that's fixed by these
bindings as well.

i put all this in a file /etc/skell/.bash_bindings and ~/.bash_bindings.
then i source it everytime an interactive shell is started.

hth,
pete

```
## .bash_bindings: source this file from your .bashrc or .bash_profile
#
# For those who want to use Vi bindings in bash, this corrects a
# few annoyances:
#
# 1) up and down arrows retrieve history lines even in insert mode
# 2) left and right arrows work in insert mode
# 3) Ctrl-A and Ctrl-E work how you expect if you have had to
#    live in Emacs mode in the past.
# 4) So does Ctrl-D.
# 5) And Ctrl-L.

## Command-mode bindings
# Ctrl-A or Home: insert at line beginning like in emacs mode
 bind -m vi-command 'Control-a: vi-insert-beg'
# Ctrl-E or End: append at line end like in emacs mode
 bind -m vi-command 'Control-e: vi-append-eol'
# to switch to emacs editing mode
 bind -m vi-command '"ZZ": emacs-editing-mode'

## Insert-mode bindings
# up arrow or PgUp: append to previous history line
 bind -m vi-insert '"\M-[A": ""'
 bind -m vi-insert '"\M-[5~": ""'
 bind -m vi-insert 'Control-p: previous-history'
# dn arrow or PgDn: append to next history line
 bind -m vi-insert '"\M-[B": ""'
```

```
 bind -m vi-insert '"\M-[6~": ""'
 bind -m vi-insert 'Control-n: next-history'
# Ctrl-A: insert at line start like in emacs mode
 bind -m vi-insert 'Control-a: beginning-of-line'
# Ctrl-E: append at line end like in emacs mode
 bind -m vi-insert 'Control-e: end-of-line'
# Ctrl-D: delete character
 bind -m vi-insert 'Control-d: delete-char'
# Ctrl-L: clear screen
 bind -m vi-insert 'Control-l: clear-screen'

## Emacs bindings
# Meta-V: go back to vi editing
 bind -m emacs '"\ev": vi-editing-mode'

## Specify vi editing mode
 set -o vi
_____
```