

ab		UNIX Shell:	All shells (bash, tcsh, sh, ksh, csh, zsh)
<b>Description:</b>	Benchmarks your Apache server by sending requests to it.		
<b>Syntax:</b>	ab [options...] [http:// hostname[:port] /path]		
-k	Enables multiple requests in one HTTP session. This is called HTTP KeepAlive		
-i	Uses an HTTP HEAD, which replaces the GET method. Not combinable with POST.		
-n requests	Specifies the number of requests to run in the benchmark.		
-t timelimit	Specifies the time in seconds to use benchmarking. This will set -n to 50,000.		
-c concurrency	Specifies the number of requests to perform simultaneously where default is 1.		
-p postfile	Specifies a file to send in any HTTP POST requests to the Apache server.		
-A user:password	Provides the server with user name and password entered with a colon between.		
-p user:password	NOTE: This will be sent whether the server needs it or not, as uuencoded data.		
-C name=value	Runs the same procedure as the -A switch, but is used with proxy servers.		
-p string	Adds a "Cookie:" to the request. The option is repeatable.		
-T content-type	Appends more headers to the request. The argument is usually in value:value form.		
-v verbosity	Specifies the content-type header for usage with POST data.		
-w output HTML	Specifies verbosity level.		
-x string	Shows results in HTML tables. (default is two columns wide, white background).		
-y string	Specifies the attributes for table.		
-z string	Specifies the attributes for tr.		
-V	Specifies the attributes for td or th.		
-h	Shows version information.		
-H attribute	Shows help information.		
[http://]hostname[:port] /path	Adds arbitrary header line. The option is repeatable.		
<b>File Name:</b>	ab	<b>Directory:</b>	/usr/apache/bin
		<b>Type:</b>	External

  

alloc		UNIX Shell:	TC shell (tcsh)
<b>Description:</b>	Shows how much memory is used and is free.		
<b>Syntax:</b>	alloc		

  

appttrace		UNIX Shell:	All shells (bash,tcsh,sh,ksh,csh,zsh)
<b>Description:</b>	Traces function calls that a specific program makes to shared libraries.		
<b>Syntax:</b>	appttrace [ options... ] command { command arguments... }		
-f	Follows any child processes created by fork and shows trace output for each process.		
-o outfile	Specifies an output file instead of STDERR.		
-F [!]tracefrom	<b>An exclamation point (!) before the argument removes the argument from the trace:</b>		
-T [!]traceto	Traces calls from a list of shared objects.		
-t [!]call	Traces calls to a list of shared objects.		
-v [!]call	Traces function calls.		
command	Shows call traces in verbose format.		
arguments...	Specifies the command that you want to trace.		
<b>File Name:</b>	appttrace	<b>Directory:</b>	/usr/bin/
		<b>Type:</b>	External

  

autoload		UNIX Shell:	Z shell (zsh)
<b>Description:</b>	Marks the function for auto loading.		
<b>Syntax:</b>	autoload function		
function	Specifies the function to be marked for auto loading.		

  

bash		UNIX Shell:	Bash shell (bash)
<b>Description:</b>	Is the GNU Bourne-again shell. It's compatible with the Bourne shell, with additional functions from Korn and C-shell.		
<b>Syntax:</b>	bash [ options... ] { file }		
--dump-po-strings	Is same as -D, but output is in GNU portable objects format (po).		

--dump-strings	Is same as -D.
--help	Shows help information.
--login	Invoked as a login shell.
--noediting	Doesn't read command line by reading the GNU readline library.
--noprofile	Doesn't read any of the initialization files when invoked as a login shell.
--norc	Doesn't read the personal initialization file (.bashrc).
--posix	Invokes shell to match to posix standard.
--rcfile <i>file</i>	Uses the specified file instead of the personal initialization file (.bashrc).
--restricted	Invokes as restricted shell.
--verbose	Verbose mode. Shows more information.
--version	Shows version information.
-c <i>string</i>	Reads command specified by string.
-D	Shows all double-quoted strings preceded by \$ on STDOUT.
-r	Runs shell as restricted shell.
-i	Makes shell interactive.
-s	Reads commands from STDIN.
-a	Marks modified and created variables for export. <u>NOTE: This option can also be set inside a script.</u>
-b	Informs the user asynchronously when a background job completes.
-e	Exits the shell if a command returns a non-zero exit status.
-f	Disables file name generation.
-h	Tracks aliases for the first encountered command.
-k	Places all keywords arguments in the environment for a command.
-m	Runs background job in separate process group. Shows a line when completed.
-n	Reads commands, but doesn't run them.
-o { <i>option</i> }	Is used to set option. If no option is given, shows current status. See below for option.
allexport	Is the same as -a.
braceexpand	Is the same as -B.
emacs	Uses an Emacs style inline editor for the command line.
errexit	Is the same as -e.
hashall	Is the same as -h.
histexpand	Is the same as -H.
history	Enables command history.
ignoreeof	Will not let the shell exit on EOF. The <code>exit</code> command must be used.
keyword	Is the same as -k.
monitor	Is the same as -m.
noclobber	Is the same as -C.
noexec	Is the same as -n.
noglob	Is the same as -f.
notify	Is the same as -b.
nounset	Is the same as -u.
onecmd	Is the same as -t.
physical	Is the same as -P.
posix	Changes to POSIX standard behavior where bash differs.
privileged	Is the same as -p.
verbose	Is the same as -v.
vi	Uses a vi-style inline editor for the command line.
xtrace	Is the same as -x.
-p	Turns on privileged mode.
-t	Runs one command and exits.
-u	Substitutes, treats unset variables as an error.
-v	Shows input lines as they are read.
-x	Show commands and their arguments when they are executed.
-B	Performs brace expansion.
-C	Doesn't allow existing files to be overwritten by shell > redirection operator.
-H	Enables ! style history substitution.

-P	Uses physical directory structure instead of symbolic link structure. <u>NOTE: If + is used instead of -, the flag is turned off.</u>				
--	Sets positional parameters with the remaining of the line.				
-	Turns off -x and -v flag and doesn't continue to examining arguments for flag.				
file	A file containing bash commands.				
<b>File Name:</b>	bash	<b>Directory:</b>	/usr/bin/	<b>Type:</b>	External

bind		UNIX Shell:	Bash shell (bash)
<b>Description:</b>	Shows or bind key sequences to a readline function or macro.		
<b>Syntax:</b>	bind options...		
-m <i>keymap</i>	Specifies the keymap to be affected by the subsequent bindings.		
-l	Shows a list of all readline function names.		
-p	Shows a list of all readline function names and bindings for readline init files.		
-P	Show a list of current readline function names and bindings.		
-v	Shows a list of readline variable names and values for readline init files.		
-V	Show a list current readline variable names and values.		
-s	Shows a list of readline key sequences bound to macros for readline init files.		
-S	Shows a list of readline key sequences bound to macros and the strings they output.		
-f <i>file</i>	Reads key bindings from file.		
-q <i>function</i>	Queries witch keys is bind to a function.		
-u <i>function</i>	Erases all key bindings to function.		
-r <i>keyseq</i>	Erases any binding for the key sequence.		
-x <i>keyseq:function</i>	Assigns function to the key sequence.		

bindkey		UNIX Shell:	TC shell (tcsh)
<b>Description:</b>	Binds keys to an edit command in the shell. Without any options, shows all bindings.		
<b>Syntax:</b>	bindkey [ options... ] { <i>key</i> } { <i>command</i> }		
-l	Shows a short descriptions for each editor command.		
-d	Sets all key bindings to default for the current editor.		
-e	Sets all key bindings to standard GNU Emacs-like bindings.		
-v	Sets all key bindings to standard vi-like bindings.		
-a	Shows or change key bindings.		
-b	Interprets a key as a control character.		
-k	Interprets a key as a symbolic arrow key name.		
-r	Removes a key binding.		
-c	Interprets a command as an internal or external command instead of an editor one.		
-s	Interprets a command as a string to use as input when a key is pressed.		
--	Specifies that the following option is a key even if it starts with -.		
-u	Shows help information.		
<i>key</i>	Specifies the key to use.		
<i>command</i>	Specifies the editor command to assign to a key.		

builtin		UNIX Shell:	Bash shell (bash), Z shell (zsh)
<b>Description:</b>	Forces the use of a shell built-in command.		
<b>Syntax:</b>	builtin <i>command</i> { <i>arguments...</i> }		
<i>command</i>	Specifies a shell builtin command to run.		
<i>arguments...</i>	Specifies arguments to command.		

builtins		UNIX Shell:	TC shell (tcsh)
<b>Description:</b>	Shows all built-in commands in tcsh.		
<b>Syntax:</b>	builtins		

bunzip2		UNIX Shell:	All shells (bash,tcsh,sh,ksh,csh,zsh)
<b>Description:</b>	Decompresses bzip2 compressed files.		
<b>Syntax:</b>	bunzip2 [ options... ] { <i>filenames...</i> }		

-k	Keeps the input files.
-f	Overwrites existing output files (default is no overwrite).
-v	Shows more information.
-V	Shows version and license information.
-L	Shows software version and license.
-s	Is used to limit the amount of memory to use.
filenames...	Specifies a file or files to decompress.
<b>File Name:</b>	bunzip2
<b>Directory:</b>	/usr/bin/
<b>Type:</b>	External

busstat		UNIX Shell:	All shells (bash,tcsh,sh,ksh,csh,zsh)
<b>Description:</b>	Collects statistics and show buss performance counters.		
<b>Syntax:</b>	busstat option busstat [ options... ] { interval count }		
-e device -l  -a -n -w device picnr=events... -r device interval count	<p><b>The following options cannot be combined:</b></p> <p>Shows a list of events that are supported by the specified device. Shows a list of devices that support performance counters.</p> <p><b>The following options can be combined:</b></p> <p>Shows only absolute counter values. Shows no output title. Specifies the device to write counters for. Specifies the counters and events to show; uses a comma-separated list. Specifies a device to read and show values for. Specifies an interval for collecting data. Specifies how many times to collect data.</p>		
<b>File Name:</b>	busstat	<b>Directory:</b>	/usr/bin/
		<b>Type:</b>	External

bye		UNIX Shell:	Z shell (zsh)
<b>Description:</b>	Leaves the shell. It's the same as exit.		
<b>Syntax:</b>	bye		

bzip2		UNIX Shell:	All shells (bash,tcsh,sh,ksh,csh,zsh)
<b>Description:</b>	Decompresses a bzip2 compressed file to STDOUT.		
<b>Syntax:</b>	bzip2 [ -s ] { files... }		
-s file...	<p>Is used to limit the memory amount allowed for decompression.</p> <p>Is the file or files to decompress.</p>		
<b>File Name:</b>	bzip2	<b>Directory:</b>	/usr/bin/
		<b>Type:</b>	External

bzip2		UNIX Shell:	All shells (bash,tcsh,sh,ksh,csh,zsh)
<b>Description:</b>	Is used to compress, decompress or test bzip2 files.		
<b>Syntax:</b>	bzip2 [ options... ] { filenames... }		
-c -d -f -k -s -t -v -z -L -1 -9 --repetitive-fast --repetitive-best filenames...	<p>Compresses or decompresses to STDOUT. Forces decompression. Overwrites output files (default is no overwrite). Is used to keep the input files during compression or decompression. Is used to limit the memory usage when compressing, decompressing, or testing. Tests the integrity on the file or files. Shows more information. Forces compression. Shows license terms and conditions. Is used to set the block size at 100K to 900K during compression. Compresses repetitive blocks faster. Compresses repetitive blocks better. Specifies the file or files to compress.</p>		
<b>File Name:</b>	bzip2	<b>Directory:</b>	/usr/bin/
		<b>Type:</b>	External

bzip2recover			UNIX Shell:	All shells (bash,tcsh,sh,ksh,csh,zsh)	
Description:		Recovers data from a damaged bzip2 compressed file.			
Syntax:		bzip2recover filename			
filename		Specifies the file to recover.			
File Name:	bzip2recove r	Directory:	/usr/bin/	Type:	External

clinfo			UNIX Shell:	All shells (bash,tcsh,sh,ksh,csh,zsh)	
Description:		Shows information about the cluster configuration on the node the command was executed.			
Syntax:		clinfo [ options... ]			
-h		Shows the highest node number in the cluster configuration.			
-n		Shows the number of the node where clinfo was executed.			
File Name:	clinfo	Directory:	/usr/sbin/	Type:	External

complete		UNIX Shell:	TC shell (tcsh)
Description:	Declares how a command should be completed.		
Syntax:	complete { command } complete command word/pattern/list[:select ]/{ suffix/ }		
command	Specifies the command to set up a completion for; or to show a completion for.		
word	Specifies which word, that is relative to the current, is to be completed.		
c	Specifies the pattern that must match the beginning of current word.		
C	Includes pattern when completing the word.		
n	Specifies that pattern that must match the beginning of previous word.		
N	Specifies the pattern that must match the word previous to the previous word (2 words previous).		
p	Specifies position-dependent match. pattern is a numeric range.		
pattern	Specifies the pattern to use.		
list	Specifies the possible completions.		
a	Uses aliases.		
b	Uses bindings (editor commands).		
c	Uses commands internal or external.		
C	Uses external commands that begin with the specified path.		
d	Uses directories.		
D	Uses directories that begin with the specified path.		
e	Uses environment variables.		
f	Uses file names.		
F	Uses file names that begin with the specified path.		
g	Uses group names.		
j	Uses jobs.		
l	Uses limits.		
n	Doesn't use anything.		
s	Uses shell variables.		
S	Uses signals.		
t	Uses plain text files.		
T	Uses plain text files that begin with the specified path.		
v	Uses any variables.		
u	Uses usernames.		
x	Shows select when list-choices is used.		
X	Uses completions.		
\$var	Uses word from the variable \$var		
(...)	Uses word from the specified list.		
`command`	Uses word from the output from the specified command.		
select	Specifies an extra pattern to use as a selection for the last three option above.		
suffix	Specifies a single character to add to the completion.		