

Quick Guide to bash Commands

IN THIS APPENDIX

Viewing the bash built-in commands

Reviewing GNU additional shell commands

Looking at bash environment variables

As you've seen throughout this book, the bash shell contains lots of features and thus has lots of commands available. This appendix provides a concise guide to allow you to quickly look up a feature or command that you can use from the bash command line or from a bash shell script.

Reviewing Built-In Commands

The bash shell includes many popular commands built into the shell. This provides for faster processing times when using these commands. Table A-1 shows the built-in commands available directly from the bash shell.

TABLE A-1 bash Built-In Commands

Command	Description
:	Expands listed arguments and redirects as specified
.	Reads and executes commands from a designated file in the current shell
alias	Defines an alias for the specified command
bg	Resumes a job in background mode
bind	Binds a keyboard sequence to a readline function or macro
break	Exits from a for, while, select, or until loop
builtin	Executes the specified shell built-in command
caller	Returns the context of any active subroutine call
cd	Changes the current directory to the specified directory

Continues

TABLE A-1 (continued)

Command	Description
command	Executes the specified command without the normal shell lookup
compgen	Generates possible completion matches for the specified word
complete	Displays how the specified words would be completed
compgpt	Changes options for how the specified words would be completed
continue	Resumes the next iteration of a <code>for</code> , <code>while</code> , <code>select</code> , or <code>until</code> loop
declare	Declares a variable or variable type
dirs	Displays a list of currently remembered directories
disown	Removes the specified jobs from the jobs table for the process
echo	Displays the specified string to <code>STDOUT</code>
enable	Enables or disables the specified built-in shell command
eval	Concatenates the specified arguments into a single command, and executes the command
exec	Replaces the shell process with the specified command
exit	Forces the shell to exit with the specified exit status
export	Sets the specified variables to be available for child shell processes
fc	Selects a list of commands from the history list
fg	Resumes a job in foreground mode
getopts	Parses the specified positional parameters
hash	Finds and remembers the full pathname of the specified command
help	Displays a help file
history	Displays the command history
jobs	Lists the active jobs
kill	Sends a system signal to the specified process ID (PID)
let	Evaluates each argument in a mathematical expression
local	Creates a limited-scope variable in a function
logout	Exits a login shell
mapfile	Reads <code>STDIN</code> lines and puts them into an indexed array
popd	Removes entries from the directory stack
printf	Displays text using formatted strings
pushd	Adds a directory to the directory stack

<code>pwd</code>	Displays the pathname of the current working directory
<code>read</code>	Reads one line of data from STDIN, and assigns it to a variable
<code>readarray</code>	Reads STDIN lines, and puts them into an indexed array
<code>readonly</code>	Reads one line of data from STDIN, and assigns it to a variable that can't be changed
<code>return</code>	Forces a function to exit with a value that can be retrieved by the calling script
<code>set</code>	Sets and displays environment variable values and shell attributes
<code>shift</code>	Rotates positional parameters down one position
<code>shopt</code>	Toggles the values of variables controlling optional shell behavior
<code>source</code>	Reads and executes commands from a designated file in the current shell
<code>suspend</code>	Suspends the execution of the shell until a SIGCONT signal is received
<code>test</code>	Returns an exit status of 0 or 1 based on the specified condition
<code>times</code>	Displays the accumulated user and system shell times.
<code>trap</code>	Executes the specified command if the specified system signal is received
<code>type</code>	Displays how the specified word would be interpreted if used as a command
<code>typeset</code>	Declares a variable or variable type
<code>ulimit</code>	Sets a limit on the specified resource for system users
<code>umask</code>	Sets default permissions for newly created files and directories
<code>unalias</code>	Removes the specified alias
<code>unset</code>	Removes the specified environment variable or shell attribute
<code>wait</code>	Waits for the specified process to complete, and returns the exit status

The built-in commands provide higher performance than external commands, but the more built-in commands that are added to a shell, the more memory it consumes with commands that you may never use. The bash shell also contains external commands that provide extended functionality for the shell. These are discussed in the following section.

Looking at Common bash Commands

In addition to the built-in commands, the bash shell utilizes external commands to allow you to maneuver around the filesystem and manipulate files and directories. Table A-2 shows the common external commands you'll want to use when working in the bash shell.

TABLE A-2 The bash Shell External Commands

Command	Description
bzip2	Compresses using the Burrows-Wheeler block sorting text compression algorithm and Huffman coding
cat	Lists the contents of the specified file
chage	Changes the password expiration date for the specified system user account
chfn	Changes the specified user account's comment information
chgrp	Changes the default group of the specified file or directory
chmod	Changes system security permissions for the specified file or directory
chown	Changes the default owner of the specified file or directory
chpasswd	Reads a file of login name and password pairs and updates the passwords
chsh	Changes the specified user account's default shell
clear	Removes text from a terminal emulator or virtual console terminal
compress	Original Unix file compression utility
coproc	Spawns a subshell in background mode and executes the designated command
cp	Copies the specified files to an alternate location
crontab	Initiates the editor for the user's crontable file, if allowed
cut	Removes a designated portion of each specified file's lines
date	Displays the date in various formats
df	Displays current disk space statistics for all mounted devices
du	Displays disk usage statistics for the specified file path
emacs	Invokes the emacs text editor
file	Views the file type of the specified file
find	Performs a recursive search for files
free	Checks available and used memory on the system
gawk	Streams editing using programming language commands
grep	Searches a file for the specified text string
gedit	Invokes the GNOME Desktop editor
getopt	Parses command options including long options
groups	Displays group membership of the designated user
groupadd	Creates a new system group
groupmod	Modifies an existing system group
gzip	The GNU Project's compression using Lempel-Ziv compression

<code>head</code>	Displays the first portion of the specified file's contents
<code>help</code>	Displays the help pages for bash built-in commands
<code>killall</code>	Sends a system signal to a running process based on process name
<code>kwrite</code>	Invokes the KWrite text editor
<code>less</code>	Advanced viewing of file contents
<code>link</code>	Creates a link to a file using an alias name
<code>ln</code>	Creates a symbolic or hard link to a designated file
<code>ls</code>	Lists directory contents
<code>makewhatis</code>	Creates the whatis database allowing man page keyword searches
<code>man</code>	Displays the man pages for the designated command or topic
<code>mkdir</code>	Creates the specified directory under the current directory
<code>more</code>	Lists the contents of the specified file, pausing after each screen of data
<code>mount</code>	Displays or mounts disk devices into the virtual filesystem
<code>mv</code>	Renames a file
<code>nano</code>	Invokes the nano text editor
<code>nice</code>	Runs a command using a different priority level on the system
<code>passwd</code>	Changes the password for a system user account
<code>ps</code>	Displays information about the running processes on the system
<code>pwd</code>	Displays the current directory
<code>renice</code>	Changes the priority of a running application on the system
<code>rm</code>	Deletes the specified file
<code>rmdir</code>	Deletes the specified directory
<code>sed</code>	Streams line editing using editor commands
<code>sleep</code>	Pauses bash shell operation for a specified amount of time
<code>sort</code>	Organizes data in a data file based on the specified order
<code>stat</code>	Views the file statistics of the specified file
<code>sudo</code>	Runs an application as the root user account
<code>tail</code>	Displays the last portion of the specified file's contents
<code>tar</code>	Archives data and directories into a single file
<code>top</code>	Displays the active processes, showing vital system statistics
<code>touch</code>	Creates a new empty file or updates the timestamp on an existing file
<code>umount</code>	Removes a mounted disk device from the virtual filesystem
<code>uptime</code>	Displays information on how long the system has been running
<code>useradd</code>	Creates a new system user account

Continues

TABLE A-2 (continued)

Command	Description
userdel	Removes an existing system user account.
usermod	Modifies an existing system user account
vi	Invokes the vim text editor
vmstat	Produces a detailed report on memory and CPU usage on the system
whereis	Displays a designated command's files, including binary, source code, and man pages
which	Finds the location of an executable file
who	Displays users currently logged into system
whoami	Displays the current user's username
xargs	Takes items from STDIN, builds commands, and executes the commands
zip	Unix version of the Windows PKZIP program

You can accomplish just about any task you need to on the command line using these commands.

Assessing Environment Variables

The bash shell also utilizes many environment variables. Although environment variables aren't commands, they often affect how shell commands operate, so it's important to know the shell environment variables. Table A-3 shows the default environment variables available in the bash shell.

TABLE A-3 bash Shell Default Environment Variables

Variable	Description
*	Contains all the command line parameters as a single text value
@	Contains all the command line parameters as separate text values
#	The number of command line parameters
?	The exit status of the most recently used foreground process
-	The current command line option flags
\$	The process ID (PID) of the current shell

!	The PID of the most recently executed background process
0	The name of the command from the command line
_	The absolute pathname of the shell
BASH	The full filename used to invoke the shell
BASHOPTS	Enabled shell options in a colon-separated list
BASHPID	The current bash shell's process ID
BASH_ALIASES	An array containing the currently used aliases.
BASH_ARGC	The number of parameters in the current subroutine
BASH_ARGV	An array containing all the command line parameters specified
BASH_CMDS	An array containing the internal hash table of commands
BASH_COMMAND	The name of the command currently being executed
BASH_ENV	When set, each bash script attempts to execute a startup file defined by this variable before running.
BASH_EXECUTION_STRING	The command used in the -c command line option
BASH_LINENO	An array containing the line numbers of each command in the script
BASH_REMATCH	An array containing text elements that match a specified regular expression
BASH_SOURCE	An array containing source file names for the declared functions in the shell
BASH_SUBSHELL	The number of subshells spawned by the current shell
BASH_VERSIONINFO	A variable array that contains the individual major and minor version numbers of the current instance of the bash shell
BASH_VERSION	The version number of the current instance of the bash shell
BASH_XTRACEFD	When set to a valid file descriptor integer, trace output is generated and separated from diagnostic and error messages. The file descriptor must have set -x enabled.
COLUMNS	Contains the terminal width of the terminal used for the current instance of the bash shell
COMP_CWORD	An index into the variable COMP_WORDS, which contains the current cursor position

Continues

TABLE A-3 (continued)

Command	Description
COMP_KEY	The completion invocation character keyboard key
COMP_LINE	The current command line
COMP_POINT	The index of the current cursor position relative to the beginning of the current command
COMP_TYPE	The completion type integer value
COM_WORDBREAKS	A set of characters used as word separators when performing word completion
COMP_WORDS	A variable array that contains the individual words on the current command line
COMPREPLY	A variable array that contains the possible completion codes generated by a shell function
COPROC	A variable array that holds file descriptors for an unnamed coprocess' I/O
DIRSTACK	A variable array that contains the current contents of the directory stack
EMACS	When set, the shell assumes it's running in an emacs shell buffer and disables line editing.
ENV	When the shell is invoked in POSIX mode, each bash script attempts to execute a startup file defined by this variable before running.
EUID	The numeric effective user ID of the current user
FCEDIT	The default editor used by the <code>fc</code> command
FIGIGNORE	A colon-separated list of suffixes to ignore when performing file name completion
FUNCNAME	The name of the currently executing shell function
FUNCNEST	The maximum level for nesting functions
GLOBIGNORE	A colon-separated list of patterns defining the set of filenames to be ignored by file name expansion
GROUPS	A variable array containing the list of groups of which the current user is a member
histchars	Up to three characters that control history expansion

HISTCMD	The history number of the current command
HISTCONTROL	Controls what commands are entered in the shell history list
HISTFILE	The name of the file to save the shell history list (<code>.bash_history</code> by default)
HISTFILESIZE	The maximum number of lines to save in the history file
HISTIGNORE	A colon-separated list of patterns used to decide which commands are ignored for the history file
HISTSIZE	The maximum number of commands stored in the history file
HISTTIMEFORMAT	When set, determines the format string for the history file entries' time stamps
HOSTFILE	Contains the name of the file that should be read when the shell needs to complete a hostname
HOSTNAME	The name of the current host
HOSTTYPE	A string describing the machine the bash shell is running on
IGNOREEOF	The number of consecutive EOF characters the shell must receive before exiting. If this value doesn't exist, the default is 1.
INPUTRC	The name of the readline initialization file (The default is <code>.inputrc</code> .)
LANG	The locale category for the shell
LC_ALL	Overrides the <code>LANG</code> variable, defining a locale category
LC_COLLATE	Sets the collation order used when sorting string values
LC_CTYPE	Determines the interpretation of characters used in file name expansion and pattern matching
LC_MESSAGES	Determines the locale setting used when interpreting double-quoted strings preceded by a dollar sign
LC_NUMERIC	Determines the locale setting used when formatting numbers
LINENO	The line number in a script currently executing
LINES	Defines the number of lines available on the terminal
MACHTYPE	A string defining the system type in <i>cpu-company-system</i> format

Continues

TABLE A-3 (continued)

Command	Description
MAILCHECK	How often (in seconds) the shell should check for new mail (default is 60)
MAPFILE	Array variable containing the <code>mapfile</code> command's read text; used only when no variable name is given
OLDPWD	The previous working directory used in the shell
OPTERR	If set to 1, the bash shell displays errors generated by the <code>getopts</code> command.
OSTYPE	A string defining the operating system the shell is running on
PIPESTATUS	A variable array containing a list of exit status values from the processes in the foreground process
POSIXLY_CORRECT	If set, bash starts in POSIX mode.
PPID	The process ID (PID) of the bash shell's parent process
PROMPT_COMMAND	If set, the command to execute before displaying the primary prompt
PS1	The primary command line prompt string
PS2	The secondary command line prompt string
PS3	The prompt to use for the <code>select</code> command
PS4	The prompt displayed before the command line is echoed if the bash <code>-x</code> parameter is used.
PWD	The current working directory
RANDOM	Returns a random number between 0 and 32767. Assigning a value to this variable seeds the random number generator.
READLINE_LINE	The readline line buffer contents
READLINE_POINT	The current readline line buffer's insertion point position
REPLY	The default variable for the <code>read</code> command
SECONDS	The number of seconds since the shell was started. Assigning a value resets the timer to the value.
SHELL	The shell's full pathname
SHELLOPTS	A colon-separated list of enabled bash shell options
SHLVL	Indicates the shell level, incremented by 1 each time a new bash shell is started

TIMEFORMAT	A format specifying how the shell displays time values
TMOUT	The value of how long (in seconds) the <code>select</code> and <code>read</code> commands should wait for input. The default of 0 indicates to wait indefinitely.
TMPDIR	When set to a directory name, the shell uses the directory as a location for temporary shell files.
UID	The numeric real user ID of the current user

You display the environment variables using the `set` built-in command. The default shell environment variables set at boot time can and often do vary between different Linux distributions.