

Linux Cli Cheatsheet

Linux Command Line Cheat Sheet

Bash Commands

Command	Description
<code>uname -a</code>	Show system and kernel
<code>head -n1 /etc/issue</code>	Show distribution
<code>mount</code>	Show mounted filesystems
<code>date</code>	Show system date
<code>uptime</code>	Show uptime
<code>whoami</code>	Show your username
<code>man *command*</code>	Show manual for <i>command</i>

Bash Shortcuts

Shortcut	Description
CTRL-c	Stop current command
CTRL-z	Sleep program
CTRL-a	Go to start of line
CTRL-e	Go to end of line
CTRL-u	Cut from start of line
CTRL-k	Cut to end of line
CTRL-r	Search history
!!	Repeat last command
!*abc*	Run last command starting with <i>abc</i>
!*abc*:p	Print last command starting with <i>abc</i>
!\$	Last argument of previous command
ALT-.	Last argument of previous command
!*	All arguments of previous command
^abc^123	Run previous command, replacing <i>abc</i> with <i>123</i>

Bash Variables

Variable	Description
<code>env</code>	Show environment variables
<code>echo \$NAME</code>	Output value of <code>\$NAME</code> variable
<code>export NAME=value</code>	Set <code>\$NAME</code> to <i>value</i>
<code>\$PATH</code>	Executable search path
<code>\$HOME</code>	Home directory
<code>\$SHELL</code>	Current shell

IO Redirection

Command	Description
<code>cmd < file</code>	Input of <code>cmd</code> from <code>file</code>
<code>cmd1 <(cmd2)</code>	Output of <code>cmd2</code> as file input to <code>cmd1</code>
<code>cmd > file</code>	Standard output (stdout) of <code>cmd</code> to <code>file</code>
<code>cmd > /dev/null</code>	Discard stdout of <code>cmd</code>

Command	Description
cmd >> file	Append stdout to file
cmd 2> file	Error output (stderr) of cmd to file
cmd 1>&2	stdout to same place as stderr
cmd 2>&1	stderr to same place as stdout
cmd &> file	Every output of cmd to file
cmd refers to a command.	

Pipes

Command	Description
cmd1 \ cmd2	stdout of cmd1 to cmd2
cmd1 \ & cmd2	stderr of cmd1 to cmd2

Command Lists

Command	Description
cmd1 ; cmd2	Run cmd1 then cmd2
cmd1 \&\& cmd2	Run cmd2 if cmd1 is successful
cmd1 \ \ cmd2	Run cmd2 if cmd1 is not successful
cmd &	Run cmd in a subshell

Directory Operations

Command	Description
pwd	Show current directory
mkdir dir	Make directory dir
cd dir	Change directory to dir
cd ..	Go up a directory
ls	List files

ls Options

Option	Description
-a	Show all (including hidden)
-R	Recursive list
-r	Reverse order
-t	Sort by last modified
-S	Sort by file size
-l	Long listing format
-1	One file per line
-m	Comma-separated output
-Q	Quoted output

Search Files

Command	Description
grep pattern files	Search for <i>pattern</i> in <i>files</i>
grep -i	Case insensitive search
grep -r	Recursive search
grep -v	Inverted search
grep -o	Show matched part of file only
find /dir/ -name name*	Find files starting with <i>name</i> in <i>dir</i>
find /dir/ -user name	Find files owned by <i>name</i> in <i>dir</i>
find /dir/ -mmin num	Find files modified less than <i>num</i> minutes ago in <i>dir</i>

Command	Description
<code>whereis command</code>	Find binary / source / manual for <i>command</i>
<code>locate file</code>	Find <i>file</i> (quick search of system index)

File Operations

Command	Description
<code>touch file1</code>	Create file1
<code>cat file1 file2</code>	Concatenate files and output
<code>less file1</code>	View and paginate file1
<code>file file1</code>	Get type of file1
<code>cp file1 file2</code>	Copy file1 to file2
<code>mv file1 file2</code>	Move file1 to file2
<code>rm file1</code>	Delete file1
<code>head file1</code>	Show first 10 lines of file1
<code>tail file1</code>	Show last 10 lines of file1
<code>tail -F file1</code>	Output last lines of file1 as it changes

Watch a Command

Command	Description
<code>watch -n 5 'ntpq -p'</code>	Issue the ‘ntpq -p’ command every 5 seconds and display output

Process Management

Command	Description
<code>ps</code>	Show snapshot of processes
<code>top</code>	Show real time processes
<code>kill pid</code>	Kill process with id <i>pid</i>
<code>pkill name</code>	Kill process with name <i>name</i>
<code>killall name</code>	Kill all processes with names beginning <i>name</i>

Nano Shortcuts

Shortcut	Description	Shortcut	Description
<code>Ctrl-R</code>	Read file	<code>ALT-A</code>	Start marking text
<code>Ctrl-O</code>	Save file	<code>CTRL-K</code>	Cut marked text or line
<code>Ctrl-X</code>	Close file	<code>CTRL-U</code>	Paste text
	Navigate File		Search File
<code>ALT-/</code>	End of file	<code>CTRL-W</code>	Find
<code>CTRL-A</code>	Beginning of line	<code>ALT-W</code>	Find next
<code>CTRL-E</code>	End of line	<code>CTRL-\</code>	Search and replace
<code>CTRL-C</code>	Show line number		
<code>CTRL-_</code>	Go to line number		
	Cut and Paste		
More nano info at: http://www.nano-editor.org/docs.php			

Screen Shortcuts

Shortcut	Description	Shortcut	Description
<code>screen</code>	Start a screen session.	<code>CTRL-A n</code>	Go to the next instance of terminal.

Shortcut	Description	Shortcut	Description
<code>screen -r</code>	Resume a screen session.	<code>CTRL-A p</code>	Go to the previous instance of terminal.
<code>screen -list</code>	Show your current screen sessions.	<code>CTRL-A "</code>	Show current instances of terminals.
<code>CTRL-A</code>	Activate commands for screen.	<code>CTRL-A A</code>	Rename the current instance.
<code>CTRL-A c</code>	Create a new instance of terminal.	More screen info at: http://www.gnu.org/software/screen/	

File Permissions

Command	Description
<code>chmod 775 file</code>	Change mode of <i>file</i> to 775
<code>chmod -R 600 folder</code>	Recursively chmod <i>folder</i> to 600
<code>chown user:group file</code>	Change <i>file</i> owner to <i>user</i> and group to <i>group</i>

File Permission Numbers

Number	Permission
4	read (r)
2	write (w)
1	execute (x)

First digit is owner permission, second is group and third is everyone. Calculate permission digits by adding numbers above.