



Mastering the Linux File System: Cheat Sheet

Basic Navigation and Listing Files

Command	Description
<code>pwd</code>	Print current working directory
<code>ls</code>	List directory contents
<code>ls -l</code>	Long listing format (permissions, size, owner, etc.)
<code>ls -a</code>	List all files, including hidden files (starting with <code>.</code>)
<code>ls -lh</code>	Long listing format with human-readable file sizes
<code>cd [directory]</code>	Change directory to <code>[directory]</code>
<code>cd ..</code>	Move to the parent directory
<code>cd ~</code>	Go to the home directory
<code>cd -</code>	Go to the previous directory

File and Directory Operations

Command	Description
<code>touch [filename]</code>	Create an empty file or update the timestamp of an existing file
<code>cat [filename]</code>	View the contents of a file
<code>cp [source] [destination]</code>	Copy file from source to destination
<code>cp -r [source] [destination]</code>	Copy a directory recursively
<code>mv [source] [destination]</code>	Move or rename a file or directory
<code>rm [filename]</code>	Remove a file
<code>rm -r [directory]</code>	Remove a directory recursively
<code>rm -f [filename]</code>	Force delete a file (without confirmation)
<code>mkdir [directory]</code>	Create a new directory
<code>rmdir [directory]</code>	Remove an empty directory

File Viewing and Editing

Command	Description
<code>cat [filename]</code>	Concatenate and display the contents of a file
<code>more [filename]</code>	View file contents page by page (forward navigation)
<code>less [filename]</code>	View file contents page by page (allows both forward and backward navigation)
<code>head [filename]</code>	Display the first 10 lines of a file
<code>head -n [number] [filename]</code>	Display the first <code>[number]</code> lines of a file
<code>tail [filename]</code>	Display the last 10 lines of a file
<code>tail -n [number] [filename]</code>	Display the last <code>[number]</code> lines of a file
<code>nano [filename]</code>	Open a file in the Nano text editor
<code>vim [filename]</code>	Open a file in the Vim text editor



File Permissions

Command	Description
<code>chmod [permissions] [file]</code>	Change file permissions (e.g., <code>chmod 755 filename</code>)
<code>chown [user]:[group] [file]</code>	Change file owner and group
<code>chgrp [group] [file]</code>	Change group ownership of a file

File Search

Command	Description
<code>find [directory] -name [name]</code>	Search for files by name in a directory
<code>locate [filename]</code>	Find the location of a file using an indexed database
<code>grep [pattern] [file]</code>	Search for a pattern in a file
<code>grep -r [pattern] [directory]</code>	Recursively search for a pattern in a directory

Text Searching with grep

Command	Description
<code>grep [pattern] [file]</code>	Search for a pattern in a file
<code>grep -i [pattern] [file]</code>	Case-insensitive search for a pattern
<code>grep -r [pattern] [directory]</code>	Recursively search for a pattern in a directory
<code>grep -v [pattern] [file]</code>	Invert match (display lines that do not match the pattern)
<code>grep -n [pattern] [file]</code>	Display matching lines with line numbers
<code>grep -o [pattern] [file]</code>	Show only the matching part of the lines
<code>grep -c [pattern] [file]</code>	Count the occurrences of a pattern

Sorting Data

Command	Description
<code>sort [file]</code>	Sort lines in a file alphabetically
<code>sort -r [file]</code>	Sort lines in reverse order
<code>sort -n [file]</code>	Sort lines numerically
<code>sort -k [column] [file]</code>	Sort by a specific column
<code>sort -t [delimiter] -k [column] [file]</code>	Sort using a custom delimiter and column (e.g., <code>-t", "</code> for CSV files)
<code>sort -u [file]</code>	Sort and remove duplicate lines



Text Processing with `awk`

Command	Description
<code>awk '{print \$1}' [file]</code>	Print the first column from a file
<code>awk '{print \$1, \$3}' [file]</code>	Print the first and third columns
<code>awk '/pattern/ {print \$0}' [file]</code>	Print lines that match a pattern
<code>awk '{sum += \$1} END {print sum}' [file]</code>	Calculate the sum of the first column
<code>awk 'BEGIN {FS=","} {print \$1, \$2}' [file]</code>	Set a custom field separator and print specific columns (for CSVs)
<code>awk '{if (\$3 > 50) print \$0}' [file]</code>	Conditional printing based on column values
<code>awk 'BEGIN {count=0} /pattern/ {count++} END {print count}' [file]</code>	Count occurrences of a pattern

Text Processing with `sed`

Command	Description
<code>sed 's/old/new/g' [file]</code>	Replace all instances of "old" with "new" in a file
<code>sed -n '/pattern/p' [file]</code>	Print lines that match a pattern
<code>sed -n '5,10p' [file]</code>	Print lines 5 to 10 from a file
<code>sed '/pattern/d' [file]</code>	Delete lines matching a pattern
<code>sed 's/^/prefix/g' [file]</code>	Add prefix to the beginning of each line

Text Processing with `cut`

Command	Description
<code>cut -d"," -f1,2 [file]</code>	Extract the first two columns from a CSV file
<code>cut -c 1-5 [file]</code>	Extract characters from column 1 to 5
<code>cut -d" " -f3 [file]</code>	Extract the third column from a space-separated file



File Compression and Archiving

Command	Description
<code>tar -cvf [archive.tar] [file/dir]</code>	Create an archive from files or directories (tar file)
<code>tar -xvf [archive.tar]</code>	Extract files from an archive
<code>tar -czvf [archive.tar.gz] [file/dir]</code>	Create a compressed archive (tar.gz file)
<code>tar -xzvf [archive.tar.gz]</code>	Extract files from a compressed archive
<code>gzip [filename]</code>	Compress a file
<code>gunzip [filename.gz]</code>	Decompress a .gz file
<code>zip [archive.zip] [file/dir]</code>	Compress files into a zip archive
<code>unzip [archive.zip]</code>	Extract a zip archive

Disk Usage and Space Management

Command	Description
<code>df -h</code>	Display disk space usage in human-readable format
<code>du -h [directory]</code>	Show disk usage of a directory in human-readable format
<code>du -sh [directory]</code>	Summarize disk usage of a directory

Links

Command	Description
<code>ln -s [target] [link]</code>	Create a symbolic (soft) link
<code>ln [target] [link]</code>	Create a hard link



Combination of Commands

Command	Description
<code>find . -name "*.txt" xargs grep "error"</code>	Find all .txt files and search for the word "error"
<code>find . -size +100M -exec ls -lh {} \;</code>	Find files larger than 100MB and display their names
<code>sort -k 2 file.txt grep "error"</code>	Sort a file by the second column and then search for lines containing "error"
<code>grep -o "error" file.txt wc -l</code>	Count how many times the word "error" appears in a file
<code>awk '{print \$2}' file.txt sort uniq</code>	Print the second column from a file, sort it, and remove duplicates
<code>ls -lt *.log grep "\$(date +%b\ %d)" sort</code>	List all .log files, sort by name, and show those modified today
<code>grep "error" file.txt awk '{print \$1, \$3}'</code>	Extract lines containing "error" and print only the first and third columns
<code>find . -name "*.log" -exec grep -n "error" {} \; awk -F: '{print \$1, \$2}'</code>	Find .log files, search for lines with "error", and print filenames and line numbers
<code>inotifywait -m . sort -k2</code>	Monitor file system changes in real-time and sort by time of modification

Here are some additional file-handling commands, specifically tailored for working with .pdb files and other scientific file formats in Linux:

Command	Description
<code>grep "ATOM" file.pdb</code>	Extract all lines containing the atom information from a .pdb file
<code>grep "HETATM" file.pdb</code>	Extract lines with heteroatoms (non-standard residues) from a .pdb file
<code>awk '/ATOM/ {print \$2, \$3, \$4}' file.pdb</code>	Print specific columns (atom serial number, atom name, residue name)
<code>sed -n '/^ATOM/,/^END/p' file.pdb</code>	Extract lines between ATOM and END keywords in a .pdb file
<code>sort -k 2 file.pdb</code>	Sort the .pdb file based on the second column (e.g., atom serial number)
<code>grep "CA" file.pdb wc -l</code>	Count the number of alpha carbon atoms (CA) in a .pdb file
<code>head -n 100 file.pdb > first_100_lines.pdb</code>	Extract the first 100 lines of a .pdb file and save to a new file
<code>tail -n 100 file.pdb > last_100_lines.pdb</code>	Extract the last 100 lines of a .pdb file and save to a new file
<code>cat file1.pdb file2.pdb > combined.pdb</code>	Concatenate two .pdb files into one



<pre>grep -A 10 "ATOM" file.pdb</pre>	Print 10 lines after each occurrence of the keyword <code>ATOM</code>
<pre>sed 's/HETATM/ATOM/g' file.pdb > modified.pdb</pre>	Replace all instances of <code>HETATM</code> with <code>ATOM</code> in a <code>.pdb</code> file
<pre>awk '{if (\$3 == "CA") print}' file.pdb</pre>	Extract only alpha carbon atoms (<code>CA</code>) from the <code>.pdb</code> file
<pre>cut -c 13-16 file.pdb sort uniq</pre>	Extract the residue names, sort them, and display unique residue types
<pre>awk '\$3 == "O" {print \$0}' file.pdb</pre>	Print lines where the third column represents oxygen atoms





vi (Vim) Editor Cheat Sheet

Basic Modes in vi

Mode	Description
Command Mode	Default mode for issuing commands. Press <code>ESC</code> to enter.
Insert Mode	Used for editing text. Enter by pressing <code>i</code> , <code>a</code> , <code>o</code> , etc.
Visual Mode	Used for selecting text. Enter by pressing <code>v</code> .
Ex Mode	Used for executing commands like saving and quitting. Enter by pressing <code>:</code> .

Basic vi Commands

Command	Description
<code>i</code>	Enter Insert Mode before the cursor
<code>I</code>	Enter Insert Mode at the beginning of the current line
<code>a</code>	Enter Insert Mode after the cursor
<code>A</code>	Enter Insert Mode at the end of the current line
<code>o</code>	Open a new line below and enter Insert Mode
<code>O</code>	Open a new line above and enter Insert Mode
<code>Esc</code>	Switch back to Command Mode
<code>v</code>	Enter Visual Mode to select text
<code>V</code>	Select entire lines in Visual Mode
<code>Ctrl + v</code>	Enter Visual Block Mode for selecting columns

Navigating Within a File

Command	Description
<code>h</code>	Move cursor left
<code>j</code>	Move cursor down
<code>k</code>	Move cursor up
<code>l</code>	Move cursor right
<code>w</code>	Move to the next word
<code>b</code>	Move to the beginning of the current/previous word
<code>e</code>	Move to the end of the current word
<code>0 (zero)</code>	Move to the beginning of the current line
<code>\$</code>	Move to the end of the current line
<code>gg</code>	Move to the beginning of the file
<code>G</code>	Move to the end of the file
<code>nG</code>	Go to line number <code>n</code> (e.g., <code>10G</code> to go to line 10)
<code>Ctrl + f</code>	Scroll forward (one screen)
<code>Ctrl + b</code>	Scroll backward (one screen)



Inserting Text

Command	Description
i	Insert before the cursor
I	Insert at the beginning of the line
a	Append after the cursor
A	Append at the end of the line
o	Open a new line below and enter Insert Mode
O	Open a new line above and enter Insert Mode

Deleting and Cutting Text

Command	Description
x	Delete the character under the cursor
dw	Delete from the cursor to the start of the next word
d\$	Delete from the cursor to the end of the line
d0	Delete from the cursor to the beginning of the line
dd	Delete (cut) the current line
n dd	Delete (cut) <i>n</i> number of lines (e.g., 5dd to delete 5 lines)
D	Delete the rest of the line after the cursor
dG	Delete from the current line to the end of the file
dgg	Delete from the current line to the beginning of the file

Copying and Pasting Text (Yank and Put)

Command	Description
yy	Yank (copy) the current line
n yy	Yank <i>n</i> lines (e.g., 3yy to yank 3 lines)
yw	Yank from the cursor to the end of the word
y\$	Yank from the cursor to the end of the line
y0	Yank from the cursor to the beginning of the line
P	Paste after the cursor or line
p	Paste before the cursor or line

Undo, Redo, and Repeat

Command	Description
u	Undo the last action
U	Undo all changes to the current line
Ctrl + r	Redo the undone action
.	Repeat the last command



Searching and Replacing Text

Command	Description
<code>/pattern</code>	Search forward for <code>pattern</code>
<code>?pattern</code>	Search backward for <code>pattern</code>
<code>n</code>	Move to the next occurrence of the search <code>pattern</code>
<code>N</code>	Move to the previous occurrence of the search <code>pattern</code>
<code>:%s/old/new/g</code>	Replace all occurrences of <code>old</code> with <code>new</code> in the file
<code>:%s/old/new/gc</code>	Replace all with confirmation (<code>c</code> prompts for each occurrence)
<code>:n,m s/old/new/g</code>	Replace between lines <code>n</code> and <code>m</code>

Saving and Exiting

Command	Description
<code>:w</code>	Save the file
<code>:w [filename]</code>	Save the file with the specified filename
<code>:q</code>	Quit
<code>:q!</code>	Quit without saving changes
<code>:wq OR ZZ</code>	Save and quit
<code>:x</code>	Save and quit (only if changes have been made)
<code>:w !sudo tee %</code>	Save as a superuser if permission is denied (after edit)

Working with Multiple Files

Command	Description
<code>:e [filename]</code>	Open a file for editing
<code>:w [filename]</code>	Save the file with the specified filename
<code>:n</code>	Go to the next file
<code>:prev</code>	Go to the previous file
<code>:bn</code>	Go to the next buffer
<code>:bp</code>	Go to the previous buffer
<code>:sp [filename]</code>	Open a file in a new horizontal split
<code>:vsp [filename]</code>	Open a file in a new vertical split
<code>Ctrl + w</code>	Switch between windows (splits)

Line Numbers

Command	Description
<code>:set nu</code>	Show line numbers
<code>:set nonu</code>	Hide line numbers
<code>:set relativenumber</code>	Show relative line numbers

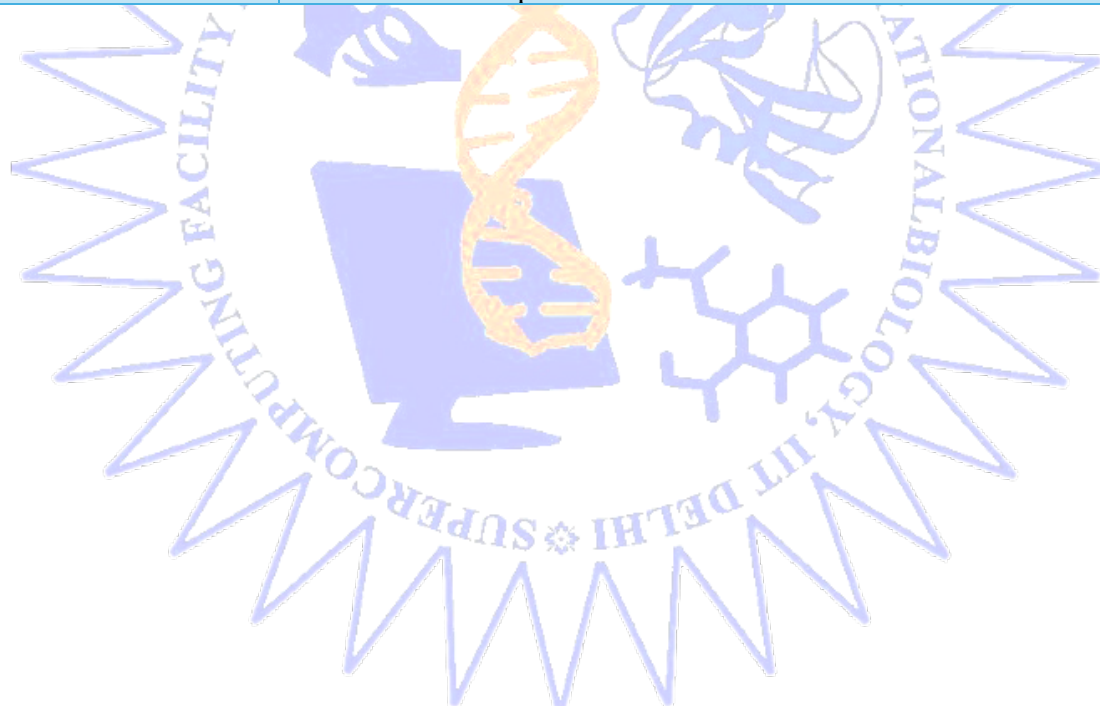


Indenting

Command	Description
>>	Indent the current line
<<	Unindent the current line
n>>	Indent n lines
n<<	Unindent n lines

Tabs and Windows

Command	Description
:tabnew [filename]	Open a file in a new tab
gt OR :tabnext	Go to the next tab
gT OR :tabprev	Go to the previous tab
:tabclose	Close the current tab
Ctrl + w s	Split window horizontally
Ctrl + w v	Split window vertically
Ctrl + w w	Switch between splits





Nano Editor Cheat Sheet

Basic Navigation

Command	Description
Ctrl + A	Move to the beginning of the current line
Ctrl + E	Move to the end of the current line
Ctrl + Y	Move up one screen (scroll up)
Ctrl + V	Move down one screen (scroll down)
Ctrl + _ + [line, column]	Jump to a specific line and column
Ctrl + G	Display help menu

Basic File Operations

Command	Description
Ctrl + O	Save the file (write out)
Ctrl + R	Insert another file into the current one
Ctrl + X	Exit the editor
Ctrl + W	Search within the file
Ctrl + \	Search and replace
Ctrl + K	Cut the current line (remove)
Ctrl + U	Paste (uncut) the last cut text
Ctrl + C	Display the cursor position (line, column)

Text Editing

Command	Description
Ctrl + K	Cut the current line or selected text
Ctrl + U	Paste the last cut text
Alt + 6	Copy the current line or selected text
Ctrl + J	Justify (format) the text
Ctrl + T	Spell check the text

Selecting and Manipulating Text

Command	Description
Ctrl + ^	Start marking/selecting text
Ctrl + K	Cut selected text
Alt + 6	Copy selected text
Ctrl + U	Paste the cut/copied text



Search and Replace

Command	Description
Ctrl + W	Search for a string or pattern
Ctrl + W, Ctrl + R	Search and replace a string
Alt + W	Find the next occurrence
Ctrl + T	Invoke spell check

Indenting and Unindenting

Command	Description
Ctrl + I	Indent selected text or the current line
Alt + I	Unindent selected text or the current line

File Management

Command	Description
Ctrl + O	Save the current file
Ctrl + X	Exit nano
Ctrl + R	Insert content from another file
Ctrl + S	Save the file (suspend write mode if file is read-only)

Navigating Text

Command	Description
Ctrl + A	Move to the beginning of the current line
Ctrl + E	Move to the end of the current line
Ctrl + Y	Move up one page (previous screen)
Ctrl + V	Move down one page (next screen)
Ctrl + _	Jump to a specific line and column
Ctrl + C	Show the cursor position (line and column)

Working with Multiple Buffers

Command	Description
Ctrl + R	Insert another file into the current buffer
Alt + F	Close the current buffer



Toggles

Command	Description
Alt + Z	Toggle wrapping of long lines
Alt + X	Toggle syntax highlighting
Alt + N	Toggle line numbers display
Alt + I	Toggle auto-indentation

Exiting

Command	Description
Ctrl + X	Exit nano (prompts to save if there are unsaved changes)
Ctrl + O	Save the current file before exiting
Ctrl + X, n	Exit without saving changes

