

TEXT EDITORS

TEXT EDITORS

- Most common: nano and vi
- nano
 - More “user friendly”
 - Easy to use
- vi
 - By default, installed on most distributions
 - Advanced
 - Powerful
 - Released in 1976



NANO

```
GNU nano 5.6.1 /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/sbin/nologin
dbus:x:81:81:system message bus:/sbin/nologin
tss:x:59:59:Account used for TPM access:/dev/null:/sbin/nologin
sssd:x:998:995:User for sssd:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.sshd:/sbin/nologin
chrony:x:997:994:chrony system user:/var/lib/chrony:/sbin/nologin
systemd-oom:x:992:992:systemd Userspace OOM Killer:/usr/sbin/nologin
ern:x:1000:1000:eResearch Admin:/home/ern:/bin/bash
ern_admin:x:1001:1001:~/home/ern_admin:/bin/bash
```

Help Exit Write Out Read File Where Is Replace Cut Paste Execute Justify Location Go To Line Undo Redo Set Mark Copy To Bracket Where Was Prev/Next Back Forward Prev/Next Word Home End Prev/Next Line Scroll Up/Down Prev/Next Block Begin/End of Paragraph Prev/Next Page First/Last Line

```
GNU nano 5.6.1 /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/sbin/nologin
dbus:x:81:81:system message bus:/sbin/nologin
tss:x:59:59:Account used for TPM access:/dev/null:/sbin/nologin
sssd:x:998:995:User for sssd:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.sshd:/sbin/nologin
chrony:x:997:994:chrony system user:/var/lib/chrony:/sbin/nologin
systemd-oom:x:992:992:systemd Userspace OOM Killer:/usr/sbin/nologin
ern:x:1000:1000:eResearch Admin:/home/ern:/bin/bash
ern_admin:x:1001:1001:~/home/ern_admin:/bin/bash
```

Help Exit Write Out Read File Where Is Replace Cut Paste Execute Justify Location Go To Line Undo Redo Set Mark Copy To Bracket Where Was Prev/Next Back Forward Prev/Next Word Home End Prev/Next Line Scroll Up/Down Prev/Next Block Begin/End of Paragraph Prev/Next Page First/Last Line

38 Options

12 Options



NANO FILE MANAGEMENT

Task	Keystroke
Open a file from within nano	Ctrl+r NOTE: tab completion is in effect; also, once this command has been entered, notice the new menu items at the bottom of the screen. For example, Ctrl+T will allow you to browse the file system and look for a file to open.
Display the next file buffer	Alt+>
Display the previous file buffer	Alt+<
Save the current file buffer to disk	Ctrl+o
Close the current file buffer	Ctrl+x NOTE: If the file hasn't been saved yet, you'll be asked if you want to save it. Also, if there's only one file buffer open, closing it will exit from nano.

NANO COPY AND PASTE

Task	Keystroke
Select a region for a cut or paste operation	Alt+a NOTE: After setting a mark with Alt+a, move the cursor to define the region, you should see it highlighted as you move the cursor. Also, to cancel the definition of the region just enter Alt+a again.
Copy a highlighted region into the clipboard	Alt+^
Cut a highlighted region into the clipboard	Ctrl+k
Paste the contents of the clipboard at the current cursor position	Ctrl+u
Cut from the current cursor position to the end-of-line (EOL)	Ctrl+k NOTE: This command doesn't require highlighting of the region.

NANO NAVIGATION

Task	Keystroke
Go to beginning of file	Alt+\
Go to end of file	Alt+/ NOTE: Very useful for finding mismatched brace compiler errors!
Move forward one screenful	Ctrl+v
Move backward one screenful	Ctrl+y
Go to a target line number	Alt+g
Jump to matching open/close symbol	Alt+] NOTE: Very useful for finding mismatched brace compiler errors!
Window scrolling	Alt+= to scroll down, Alt+- to scroll up
Indenting/Outdenting selected blocks	Use Alt+a to select a block, then Alt+} will indent the selected block, and Alt+{ will outdent the block.

vi MODES

Name	Description	Help page
normal	For navigation and manipulation of text. This is the mode that vim will usually start in, which you can usually get back to with ESC.	:help Normal-mode
insert	For inserting new text. The main difference from vi is that many important "normal" commands are also available in insert mode - provided you have a keyboard with enough meta keys (such as Ctrl, Alt, Windows-key, etc.).	:help Insert-mode
visual	For navigation and manipulation of text selections, this mode allows you to perform most normal commands, and a few extra commands, on selected text.	:help Visual-mode
select	Similar to visual, but with a more MS Windows-like behaviour.	:help Select-mode
command-line	For entering editor commands - like the help commands in the 3rd column.	:help Command-line-mode
Ex-mode	Similar to the command-line mode but optimized for batch processing.	:help Ex-mode

Visual Line mode: Shift+V

Visual Block mode: Ctrl+V

https://en.wikibooks.org/wiki/Learning_the_vi_Editor/Vim/Modes

Vi & Vim

CHEAT SHEET 101

Press **ESC** (escape key) to get into the Vi command mode. You will need to press the **RETURN** key to execute a command starting with the character colon ":", slash "/", or question mark "?". Use **CTRL+c** to cancel a command, the dot "." to repeat your last command, and **!:cmd d** to execute a shell command named cmd.

Insert & Replace		insert mode
i	insert text before the cursor	
a	insert text after the cursor	
I	insert text at the beginning of the current line	
A	insert text at the end of the current line	
o	insert text in a new line below the cursor	
O	insert text in a new line above the cursor	
r	replace a character at the cursor position	
R	replace characters starting at the cursor position	

Cursor Motion		navigation
h or ←	insert text before the cursor	
l or →	insert text after the cursor	
k or ↑	insert text at the beginning of the current line	
j or ↓	insert text at the end of the current line	
G	goto the end of file	
nG or :n	goto the line number n	
0	move to the beginning of the line	
\$	move to the end of the line	
CTRL+f	move one screen view forward	
CTRL+b	move one screen view backward	

Ex Commands		vi features
:q	quit current open file	
:q!	force to quit without saving changes	
:w	save file	
:w file	save file as file	
:wq!	force to save file then quit	
:,x,yw file	write from line x to line y into file	
:w » file	append buffer to file	
:e file	edit another file	
:e! file	edit another file without saving current changes	
:r file	insert file content at the current cursor position	
:n	edit next file in vi arguments file list	
CTRL+G	display current file name and position	

You can start vi with arguments, the -c option is particularly useful to execute commands in a file from the command line. You can separate multiple commands with a pipe "|".

Example:
vi -c "%s/false/true/g|:wq" file.txt

Going Further with:
:map for mapping a key in command mode to a group of commands.
 Example: **:"map de :l,\$dAM"** will delete all lines when using the :de command.
:set to define or show your editor options.
 Example: **:"set number"** will make vi display line numbers, and **:"set all"** will display all current vi options.
:ab to define a text abbreviation in insert mode.
 Example: **:"ab VIM Vi Improved"** will auto-complete "VIM" in insert mode for the phrase "Vi Improved".

Search & Substitute		patterns
/string	search forward for string	
?string	search backward for string	
n	repeat the last search	
:/s/str1/str2/gc	search and substitute str1 by str2 from the current line to the end of file. A line number can be specified before the comma ",". The "c" ask for confirmation before each substitution	
:/s/str1/str2/g	replace all str1 by str2 in all the file without prompting for confirmation	
:/s/str1/str2/	replace str1 by str2 on the first occurrence in each line of the file	

Undo, Delete, and Copy		edit commands
u	undo last change	
CTRL+r	redo last undo change	
mA	set mark A to current position. Marks can be the letter [a-z] and [A-Z]	
y'A	yank (copy) from current line to mark A	
d'A	delete from current line to mark A	
P	put the buffer content before the cursor	
p	put the buffer content after the cursor	
x	delete character at cursor position	
dW	delete first word after cursor position	
d\$ or D	delete from cursor position to end of line	
dd	delete current line	
J	join current line with following line	

<https://www.shell-tips.com>

shell-tips!

Essentials References for the **vi/vim text editor**
Last Updated: Mar 20, 2022

https://www.shell-tips.com/cheat-sheets/vim-quick-references/vi_vim_cheat_sheet.pdf

VIM EXAMPLES

- Download the example `get_data.sh` script
- Make the script executable and execute it
- Open the file `mobi.txt` with `vi`
 - Search for the second line starting with `***`
 - Delete 352 lines
 - Go to the first line, delete the first 844 lines
 - On which line does Chapter 42 start?
 - What is the first word of the 4th sentence in that chapter?
 - Go to the bottom of the file
 - How many lines are in the file
 - Exit `vi`, without saving the changes



VIM EXAMPLES

- unzip the convertcsv.txt.zip file
- Open convertcsv.txt with vi
- In vi:
 - Search for John
 - Search for John, excluding other names, e.g., Johnathan
- Delete the whole first column
- What is the name of the person on line 13 735?
- Undo all changes, without quitting and without saving
- Go to line 1
 - Add a semi-colon ; as a separator between columns
(for all records)
- Save the file to a new name: **converted.csv**
- Exit vi.... Is it asking to save the file again?



MODIFY SUDOERS



- Become root
- Open either `/etc/sudoers` or execute `visudo`
 - Search for a line starting with `%wheel`
 - Comment out that line (add `#` in the front of the line)
 - Search for a line containing `NOPASSWD`
 - Uncomment that line (remove the `#` from the front)

 - Save and exit vi
 - Log out of the system
 - Log back in
 - Execute a sudo command....was the password requested