**UNIX FUNDAMENTALS**

Additional Student Materials - Crib Sheets

[Useful UNIX Commands](#_Toc178312704)

[vi Crib Sheets](#_Toc178312705)

[Useful Web References](#_Toc178312706)

Useful UNIX Commands

## Setup

printenv   Display environment variable settings

printenv *var*   Display a particular environment variable – e.g. ‘EDITOR’

setenv EDITOR xxx   Change the default editor

passwd   Change password

## On-Line Help

whatis cmd   A one-line description of command ‘cmd’

man cmd   Display the manual page for the command

man -k topic   Display all commands dealing with ‘topic’

## Alias Commands

alias   Display all defined aliases

alias xxx   Display command associated with alias xxx

alias xxx ‘cmd’   Assign a command to alias xxx. To make it permanent, insert into .cshrc file

## Access permissions on files

set-access   Set default for future files \*

make-public name   Make file or directory read-access to others \*

make-private name   Make file or directory inaccessible to others \*

umask 077   Set permissions on all new files and directories to ‘no-read / no-write’

umask 022   Set permissions on all new files and directories to ‘read / no-write’

chmod 600 file   Make ‘file’ inaccessible to others

chmod 644 file   Make ‘file’ read-accessible to others

chmod 700 dir   Make ‘dir’ inaccessible to others

chmod 755 dir   Make ‘dir’ read-accessible to others

## Display Directory on the Screen

ls   Display directory listing

ls -a   Include invisible files (or ‘la’)

ls -l   Include file sizes in bytes, and access permissions. (or ‘ll’)

ls dir   List files in another directory

ls file   List a single filename. Use wildcard characters (? and \*) to list multiple files

ls -Fal dir   Example of a complex command

## Print Directory Contents

ls | lpr   Print current directory on default printer

ls dir | lpr   Print directory dir

## Display File Contents on Screen

cat file   Display ‘file’ contents on the screen

head file   Display the first 10 lines of a file

head -n file   Display the first n lines of a file

tail file Display the last 10 lines of a file.

tail -n file   Display the last n lines of a file

more file   Display a file one screen at a time

? to display keystrokes available  
‘b’ to go back one screen  
SPC to display the next screenful  
/string to search for a string  
RET to display the next line  
‘n’ to move to next occurrence of string  
9 and RET to display next 9 lines  
‘q’ to quit from the ’more’ command

## Important Characters

| “pipe” directs the output of the first command to the input of another.

> Sends the output of a command to a designated file

>> Appends the output of a command to a designated file

& Runs command in the background; you can still work in the window

~ Designates the home directory ($HOME)

< Designates input from somewhere other than terminal

## Wildcards

\* Any string of characters

? Any one character

[ ] Match any character in the brackets

## Print File Contents

lpr file   Print the contents of ’file’

lpq   Display the queue of printer jobs

lprm job#   Remove a printer job

lprm -   Remove all your printer jobs

lpr -Pprinter file   Print file on the named ‘printer’

lpconfig **printer\_id** **queue** Configures remote printers to a local print queue

lpconfig -r **queue** Removes the said queue from the local system

lpconfig -d **queue** Makes the said queue the default queue

lpstat (-*options*) Prints printer status information to screen (*options not required*)

lpstat -u“**user1, user2**” Prints the status of requests made by the specified users

lpstat s Prints the queues and the printers they print to

lpstat -t Shows all print status information

lpstat -d Shows the default printer for the lp command

lpstat -r Lets you know if the line printer scheduler is running

lp (-*option*) **file**(**s**) Like pr, this prints designated files on the connected printer(s) (*options* *not* *required and options may be combined*).

lp -d*dest* **file**(**s**) Prints the file(s) to a specific destination

lp -n*number* **file**(**s**) Allows user to designate the number of copies to be printed

lp -t*title* **file**(**s**) Places *title* on the banner page

lp -o*option* **file**(**s**) Allows printer-specific options to be used (i.e., double-sided or two pages per side, etc.)

cancel **request\_id** Stops print jobs or removes them from the queue (**request\_ids** are obtained using lpstat)

cancel -a **printer** Removes all print requests from the current user on the specified printer

cancel -u **login\_id** Removes any print requests queued belonging to the user

## Redirect Output

script file  Send screen output from now on, to ‘file’. Type ‘exit’ to stop sending

cmd > file   Redirect output to file (no overwrite.)

cmd >> file   Redirect output to file (append).

cmd >! file   Redirect output to file (overwrite).

cmd | cmd   Pipe output of first command to second command

cmd | tee file   Pipe output to file and screen

## Change / Create Directory

pwd   Display the name of the current directory

cd   Change to your home directory

cd ..   Change to parent directory (one level up)

cd dir   Change to directory ‘dir’

mkdir dir   Make directory (create a new directory)

## Remove Files and Directories

rm file   Remove (delete) file

rmdir dir   Remove an empty directory

rm -r dir   Remove a directory and all its contents

## Move / Rename / Copy

mv source-file destination

* If destination is an existing directory, the source file will be moved into that directory
* If destination does not exist, the source file will be renamed
* If destination is an existing file, the source file will over-write the destination file

mv source-directory destination

* If destination is an existing directory, move the source directory into it.
* If destination does not exist, rename the directory.

cp src dst

* Copy works the same as mv, but keeps original file or directory.

## Search for File or Strings within Files

search ‘str’   Search in this and lower directories for all filenames containing str \*

find . –name ‘str’ -print   Search in directories for filename ‘str’ – may use wildcards in str

grep ‘str’ file   Search within file/s for ‘str’ – may use wildcards in filename

## Get Info about File / Spell Check

wc file   Display how many lines, words and chars in a file

wc -w file   Display a word count of file

wc -l file   Display a line count of file

ispell file   Check spelling (Once in, press ? for help)

## Global Changes within a File

tr old new < a > b   Search and replace

sed <a > b s/old/new   Search and replace

expand a > b   Expand tab characters to spaces

colrm n m < a > b   Remove columns n thru m

<   Input from a file > Will not over-write an existing file

>   Output to a file >>   Will append to an existing file

>!   Will over-write an existing file

## Concatenate Files

cat a b > c   Concatenate files

paste a b > c   Paste files together horizontally

## Sort / Split / Convert Files

sort a > c   Sort lines contained in file a, output to c

split   Split one file into two or more files (See man page for syntax)

uniq   Remove or report adjacent duplicate lines

unix2dos   Convert from Unix to DOS format

dos2unix   Convert from DOS to Unix format

## Compare Files

diff a b   Display line-by-line differences

sdiff a b   Display two files side by side

## Disk Space Information

du   (Disk usage) Display number of kilobyte blocks per file in current directory

du dir   Display number of kilobyte blocks per file in dir

du -s \*   Disk usage summary

## User Information

whoami   Display your own username

id   Display your own username, ID, and group memberships

groups   Display your own group memberships

groups user   Display group memberships of user

who   Who is logged on

w   Who is logged on and what are they doing

finger   Display info about all logged-on users

finger user   Display user information (‘user’ is a person’s login, first name, or last name)

## Talk to Another User

talk user   Send a message (immediately) to another logged-on user. CTRL-c to exit

mesg n   Disallow ‘talk’ requests (Use ‘mesg y’ to allow)

mesg   Check whether ‘talk’ requests are allowed

## Date / Time / Calendar

date   Display the date and time

cal yyyy   Display a yearly calendar

cal   Display this month’s calendar

vacation   Reply to mail automatically while you are away

## Basic Calculator

bc   Get into calculator interactive mode

1+1   Type arithmetic expressions, using + - \* /

scale=n   Change the number of decimal places

quit   Exit from the calculator.

## Miscellaneous Commands

!!   Repeat the last command

!xxx   Repeat the last command that started with xxx

!n   Repeat command n, (n is command line number)

;   Command separator

clear   Clear the screen

history 9   Display the last 9 commands. The number is optional

## Batch Unix Commands and Background Jobs

csh file   Execute the Unix commands contained in a file

cmd&   Run command in background. (Only run one; do not logout until finished)

CTRL-z   Suspend current job. (fg to resume in foreground; bg to resume in background)

jobs   List background jobs (and job numbers).

ps   Display status of foreground and background processes (and process id numbers)

fg n   Bring job number n to foreground. (Omit the ‘n’ for most recent job)

kill PID   Terminate job (PID is process id number)

## Special Control Keys

CTRL-c   Interrupts most programs

CTRL-s   Stop screen scroll

CTRL-q   Restart screen scroll

CTRL-z   Suspend current job (type fg to resume)

vi Crib Sheets

Before doing anything to a document, type the following command followed by a carriage return:

**:set showmode**

**GOOD PRACTICE NOTE ESPECIALLY FOR BEGINNERS: WHEN USING VI, HIT [ESC] TWICE BEFORE EVERY NEW COMMAND. THIS MAKES SURE YOU AREN’T IN THE WRONG MODE.**

When you open VI, you can’t just start typing a thesis because VI has different modes such as APPEND MODE, INSERT MODE, REPLACE MODE, and COMMAND MODE. So to do certain actions, you must first enter the appropriate mode. See “MODES” for more in-depth information.

The final page of this document is a list of VI core commands. These are the bare essentials of VI compressed onto one page.

**STARTING VI** (VI is **CaSe SEnsItiVe**!!! So make sure Caps Lock is **OFF**.)

|  |  |  |
| --- | --- | --- |
| **Command** | **Result** | |
| vi *filename* | | **Edits** *filename*. |
| vi -r *filename* | | **Edits last saved** versionof*filename* **after a crash**. |
| vi + **n** *filename* | | **Edits** *filename* and **places** the **cursor** at **line n**. |
| vi + *filename* | | **Edits** *filename* and **places cursor on** the **last line**. |
| vi *filename* *file2 ...* | | **Edits** *filename* andthen **edits *file2*** andsoon. After saving *file1* enter **:n** for the next file. |
| vi +/**string** *file* | | **Edits** *file*and **places cursor at** the **first line containing string**. |

**ENDING VI** (Saving, exiting, etc.)

|  |  |  |
| --- | --- | --- |
| **Command** | **Result** | |
| ZZ or :wq or :x | | **Saves** and **exits VI.** |
| :w | | **Saves** current **file** but **does not exit**. |
| :w *file* | | **Saves** current **as *file*** but **does not exit**. |
| :w! *file* | | **Saves** *file* **overriding normal checking**. |
| :**n**,**m**w *file* | | **Saves lines n through m to** afilenamed ***file***. |
| :**n**,**m**w>>*file* | | **Appends lines n through m to** the **end of** afilenamed ***file***. |
| :q | | **Quits** VI**, saving changes** beforeleaving. (You may be prompted to save.) |
| :q! | | **Quits** VI **without saving changes.** |
| Q | | **Escapes** VI **into ex editor** withthesamefile**; :vi returns**. |
| :e! | | **Re-edits** currentfile **disregarding changes** sincelastsave. |
| :we! | | **Re-edits** currentfile **saving changes**. |

**STATUS** (Line numbers, etc.)

|  |  |
| --- | --- |
| **Command** | **Result** |
| :.= | **Shows** current **line number**. |
| := | **Shows number of lines** infile. |
| CTRL-g | **Shows** **filename,** current **line number, total lines** infile**,** and **%** offile **location**. |
| :l (letter “l”) | **Displays** **tab** (^l) **backslash** (\) **backspace** (^H) **newline** ($) **bell** (^G) **formfeed** (^L^) of current line. |

**MOVING** (These will tell you how to get the cursor where you want it–fast.)

|  |  |  |  |
| --- | --- | --- | --- |
| **MOVING THE CURSOR** | | | |
| **Command** (**ESC** exits all modes except the initial Command Mode.) | | **Result** | |
| arrow keys | These *do* work, but they may be too slow on big files. | | |
| h j k l | **Left, down, up, and right,** respectively. | | |
| CTRL-d | Moves **forward ½ screenful**. | | |
| CTRL-f | Moves **forward one entire screenful**. | | |
| CTRL-u | Moves **backward ½ screenful**. | | |
| CTRL-b | Moves **backward one entire screenful**. | | |
| **n**G | Moves to **line n** **in the file**. | | |
| G | Moves to the **end of file**. | | |
| H | Moves to the **top** of the screen. | | |
| **n**H | Moves to **line n from** the **top** of the screen. | | |
| M | Moves to the **middle** of the **screen**. | | |
| L | Moves to the **bottom** of the **screen**. | | |
| **n**L | Moves to **line n from** the **bottom** of the screen. | | |
| w or W | Moves to the **start** ofthe **next word** in the line; W ignores punctuation. | | |
| b or B | Moves to the **start** ofthe **previous word** in the line; B ignores punctuation. | | |
| e or E | Moves to the **end** ofthe **next word** in the line; E ignores punctuation. | | |
| 0 (zero) or | | Moves to the **first column** in the current line. | | |
| **n**| | Moves to the **column n** in the current line. | | |
| ^ | Moves to the **first non-blank character** in the current line. | | |
| $ | Moves to the **last character** in the current line. | | |
| + or <CR> | Moves to the **first character** in the **next line**. | | |
| - | Moves to the **first non-blank character** in the **previous line**. | | |
| ( | Moves **back** to the **beginning of sentence**. | | |
| ) | Moves **forward** to the **beginning** of the **next sentence**. | | |
| { | Moves **back** to the **beginning** of the **paragraph**. | | |
| } | Moves **forward** to the **beginning** of the **next paragraph**. | | |
| **MOVING THE SCREEN** | | | |
| **Command** (**ESC** exits all modes except the initial Command Mode.) | | | **Result** |
| CTRL–e | Moves **screen up one line**. | | |
| CTRL–y | Moves **screen down one line**. | | |
| CTRL–u | Moves **screen up ½ page**. | | |
| CTRL–d | Moves **screen down ½ page**. | | |
| CTRL–b | Moves **screen up one page**. | | |
| CTRL–f | Moves **screen down one page**. | | |
| CTRL–I | **Redraws screen**. | | |
| z <CR> | z–carriage return makes the **current line the top line** on the page. | | |
| **n**z <CR> | Makes the **line n** the **top line** on the page. | | |
| z. | Makes the **current line** the **middle line** on the page. | | |
| **n**z. | Makes the **line n** the **middle line** on the page. | | |
| z– | Makes the **current line** the **bottom line** on the page. | | |
| **n**z– | Makes the **line n** the **bottom line** on the page. | | |

**MODES** (Adding and/or replacing text)

|  |  |  |
| --- | --- | --- |
| **Command** (**ESC** exits all modes except the initial Command Mode.) | | **Result** |
| i*text* ESC | **insert mode**—You can start typing and it will insert *text* before the letter your cursor currently highlights until ESC is used. | |
| I*text* ESC | **insert mode—**Same as (i) except it will insert *text* before the first nonblank character on the line until ESC is used. | |
| a*text* ESC | **append mode—**Allows you to insert *text* after the letter your cursor currently highlights until ESC is used. | |
| A*text* ESC | **append mode—**Same as (a) except it will insert *text* at the end of the line until ESC is used. | |
| r*char* ESC | **replace mode—**Replaces the currently highlighted character with *char* until ESC is used. | |
| R*text* ESC | **replace mode—**Same as (r) except it will overwrite until ESC is used. | |
| o | **Opens** new **line *below*** thecurrent line and **insert** until ESC is used. | |
| O (letter “O”) | **Opens** new **line *above*** the current line and **insert** until ESC is used. | |
| CTRL–v *char* | While **inserting, ignores special meaning of *char*** (e.g., for inserting characters like ESC and CTRL) until ESC is used. | |
| :r *file* | **Reads** *file*and **inserts** it **after current line**. | |
| :**n**r *file* | **Reads** *file*and **inserts** it **after line number n**. | |

**SEARCHING** (Find and/or replace what you need)

|  |  |  |  |
| --- | --- | --- | --- |
| **Characters** | **What they match...** | | |
| ^ (caret) | | Matches beginning of line. | |
| $ | | Matches end of line. | |
| . | | Matches any single character. | |
| \< | | Matches beginning of word. | |
| \> | | Matches end of word. | |
| [*str*] | | Matches any single character in *str*. | |
| [^*str*] | | Matches any character not in *str*. | |
| [*a*-*n*] | | Matches any character between *a* and *n*. | |
| \* | | Matches zero or more occurrences of previous character in expression. | |
| \ | | Escapes the meaning of the next character (e.g., \$ allows you to search for $). | |
| \\ | | Escapes the \ character. | |
| **Command**  (**ESC** exits all modes except the initial Command Mode.) | | | **Result** |
| % | | **Searches to beginning of balancing ( ) [ ] or { }**. | |
| f*char* | | **Searches forward** incurrent lineto ***char***. | |
| F*char* | | **Searches backward** in current line to ***char.*** | |
| t*char* | | **Searches forward** in current line to **character before *char***. | |
| T*char* | | **Searches backward** in current line to **character before *char***. | |
| /*str* <CR> | | **Finds forward** to ***str***. | |
| ?*str* <CR> | | **Finds** in **reverse** for ***str***. | |
| :set ic | | **Ignores case** when **searching**. | |
| :set noic | | **Pays attention to case** when **searching**. | |
| :**n**,**m**s/*str1*/*str2*/*opt* | | **Searches** from **n** to **m** for *str1*; **replaces** *str1* to *str2*; using *opt*–*opt* can be **g** for global change, **c** to confirm change (y to acknowledge, <CR> to suppress), and **p** to print changed lines. | |
| & | | **Repeats** last **:s command**. | |
| :g/*str*/cmd | | **Runs cmd** on **all** lines that **contain** ***str***. | |
| :g/*str1*/s/*str2*/*str3*/ | | **Finds** the line containing *str1*, **replaces** *str2* with *str3*. | |
| :v/*str*/cmd | | **Executes** cmd on all lines that **do** **not match** *str*. | |

**EDITING** (Deleting, copying, placing text, joining lines, changing text, repeat command, undo command)

| **Command** (**ESC** exits all modes except the initial Command Mode.) | | **Result** |
| --- | --- | --- |
| CTRL–h or  Backspace | **While inserting,** deletes **previous character**. | |
| CTRL–w | **While inserting,** deletes **previous word** | |
| CTRL–x | **While inserting,** deletes **to start of inserted text**. | |
| **n**x | **Deletes n characters starting with current**; omitting **n** deletes current character only. | |
| **n**X | **Deletes previous n characters**; omitting **n** deletes previous character only. | |
| xp | **Switches character at cursor with following character**. | |
| **n**dw | **Deletes** the **next n words starting with current**; omitting **n** deletes the current word only. | |
| **n**db | **Deletes** the **previous n words starting with current**; omitting **n** deletes the previous word only. | |
| **n**dd | **Deletes n lines beginning with the current line**; omitting **n** deletes the current line only. | |
| :**n**,**m**d | **Deletes lines n through m**. | |
| D or d$ | **Deletes from** the **cursor to** the **end** of the line. | |
| d**cursor\_cmd** | **Deletes** **everything included in** the **cursor command** (e.g., dG would delete from current position to the end of the file, and d4 would delete to the end of the fourth sentence). | |
| **n**yy or **n**Y | **Places** **n** **lines** **in** the **buffer**–**copies**; omitting **n** copies only the current line to the buffer. | |
| y**cursor\_cmd** | **Copies from cursor to cursor\_cmd** (e.g., yG copies current line to the last line in the file to the buffer). | |
| “(a-z)**n**yy or  “(a-z)**n**dd | **Copies or cuts (deletes) n lines into** a **named buffer a through z**; omitting **n** works on current line. | |
| p | **Pastes copied text after cursor**; also prints last deleted text. | |
| P | **Pastes copied text before the cursor**; also prints last deleted text. | |
| “(a-z)p or  “(a-z)P | **Pastes text from a named buffer a through z after** or **before** the current line. | |
| **n**J | **Joins** the **next n lines together**; omitting **n** joins the beginning of the next line to the end of the current line. | |
| s*text* ESC | **Substitutes *text* for** the **current character** until ESC is used. | |
| S or cc *text* ESC | **Substitutes *text* for** the **entire line** until ESC is used. | |
| cw*text* ESC | **Changes** current **word to *text*** until ESC is used. | |
| C*text* ESC | **Changes rest of** the current **line to *text*** until ESC is used. | |
| c**cursr\_cmd***text* ESC | **Changes to *text* from current position to cursr\_cmd** until ESC is used. | |
| u | **Undoes last** command. | |
| U | **Restores current line** to its original state. | |
| “**n**p | **Retrieves the last nth delete** (last 9 deletes are kept in a buffer). | |
| “1pu.u. | **Scrolls through the delete buffer** until the desired delete is retrieved **(repeat u.).** | |
| n | **Repeats last / or ? search command**. | |
| N | **Repeats,** in **reverse** order**, last / or ? search command**. | |
| ; (semi-colon) | **Repeats last f F t or T search command**. | |
| , (comma) | **Repeats,** in **reverse** direction**, last / or ? search command**. | |
| . (period) | **Repeats last** text **change command**. | |
| CTRL–i or TAB | While inserting, **inserts one shift width**. | |
| **n**<< or **n**>> | **Shifts *n* lines** left or right (respectively) by **one shift width**; omitting **n** shifts one line. | |
| < or > | Use with cursor command to **shift multiple lines** left or right. | |

**SHELL WORKS** (Commands that execute outside of the VI program.)

|  |  |  |
| --- | --- | --- |
| **Command** (**ESC** exits all modes except the initial Command Mode) | | **Result** |
| :! cmd | **Executes** shell command **cmd**; you can add these special characters to indicate: % name of current file # name of last file edited | |
| !! cmd | **Executes** shell command **cmd**, **places output** in file starting at current line. | |
| :!! | **Executes last** shell command. | |
| :r! cmd | **Reads** and **inserts output** from **cmd**. | |
| :f *file* | **Renames** current file **to *file***. | |
| :w !cmd | **Sends** currently edited file **to cmd** as standard input and **execute cmd**. | |
| :cd dir | **Changes** current working **directory** **to** **dir**. | |
| :sh | **Starts** a **sub-shell** (CTRL-d returns to editor). | |
| :so *file* | **Reads** and **executes** commands in *file* (*file* is a shell script). | |
| !cursor\_cmd cmd | **Sends text** from current position to cursor\_cmd **to shell command cmd**. | |
| !}sort <CR> | **Sorts** from current position to end of paragraph and replaces text with sorted text. | |

**MACROS, SUBS, ABBREVIATIONS** (Macro functions, substitution strings, and abbreviations)

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** (**ESC** exits all modes except the initial Command Mode.) | | | **Result** |
| :[**address**]s/*search-string*/***replace-string***/[/g] | | This is the format of a substitute command––description below. | |
| **address** | One line number or two line numbers separated by a comma. A. represents the current line, $ represents the last line, and % represents the entire file. | | |
| *search-string* | A regular expression that can be a simple string of characters. | | |
| ***replace-string*** | The replacement string. | | |
| g | Indicates a global replacement (more than one replacement per line). | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Setting Options** | | | | | | |
| **Command** | **Result** | | **Command** | | **Result** | |
| NOTE: Map allows you to define strings of VI commands. If you create a file called “.exrc” in your home directory, any map or set command you place inside this file will be executed every time you run VI. To imbed control characters like ESC in the macro, you need to precede them with CTRL–v. If you need to include quotes (“), precede them with a \ (backslash). Unused keys in vi are: K V g q v \* = and the function keys.  EX: :map v /I CTRL–v ESC dwiYou CTRL–v ESC ESC  when v is pressed, search for “I” (/I ESC), delete word (dw), and insert “You” (iYou ESC). CTRL–v allows ESC to be inserted. | | | | | | |
| :map *key* **cmd\_seq** | | Defines *key* to run **cmd\_seq** when pressed. | | :map | | Displays all created macros on status line. |
| :unmap *key* | | Removes macro definition for key. | | :ab *str* **string** | | When *str* is input, replaces it with **string**. |
| :ab | | Displays all abbreviations. | | :una *str* | | Unabbreviates *str*. |
| Options given are default. To change them, enter type :set *option* to turn them on or :set no*option* to turn them off.  To make them execute every time you open VI, create a file in your HOME directory called ***.exrc*** and type the options without the colon (:) preceding the option. | | | | | | |
| **Command** | **Result** | | **Command** | | **Result** | |
| :set all | Prints all options to the screen. | | :set no*option* | | Turns off *option*. | |
| :set *ai* | Turns on auto indentation. | | :set *ap* | | Prints line after *d c J m :s t u* commands. | |
| :set *bf* | Discards control characters from input. | | :set *eb* | | Precedes error messages with a bell. | |
| :set *ic* | Ignores case when searching. | | :set *dir=tmp* | | Sets directory or buffer file. | |
| :set *lisp* | Modifies brackets for Lisp compatibility. | | :set *magic* | | Allows pattern matching with special characters. | |
| :set *mesg* | Allows others to send messages. | | :set *list* | | Shows tabs (^l) and end of line ($). | |
| :set *nu* | Shows line numbers. | | :set *opt* | | Speeds output; eliminates automatic RETURN. | |
| :set *prompt* | Prompts for command input with :. | | :set *re* | | Simulates smart terminal on dumb terminal. | |
| :set *report* | Indicates largest size of changes reported on status line. | | :set *ro* | | Changes file type to “read only.” | |
| :set *scroll=****n*** | set ***n*** lines for CTRL–d and z | | :set *sh=****shell\_path*** | | set shell escape (default is /bin/sh) | |
| :set *showmode* | Indicates input or replace mode at bottom. | | :set *sw=****n*** | | Sets shift width to **n** characters. | |
| :set *term* | Prints terminal type. | | :set *terse* | | Shorten messages with terse. | |
| :set *timeout* | Eliminates one-second time limit for macros. | | :set *tl=****n*** | | Sets significance of tags beyond **n** characters (0 means all). | |
| :set *ts=****n*** | Sets tab stops to **n** for text input. | | :set *wa* | | Inhibits normal checks before write commands. | |
| :set *warn* | Warns “no write since last change.” | | :set *window=****n*** | | Sets number of lines in a text window to **n**. | |
| :set *wm=****n*** | Sets automatic wraparound **n** spaces from right margin. | |  | |  | |

VI Core Commands (The minimum functions)

|  |  |
| --- | --- |
| Command | Result |

STARTING VI (VI is CaSe SEnsItiVe!!! So make sure Caps Lock is OFF.)

|  |  |
| --- | --- |
| vi *file* | edit *file* |

ENDING VI (Saving, exiting, etc.)

|  |  |
| --- | --- |
| ZZ or :wq or :x | Saves and exits VI. |
| :q | Quits VI, saving changes before leaving. (You may be prompted to save.) |
| :q! | Quits VI without saving changes. |

MOVING (These will tell you how to get the cursor where you want it–fast.)

|  |  |
| --- | --- |
| arrow keys | These *do* work, but they may be too slow on big files. |
| h j k l | Left, down, up, and right, respectively. |
| CTRL-d | Moves forward ½ screenful. |
| CTRL-f | Moves forward 1 whole screenful. |
| CTRL-u | Moves backward ½ screenful. |
| CTRL-b | Moves backward 1 entire screenful. |
| nG | Moves to line n in the file. |
| G | Moves to the end of file. |
| H | Moves to the top of the screen. |
| M | Moves to the middle of the screen. |
| L | Moves to the bottom of the screen. |
| w or W | Moves to the start of the next word in the line; W ignores punctuation. |
| b or B | Moves to the start of the previous word in the line; B ignores punctuation. |
| e or E | Moves to the end of the next word in the line; E ignores punctuation. |
| ^ | Moves to the first non-blank character in the current line. |
| $ | Moves to the last character in the current line. |
| + or <CR> | Moves to the first character in the next line. |

MODES (Adding and/or replacing text)

|  |  |
| --- | --- |
| i*text* ESC | insert mode—You can start typing, and it will insert *text* before the letter your cursor currently highlights until ESC is used. |
| a*text* ESC | append mode—This allows you to insert *text* after the letter your cursor currently highlights until ESC is used. |
| A*text* ESC | append mode—Same as (a) except it will insert *text* at the end of the line until ESC is used. |
| r*char* ESC | replace mode—Replaces the currently one highlighted character with *char* until ESC is used. |
| R*text* ESC | replace mode—Same as (r) except it will overwrite until ESC is used. |
| o | Opens new line *below* the current line and insert until ESC is used. |
| O (letter “O”) | Opens new line *above* the current line and insert until ESC is used. |

SEARCHING (Find and/or replace what you need)

|  |  |
| --- | --- |
| f*char* | F*char* | Searches forward in current line to *char.* | Searches backward in current line to *char.* |
| /*str* <CR> | ?*str* <CR> | Finds forward to *str*. | Finds in reverse for *str*. |

EDITING (Deleting, copying, placing text, joining lines, changing text, repeat command, undo command)

|  |  |
| --- | --- |
| CTRL–h or Backspace | While inserting, deletes previous character. |
| CTRL–w | While inserting, deletes previous word |
| nx | Deletes n characters starting with current; omitting n deletes current character only. |
| ndw | Deletes the next n words starting with current; omitting n deletes the current word only. |
| ndd | Deletes n lines beginning with the current line; omitting n deletes the current line only. |
| D or d$ | Deletes from the cursor to the end of the line. |
| nyy or nY | Places n lines in the buffer–copies; omitting n copies only the current line to the buffer. |
| p | Pastes copied text after cursor; also prints last deleted text. |
| nJ | Joins the next n lines together; omitting n joins the beginning of the next line to the end of the current line. |
| u | Undoes last command. |
| U | Restores current line to its original state. |
| n | N | Repeats last / or ? search command. | Repeats, in reverse order, last / or ? search command*.* |
| . (period) | Repeats last text change command*.* |
| CTRL–i or TAB | While inserting, inserts one shift width. |

OPTIONS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command | Result | Command | | Result |
| :set all | Prints all options to the screen. | :set no*option* | Turns off *option*. | |
| :set *ic* | Ignores case when searching. | :set *magic* | Allows pattern matching with special characters. | |
| :set *nu* | Shows line numbers. | :set *ro* | Changes file type to “read only.” | |
| :set *scroll=n* | Sets *n* lines for CTRL–d and z. | :set *sw=n* | Sets shift width to n characters. | |
| :set *showmode* | Indicates input or replace mode at bottom. | :set *window=n* | Sets number of lines in a text window to n. | |
| :set *ts=n* | Sets tab stops to n for text input. | :set *wm=n* | Sets automatic wraparound n spaces from right margin. | |

# Useful Web References

## General UNIX Reference:

Unix Guide:

<http://www.unixguide.net/cgi-bin/unixguide.cgi>

Rosetta Stone For UNIX

<http://bhami.com/rosetta.html>

Basic UNIX/Linux Tutorial:

<http://www.ee.surrey.ac.uk/Teaching/Unix/>

SysAdmin Pocket Guide to Surviving UNIX

<http://www.cs.fiu.edu/~tho01/psg/psg.html>

## IBM AIX:

General AIX website:

<http://www.rootvg.net/>

IBM AIX Infocenter

<http://publib16.boulder.ibm.com/pseries/en_US/infocenter/base/resources.htm>

## SUN Solaris:

Official Sun Solaris Administrator Site:

<http://www.sun.com/bigadmin/home/index.html>

General Sun Solaris:

<http://www.adminschoice.com/>

## HP-UX:

HP-UX General Reference:

<http://docs.hp.com/en/B3921-90010/index.html>

HP-UX FAQ’s

<http://www.unixguide.net/hp/faq/>

AIX – HP-UX Interoperability

<http://www.freelab.net/unix/hp-ux/index.htm>