Computing Resources

Andrew Alverson and John Tipton

Created: 8/12/202
Last modified: 8/31/2020

Helpful keyboard shortcuts everyone should know

These are basic keyboard commands that everyone should know. Most people know these but for those that don't, they can be life changing.

On a **PC**, the `ctrl` is the control key. It is typically in both the lower-left and lower-right of the keyboard. The symbol `ctrl-c` represents pressing the `c` key while holding down the `ctrl` key. On a **Mac**, you execute these commands with the `command` key. Macs have both `ctrl` and `command` keys.

- `ctrl-c`
  - "Copy" – Copy the highlighted text and leave the text in place
- `ctrl-x`
  - "Cut" – Copy the highlighted text and remove the highlighted text
- `ctrl-v`
  - "Paste" – Paste the copied text
- `ctrl-z`
  - "Undo" – Undo the last action performed
- `ctrl-a`
  - "Select all" – Select everything in the current document
- `ctrl-s`
  - "Save" – Save the current document.
  - **This shortcut might be the most important in learning to code!**
- `ctrl-n`
  - "New" – Open an new document or page
- `ctrl-p`
  - "Print" – Print the current document
- `alt-tab`
  - "Switch applications" – Toggle between open applications

Terminal (command line)

https://swcarpentry.github.io/shell-novice/

Fundamental commands

- `ls`
  - 'list' – list the contents in the directory
- `pwd`
  - 'present working directory' – tells you where the current working directory is in the filepath
- `cd`

  - ‘change directory’ – move the working directory a given location in the filepath
- `mv`
  - ‘move file/folder’ – move a file/folder to a given location. This command removes the file/folder from the original location
- `cp`
  - ‘copy file/folder’ – move a file/folder to a given location. This command keeps a copy of the file/folder in the original location
- `cat`
  - ‘concatenate’ – view the contents of a file or multiple files
- `touch`
  - if the file doesn’t exists, create a new file with that name
- `mkdir`
  - ‘make directory’ – create a directory at a given location
- `rm`
  - ’remove‘ – delete the file or directory. Be very carefule with this command as there is no recycle bin/undo
- `clear`
  - clears the terminal output
  - Also: `ctrl-l`
- ’history‘
  - print out your command history
- `grep`
  - search for a string
  - example: `grep Ahab moby_dick.txt`, search the text file, ‘moby_dick.txt’ for all lines that contain the string, ‘Ahab’
- `more`
  - simple file viewing; exit `more` by pressing ‘q’
  - example: `more moby_dick.txt`
- `less`
  - more advanced file viewing; exit `less` by pressing ‘q’
  - example: `more moby_dick.txt`
  - `man less` will show the many options available for this program
- `|`
  - or ‘pipe’, channel the output from one command (i.e., `STDOUT`) into the input of a subsequent command (i.e., `STDIN`)
  - example: `grep Ahab moby_dick.txt | wc -l`, search for all lines containing the string, ‘Ahab’, then pipe those lines into a command (`wc -l`) to count them
- `help`
- `man`
  - ‘manual’ – show the manual for this command; exit `less` by pressing `q`

- example: `man grep`

- `whoami`
  - show the username of the user currently logged in
- `exit`
  - exit the Terminal session
- `ssh`
  - establish a **s**ecure **sh**ell, or more simply, log onto a remote server
  - format: `ssh username@domain`
  - example: `ssh ahab@rstats.uark.edu`
- `sftp`
- `echo`
  - print arguments to the screen
- `>`
  - 'redirect' – send the output (`STDOUT`) of this command to a file, either creating the file if it doesn't exist or over-writing the previous file
  - example: `grep Ahab moby_dick.txt > ahabs.txt`, write the results of this `grep` command to a file called 'ahabs.txt'
- `>>`
  - 'append' – like redirect, except that it will either create the file if it doesn't exist or *append* to the file if it does exist
  - example: `grep Ahab ahabs_bride.txt >> ahabs.txt`, append the results of this command to the file 'ahabs.txt', or create and write out to the file if it doesn't exist already
- `chmod`
  - 'change mode' – change the permssions of a file for **u**ser, **g**roup, or **o**ther by either granting (`+`) or removing (`-`) **r**ead, **w**rite, or e**x**ecute permission
  - example: `chmod u+rwx ahabs.txt`, give read, write, and execute permission to the user for the user
- `diff`
  - 'difference' – show differences, line by line, between two text files
- `wc`
  - '**w**ord **c**ount' – show statistics (e.g., character, word, and line counts) about a text file
- `*`
  - a wildcard character that, in the Unix environment, matches any character any number of times (including 0); the `*` behaves differently in regular expressions
  - example: `grep Ahab *.txt >> ahabs.txt`, look for lines containing the string 'Ahab' in all the files ending in '.txt' in the working directory
- `env | grep PATH`
  - Also: `echo $PATH`
- `top` – monitor computer resources (CPU and MEM)

RStudio keyboard short[cuts

https://rstudio.com/wp-content/uploads/2016/01/rstudio-IDE-cheatsheet.pdf

https://support.rstudio.com/hc/en-us/articles/200711853-Keyboard-Shortcuts

- `ctrl-enter`
  - 'Run current line of code' – run the current line of code that the cursor is on or if multiple lines are selected, run the selected lines of code
- `ctrl-k`
  - 'Knit' – if an Rmarkdown (.Rmd) file is open, knit the document to the format specified in the YAML header (the first few lines at the top of the document
- `ctrl-shift-M`
  - 'insert pipe' – Insert the `%>%` symbol into your `.R` and `.Rmd` document
- `ctrl-shift-c`
  - 'comment/uncomment' – comment/uncomment the current line of code that the cursor is on or if multiple lines are selected, comment/uncomment the selected lines of code

Text Editors

Basic Text editors

- Many different text editors are available in Unix

Advanced editor shortcuts (Emacs and Vim)

There are two text editors that are commonly used among more experienced (and probably older in age) programmers.

- Beginners guide to emacs