A friendly guide to Luke's i3 rice

Use vim keys (h/j/k/l) to navigate this document. Pressing W will fit it to window width. + and - zoom in and out. f to toggle fullscreen. q to quit. (These are general mupdf shortcuts.)

- Mod+F1 will show this document at any time.
- By "Mod" I mean the Super Key, usually known as "the Windows Key."

Questions or suggestions? Email me at luke@lukesmith.xyz.

Basic goals and principles of my rice:

- Naturalness – I want the number of keypresses I have to make to get what I want to be as little as possible.
- Economy – All the basic programs I use should be simple and light on system resources. Because of this, many are terminal or ncurses programsj
- Keyboard/vim-centrality – All my terminal apps (and other programs) use vim keys when possible. My hands never need leave the home row or thereabout.
- Lots of color – Many rices stick to one general color palatte. I like my system to be very vibrant. If you disagree, you can easily change it.

General changes

- Capslock is now an alternative escape. Makes vim-craft much more efficient.
- The menu button (usually between the right Alt and Ctrl) is an alternative Super/Mod button. This is to make one-handing on my laptops easier.
- The rice also uses the US International keyboard by default. This allows you to type a lot of characters in many different European languages.
- The Trackpad is turned off by default in my system. I only use either the Trackpoint or an external mouse.

The Polybar Status Bar

If you're new to i3, notice the status bar on the top of the screen. This is Polybar. To the left side, you'll see the numbers of your current workspace(s). If you have a song playing in mpd, its name will appear to the left as well. On the right side, you'll see various system status notifiers, date, CPU tempurature, remaining hard drive space, etc. I'm sure you can figure it out. Several modules will be click-sensitive, although if you're using my system as indended, you probably won't be doing much clicking.

Shortcut keys

Window basics

Notice the case sensitivity of the shortcuts.

Be sure you play around with these. Be flexible with the basic commands and the rice will grow on you quick.

- Mod+Enter – Spawn terminal
- Mod+q or Q – Close window
- Mod+d – rofi (For running commands or programs without shortcuts)
- Mod+t – Toggle between spawning vertically and horizontally
- Mod+f or F11 – Fullscreen
- Mod+h/j/k/l – Move to different windows
- Mod+H/J/K/L – Move a window around
- Mod+Y/U/I/O – Resize windows
- Mod+/ – Spawn vertical terminal
- Mod+' – Spawn horizonal terminal
- Mod+s/S – Increase/decrease inner gaps
- Mod+z/Z – Increase/decrease outer gaps
- Mod+D – Reduce gaps to 0 pixels
- Mod+T – Restore gaps to default (15 pixels)
- Mod+Shift+Space – Make a window float (you can still resize and move floating windows with the same keys above)
- Mod+Space – Switch from a floating window to a non-floating one (or vice versa)

Basic Programs

- Mod+r – ranger (file browser/manager)
- Mod+e – mutt (email)
- Mod+m – ncmpcpp (music player)
- Mod+A – alsamixer (volume control)
- Mod+a – R calculator (close with Mod+a for reusability)
- Mod+i – htop (system info)
- Mod+N – newsbeuter (RSS feed reader)
- Mod+y – calcurse (calendar and schedule)
- Mod+u – "Dropdown" terminal (close with Mod+u for reusability)
- Mod+Shift+Enter – tmux

Larger programs

- Mod+W – Firefox

- Mod+B – Blender
- Mod+G – GIMP
- Mod+P – MyPaint

System

- Mod+R – Restart/refresh i3 (renews configs, does not close any programs)
- Mod+x – i3lock (Enter password to return)
- Mod+X – shutdown now (Be careful with this one!)*
- Mod+Shift+Backspace – reboot (And this one!)
- Mod+Shift+Escape – exit i3 (And this one as well!)
- Mod+F1 – Shows this document
- Mod+F2 – Recreate dynamic config files (see below)
- Mod+F3 – arandr (For adding screens/HDMI/VGA connections)
- Mod+F4 – Hibernate
- Mod+F5 – Reset Network Manager*
- Mod+F7 – Increase window transparency
- Mod+F8 – Decrease window transparency
- Mod+F10 – Switch to laptop screen
- Mod+F11 – Switch to VGA display (if available)
- Mod+F12 – Switch to dual VGA/laptop display (if available)

*Note: I am on Void Linux now, which uses runit instead of systemd. As a result, users of most Linux distros may have to adjust the bindings for these commands.

Audio

I use ncmpcpp as a music player, which is a front end for mpd. If you prefer cmus or mocp, I have commented out shortcuts you can activate for it instead in the i3 config.

- Mod+m – ncmpcpp music player
- Mod+n – Next track
- Mod+b – Previous track
- Mod+o – Restart track
- Mod+p – Pause
- Mod+M – Mute all audio
- Mod+v – visualizer
- Mod+- – Decrease volume (holding shift increases amount)
- Mod++ – Increase volume (holding shift increases amount)
- Mod+[– Back 10 seconds (holding shift increases amount)
- Mod+] – Forward 10 seconds (holding shift increases amount)
- Mod+A – alsamixer (volume control)

Workspaces

There are ten workspaces. They work just like those in vanilla i3 with some additions.

- Mod+(Number) – Go to that number workspace
- Mod+Shift+(Number) – Send window to that workspace
- Mod+Tab – Go to previous workspace
- Mod+g or escape – Go to left workspace
- Mod+; – Go to right workspace
- Mod+Shift+Delete – "Porno Emergency!" Press this key sequence if you want to hide what you have on your screen. Moves to a totally new workspace, mutes sound, pauses music and brings up distraction windows.

Recording

I use scrot and ffmpeg to make different recordings of the desktop and audio. All of these recording shortcuts will output into `~`, and will not overwrite previous recordings.

- Print Screen – Take a scrot screenshot
- Shift+Print Screen – Take a scrot screenshot of only selected window
- Mod+Insert – Begin screencast.
- Mod+ScrollLock – Begin audio recording.
- Mod+Pause – Begin screen recording without audio.
- Mod+Print Screen – Start screenkey
- Mod+Delete – kills ffmpeg, thus ending recordings and screen
- ThinkVantage button (on Thinkpads) – kills ffmpeg, thus ending recordings

Each of the recording scripts are located in `~/.config/Scripts/`. You can check them out or modify them if needed.

Other buttons

I've mapped those extra buttons that some keyboards have (play and pause buttons, email, webbrowsing buttons, etc.) to what you would expect.

Special traits of my rice

Easy config access

Open a terminal and type `cfc`. This will open a file where you will see customizable pairs of key shortcuts and config files. Enter any of these shortcuts in bash or ranger to immediately open the file in vim.

You may add new entries here and they will be refreshed when you refresh i3 (Mod+R) or simply press Mod+F2,

Folder and config shortcuts

Open a terminal and type `cff`. This opens a file when you can keep and create folder shortcuts. There are only a few here now, because I don't know what your folder structure is going to look like, but on my machine, I have 81 and growing.

Each line has a shortcut key/keys and its target. These can be used in serveral applications. In bash, simply press `d`, the shortcut for `~/Documents` and you will cd there (and automatically `ls -a`).

ranger works similarly. When in ranger, just press `g` then the shortcut of the folder you want to go to. You may also press `t` plus the shortcut to open a new tab there. `m` plus the shortcut moves the selected files to the folder and `Y` copies them there. **Get good at this. It will make management of even the most complex file system easy.**

Lastly qutebrowser implements these shortcuts as well. When you see a file or image you want to download, press `;` followed by the folder shortcut and qutebrowser will let you select the file with its hint system. The file will then download to the directory you chose.

Dynamically constructed configs

To keep these different shortcuts in sync, my rice will dynamically reconstruct the configs to bash, qutebrowser and ranger each time you refresh i3 (Mod+R) or run the refresh configs script (Mod+F2).

Each time i3 starts or restarts, it will run `~/.config/Scripts/shortcuts.py`, which reads the entries in the folder shortcut and config shortcut files and then translate them into the approriate syntax of all three programs.

It then takes that output and appends it to base configs of each program (`~/.config/Scripts/bashrc`, `~/config/qutebrowser/keys.conf.base`, `~/.config/rc.conf.base`) and puts the output in the proper places for each program.

So what do I need to know?

Use the files in `cff` and `cfc` to add/change shortcuts. These shortcuts will be synced between bash, ranger and qutebrowser when you press Mod+F2 to refresh them. If you want to make permanent changes to your bash/ranger/qutebrowser configs, make them to the base files which you can access with `cfb`, `cfr`, and `cfq`, respectively, then press Mod+F2 to make those changes active.

Explore and customize

Don't like something about the rice? Change it. If you have a problem, try figuring it out yourself, but if you can't, ask on my YouTube or by my email.

EDIT: July 31, 2017. All the feedback I've gotten is tremendous and I'm glad this has gotten so many people into customization!

Contact

luke@lukesmith.xyz

http://lukesmith.xyz

Send Me Money!

My Github Page

Twitter