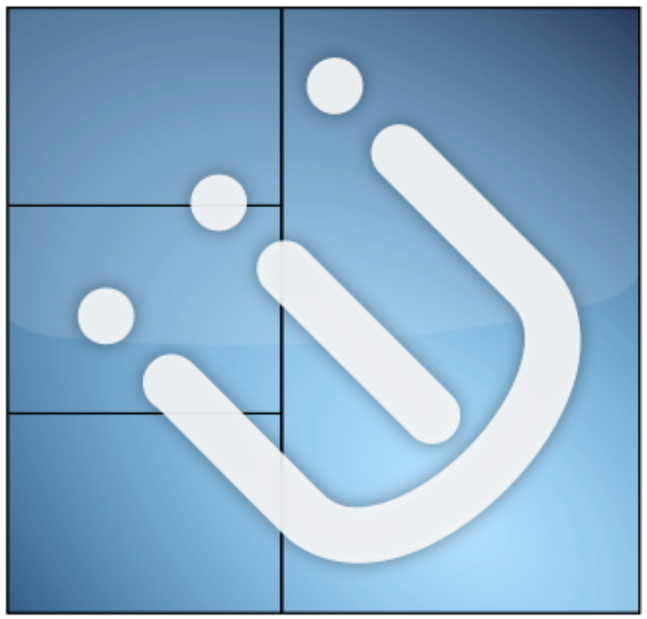


i3: the tree branch

an introduction to what‘s coming

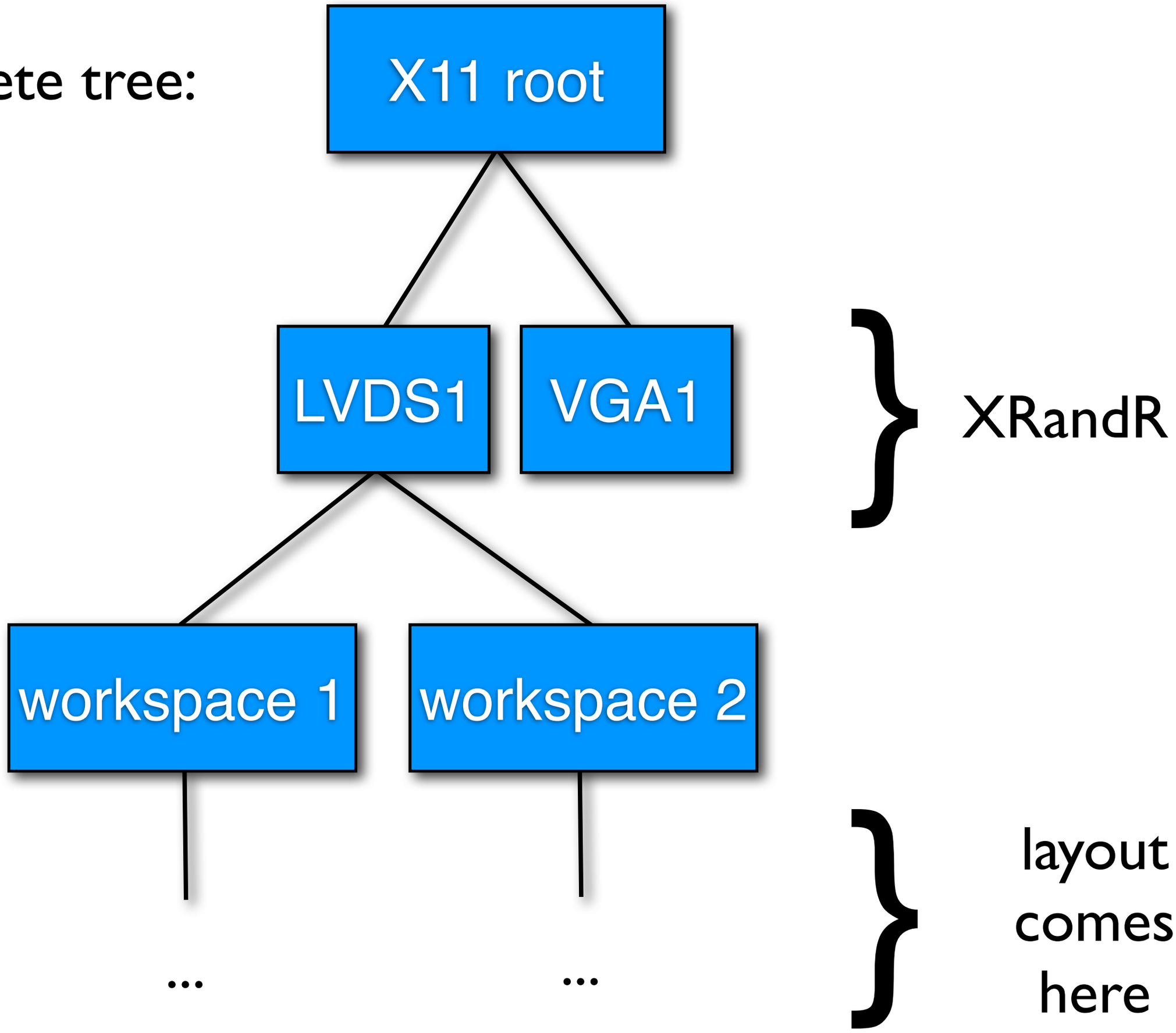
Disclaimer

- You need to know the current version of i3, this is not a presentation for beginners.
- The colors inside the screenshots look strange because it is a debugging version.

What is the tree branch?

- major refactoring of i3 (still in progress)
- easier data structures (good for devs and users)
- layout is tree-based now
- tree exposed via IPC (in JSON format):
  much more test-driven development

a complete tree:

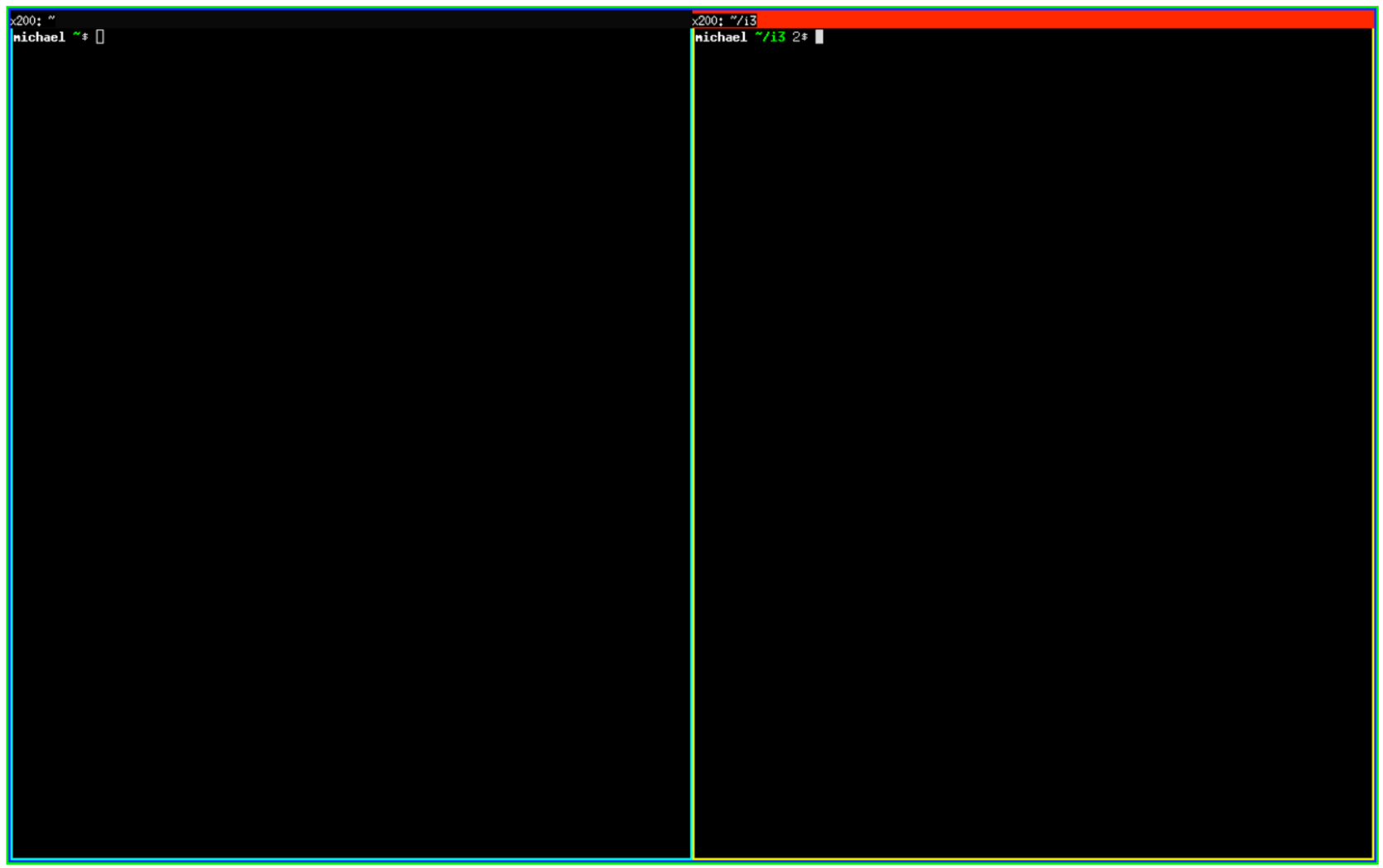


2: basic window management

NEW: The table metaphor is gone.
(It was too complicated and had too many ugly side-effects)

So, how do you move windows around?

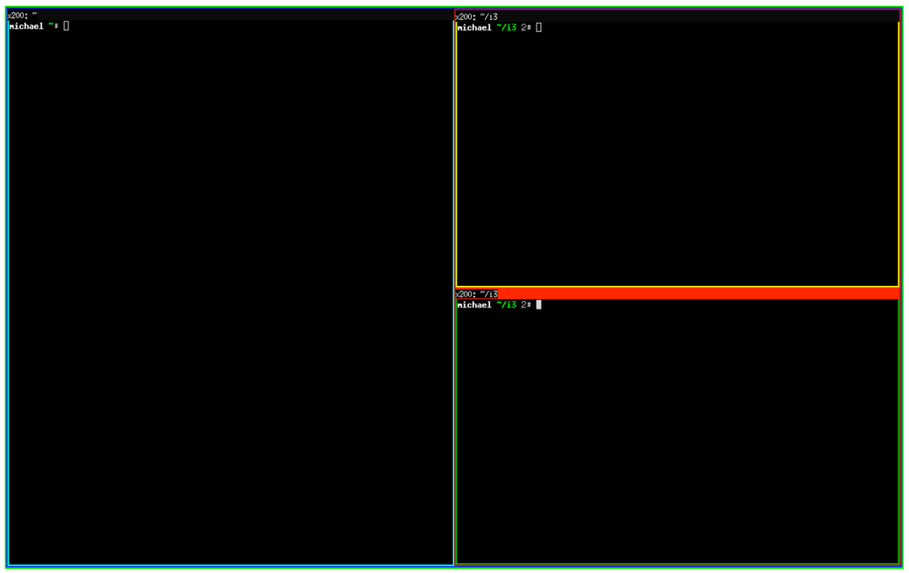
Let‘s open two terminals (alt+enter):

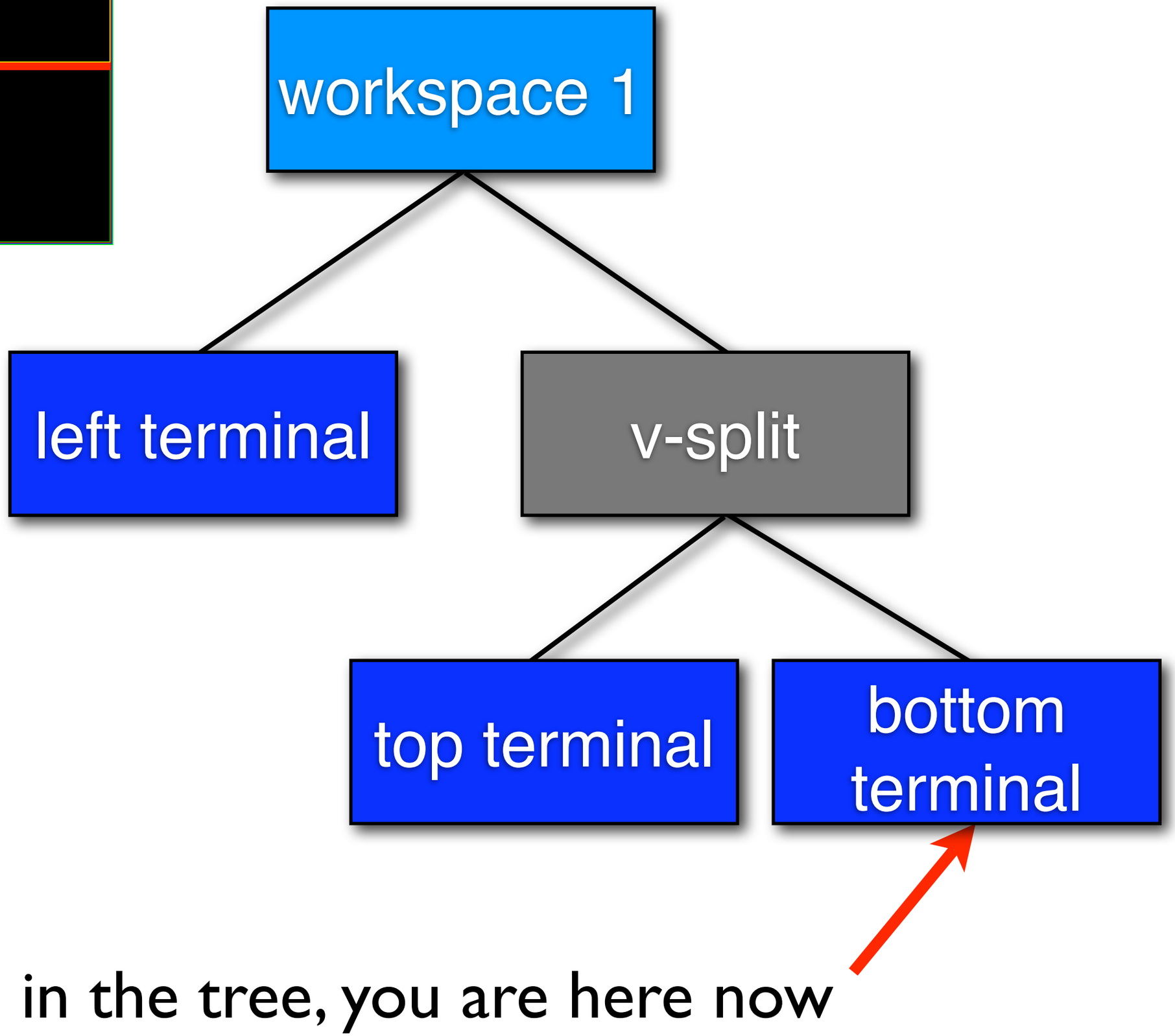


default mode is horizontal tiling
(most displays are wide-screen nowadays)

NEW: Every window you see is in a container.
This means you can split everything.

Let‘s split the current container vertically (alt+v)
and open a new terminal (alt+enter):


%200: ~
michael ~# []
%200: ~/13
michael ~/13 2# []
%200: ~/13
michael ~/13 2#

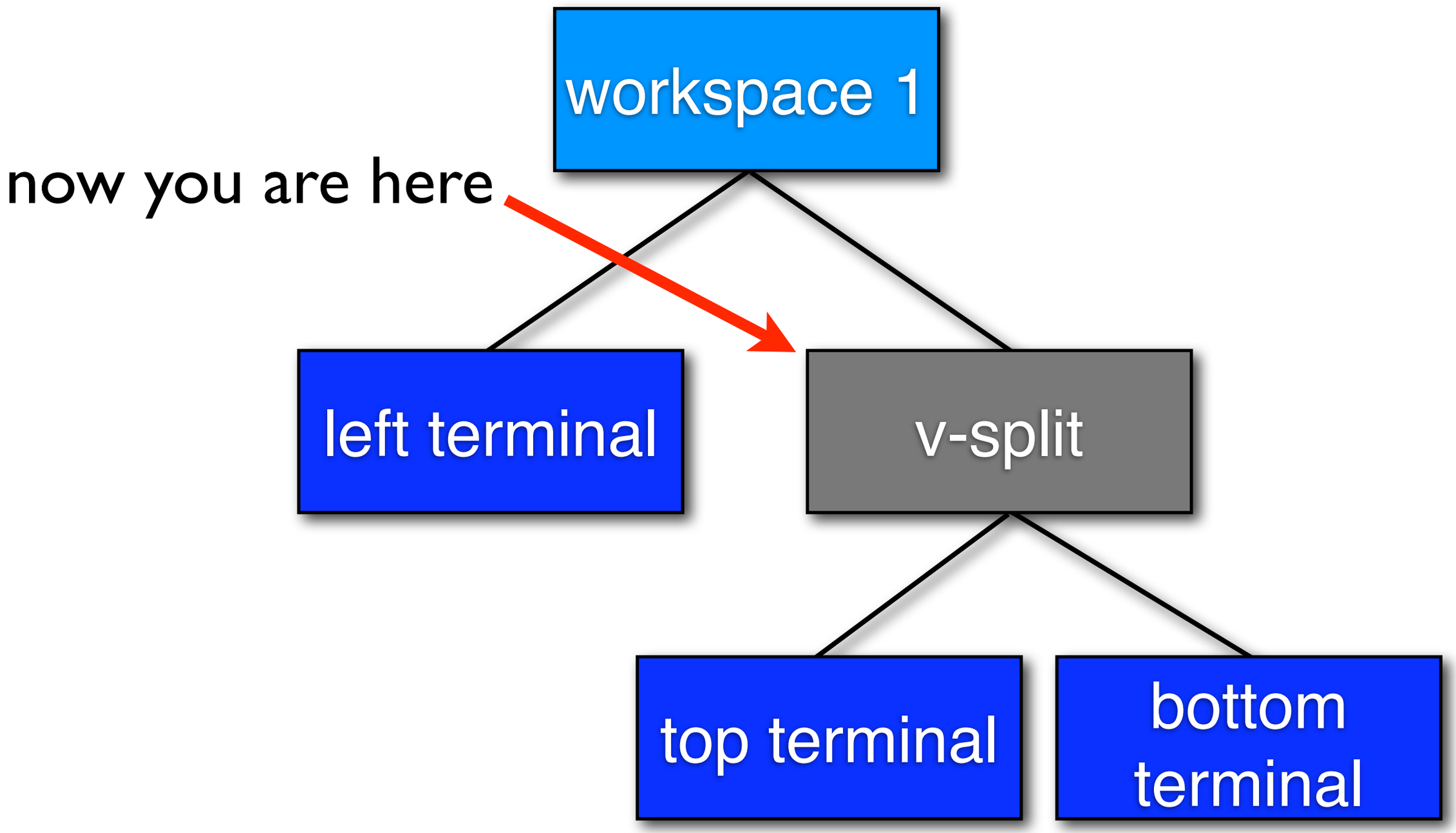

workspace 1
left terminal
v-split
top terminal
bottom terminal
in the tree, you are here now

But wait a second?

The new window automatically opened in the split container.

So, how can we open a new window next to the split container?

Just use level up (alt+u)!



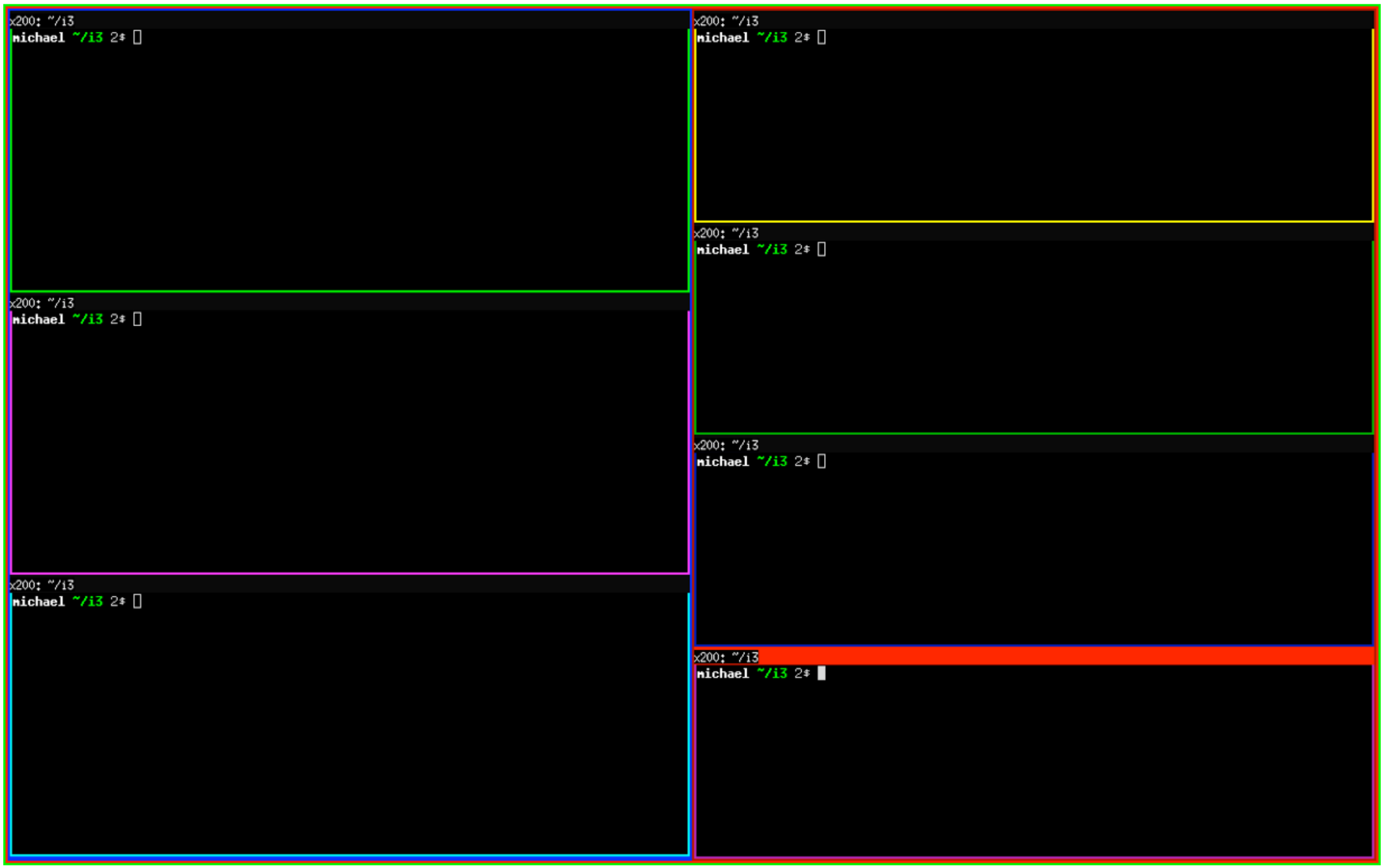
...then open a new terminal (alt+enter):

3: some layouts

The tree branch is supposed to make things easier. Let‘s have a look at some common layouts and how to create them.

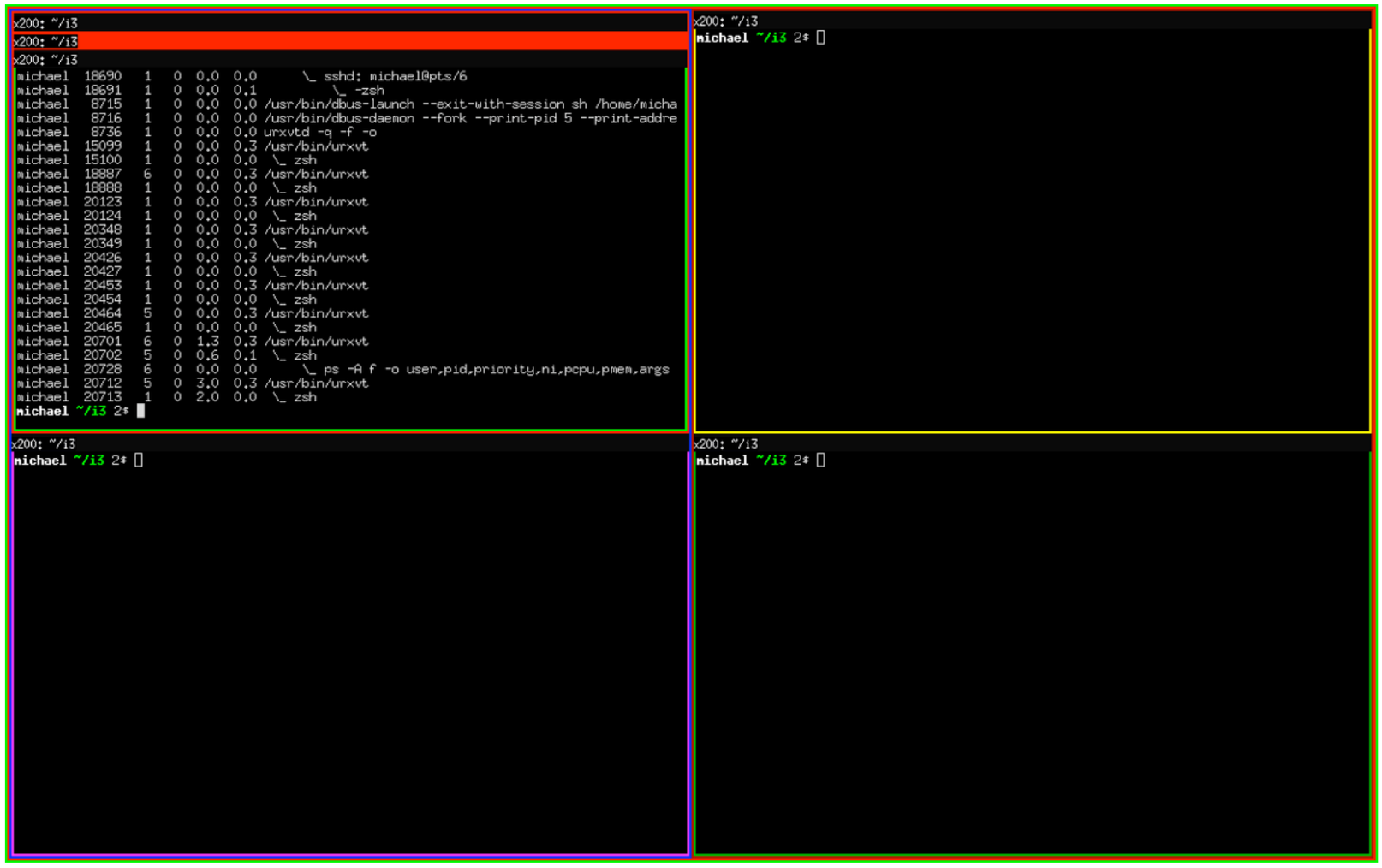
(of course, there is more than one way to do it)

lots of terminals, aligned in two columns:



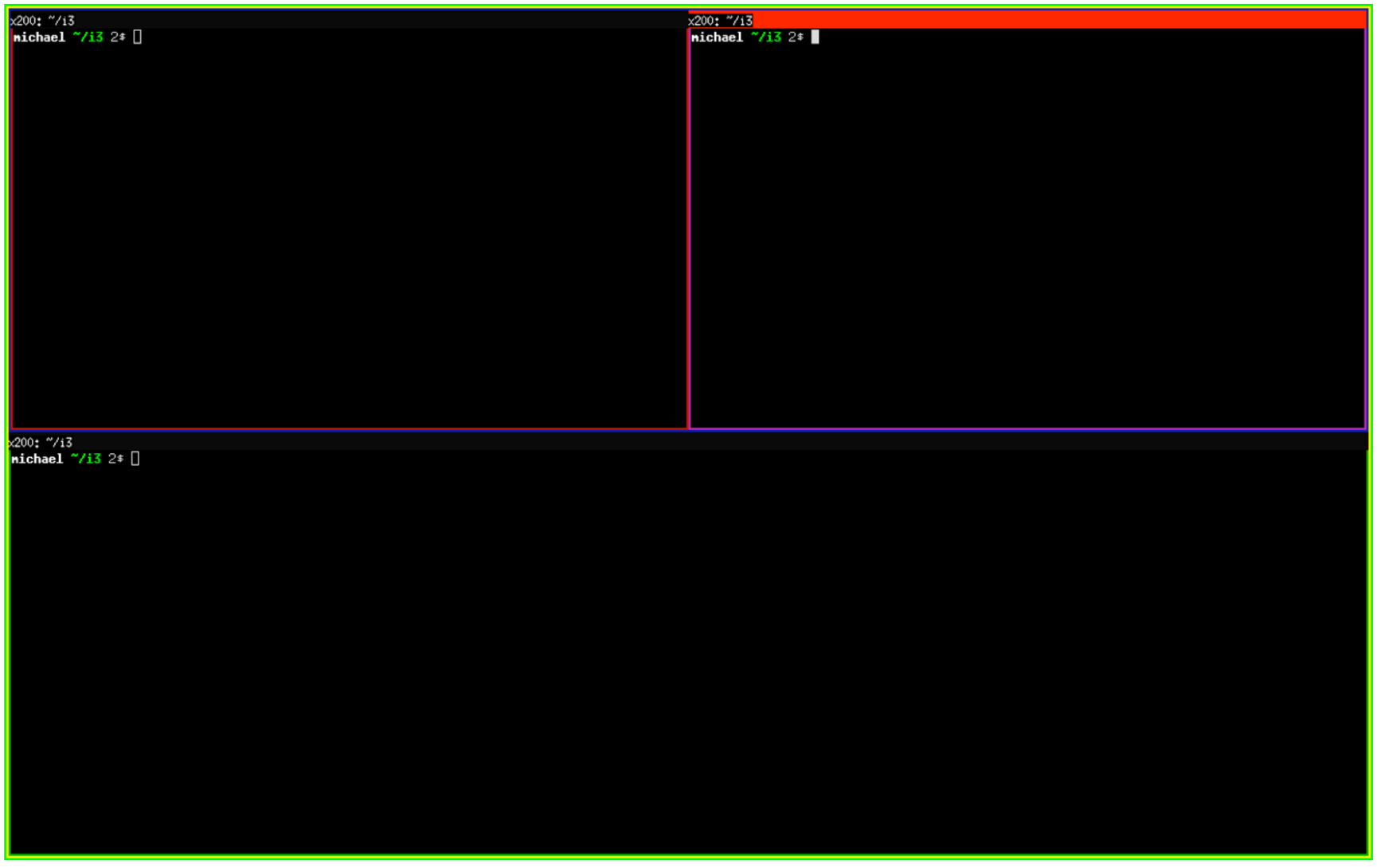
howto: open, split v, open, open, (left col done)
level up, open, split v, open, open

grid layout with multiple stacks inside



howto: open, split v, open, (left col done)
level up, open, split v, open (right col done)
split v, open, open, stacking (for each cell)

one big window on the bottom, two above



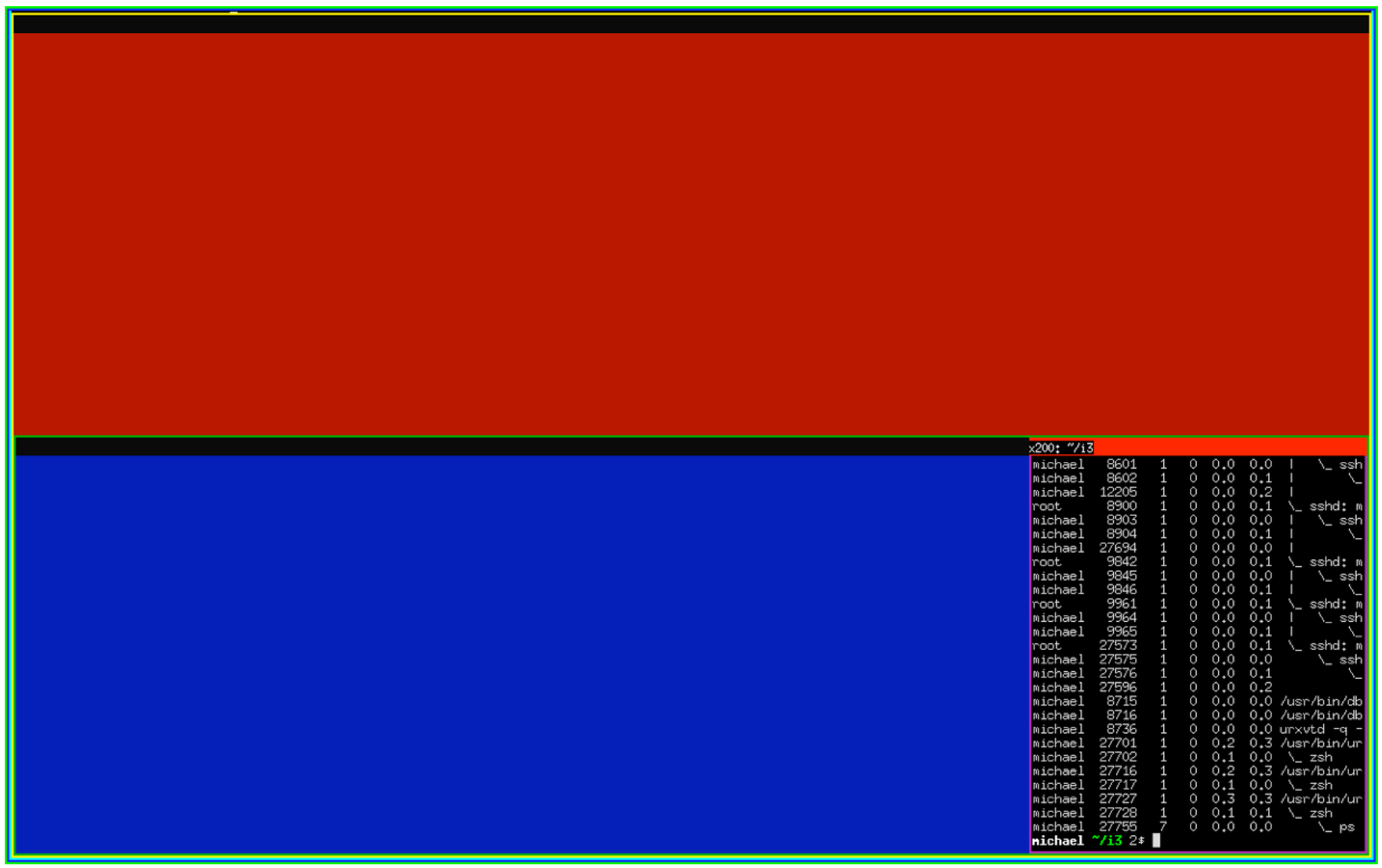
howto: split v, open, open (two rows done),
go up, split h, open (top terminals done)

4: goodies

If it was only for the layout, we probably would not have done the refactoring.

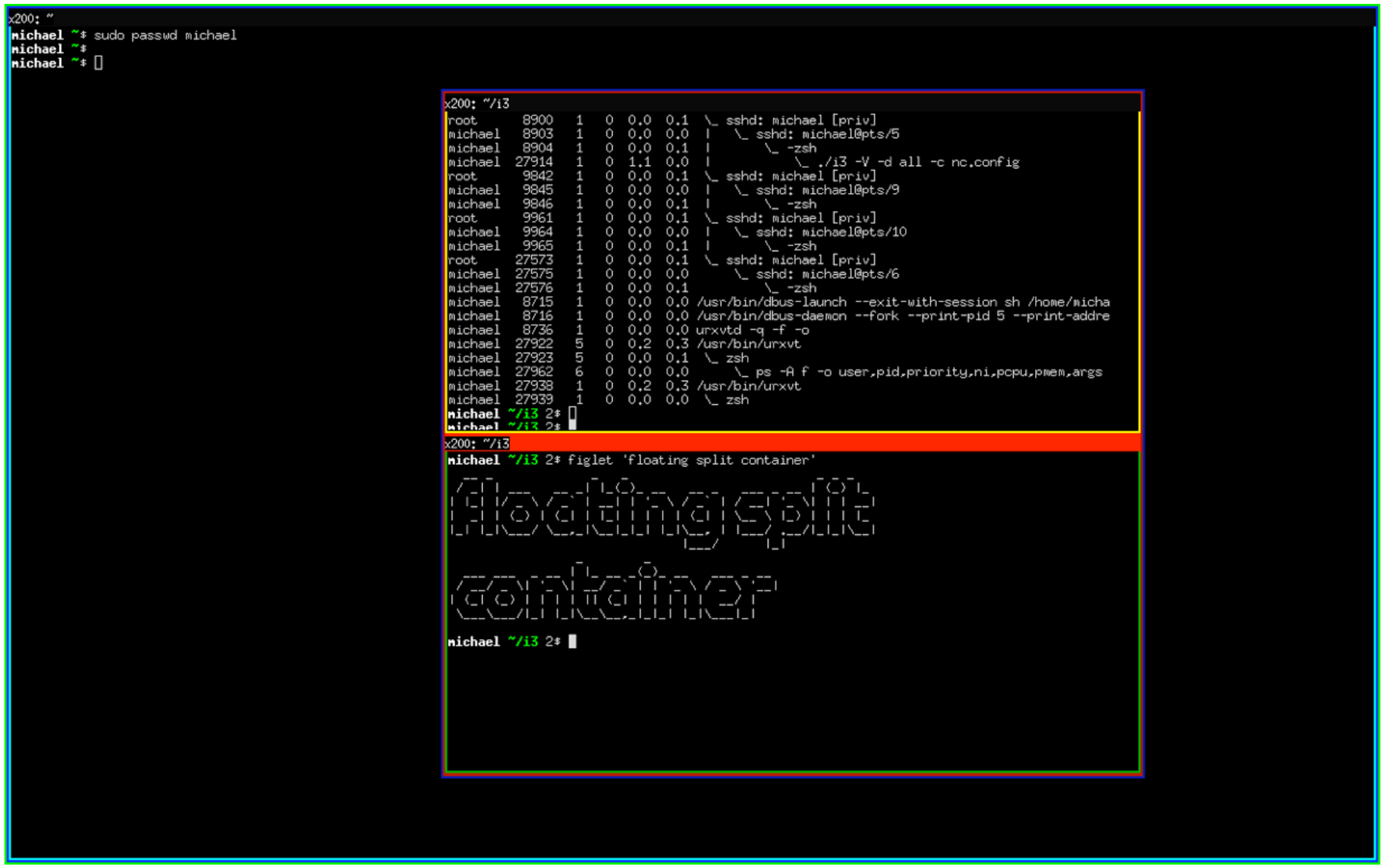
However, a lot of cool stuff is possible now.

Resize everything



NEW: You can now resize everything.

Float everything



NEW: You can set everything to floating, for example split containers.

Fullscreen everything

NEW: You can put everything into fullscreen mode.

Restore layout parts

In addition to providing the tree as JSON,
i3 can also read it back from JSON.

This way, you can restore a layout at any
place in the tree (container, workspace, ...).

This is handy for applications which have
many windows, like GIMP for example.

Survive inplace restarts

Thanks to the layout restore feature, surviving an inplace restart without losing your layout is easy: i3 stores the layout in ~/.i3/ before restarting and restores it afterwards.

Hooray!

Session saving

With the layout restore function and some scripting magic, it should be manageable to implement a session saver.

New command parser

The old cryptic commands are replaced
by explicit commands:

[next|prev] [h|v]
move [before|after] [h|v]
level [up|down]
layout [default|stacking|tabbed]
workspace <name>
fullscreen
kill
restore <path-to-layout>
...

New command parser

Also, you can now chain commands:

```
workspace 3; exec /usr/bin/xterm -e cmus
```

And make them match only specific windows:

```
[title="*Firefox*"] kill
```