High Performance Computing

NeoVim: A more a potent text editor

Martin Raum

Productivity gains from using a text editor that scales with your skills and requirements can be huge. We introduce NeoVim. Recall that there are other text editors of that kind, for instance, Vim, Emacs, and Helix.

When choosing, however, do not confuse text editing features with language integration; Rather install language integration in one of the editors that provide efficient text features.

An argument for a text editor that scales

LLMs (Large Language Models aka AI) generate responses that likely need post-processing. This can be achieved by further inquiring a model, but that can be a time consuming procedure that nullifies or degrades the advantages of LLMs. If your editor allows you to make changes quickly, you can break your LLM-workflow early and be more efficient.

NeoVim feature set

NeoVim offers vast possibilities. It is a mistake to attempt to learn them all at once.

As a first step, we merely discuss basic text edition in the style of nano, and two features beyond, which students reported as the initially most helpful ones.

In a second video, we look at more advanced features. A strategy can be to pick a small set of editing commands every second day for two to four weeks, and train them while programming before proceeding to the next set. This avoids cognitive overload and confusion.

You can open a file by providing it as an argument to the command nvim.

Navigating in NeoVim

You can navigate a text file in NeoVim with the arrow keys. Later this is not the recommended way, but for a start we stick to this customary navigation.

Inserting text

Inserting and deleting text works differently from other editors:
You need to change to insert mode. At the start, you are in normal
mode.

Press i to change to insert mode and place the cursor (a line
between two characters) in front of the cursor from normal mode
(a highlighted character). Now you can type as usually.

To return to normal mode, press the escape key. In case you are
unsure in which mode you are, observe that insert mode is
indicated at the bottom left.

Saving files is performed via a command. You can initiate a command while in normal mode by typing the colon character :. At the bottom you now see a command window. The command for saving is :w (for write). Confirm with enter.

Closing NeoVim

You can close NeoVim with the command :q (for quit). If you have unsaved changes, NeoVim will warn you and refuse to quit. Either save before quitting or force quit with the command :q!.

If you accidentally type q: you open an extended command window. Despite the different look you can now just type the quit command (namely q) after pressing i to enter insert mode. Confirm with enter as in the usual command window.

Help in NeoVim

In NeoVim you can find help on most if not all commands via the command :help TOPIC. For example in normal mode type :help :w and confirm with enter to open the help page for the write command.

Exit this page with the command :q.

Note that :help w will show a different help page describing the meaning of w (without leading colon) in normal mode.

Ideally keep yourself in the beginning from reading to much from the help pages. They are comprehensive and thus potentially overwhelming.

Undo and redo

To undo a text change hit u in normal mode. To redo the change use CTRL−r in normal mode.

Deleting and moving lines

To delete the whole line on which the cursor rests in normal mode hit dd. If you want to delete further lines following the current one, for instance three in total, you type d3d.

The deleted lines are stored (in a so called register). In normal mode you can insert the content of this register after the current line by hitting p.