Poprvé s Visual Studio Code

Špičkový editor VS Code (příkaz code) není jen editor: IDE, (live) viewer, SSH terminal.
Microsoft. Windows, Linux, macOS. Remote SSH & WSL & Dev Containers & Repositories.
Anketa StackOverflow 2021: 70+ % respondentů užívá primárně VS Code.
30000 extenzí na Marketplace.

Editace

cursor style, cursor blinking
smooth scrolling, sticky scroll, fast scroll (Alt+šipky)
word wrap, tab size, insert spaces, indentation, folding (ctrl+shift+[]), bracket pairs (color, underline)
render whitespace (selection), trim auto whitespace, selection highlight
editor hover, (clickable) editor links
minimap (side, scale)
editor suggest, quick suggestion delay
vertical rulers
intellisense (ctrl+space)
files encoding, files eol
breadcrumbs, outline
window zoom level
move line up/down alt+up/down
copy line up/down shift+alt+up/down
F1, transform to kebab case/lowercase/snake case/title case/uppercase
Zen Mode (Ctrl+K Z, zpět Esc+Esc)
https://code.visualstudio.com/docs/getstarted/tips-and-tricks

Konfigurace

Search settings a Filter settings
JSON: settings.json v (User) C:\Users\nobody\AppData\Roaming\Code\User a (Workspace) .vscode
Command Palette (F1) & Reload Window
Keyboard Shortcuts (Ctrl+K Ctrl+S)
Extenze: nabídka Extensions (Ctrl+Shift+X), též https://marketplace.visualstudio.com/

Podpora z Pythonu

Některé extenze potřebují Python a některé jeho balíčky. Ve Windows je bezproblémová instalace Pythonu z Microsoft Store (není-li Python, řádkový příkaz python otevře Microsoft Store a nabídne instalaci), balíčky se pak instalují řádkovým příkazem pip (Package Installer for Python), např. pip install numpy scipy matplotlib numba. (Distribuci Anaconda lze prý také použít, nikoliv pak však s pip.) Mají-li balíčky formu exe-souborů, je vhodné mít jejich polohu uloženou v proměnné (environment variable) PATH (výpis příkazem path nebo echo %PATH%). Poloha exe-souboru foo.exe se zjistí příkazem where foo, zjištěný adresář se přidá k obsahu proměnné PATH pomocí příkazu systempropertiesadvanced (vhodné zkrátit např. do spa.bat). Ne všechny balíčky se uloží do téhož adresáře. Polohu potřebného balíčku lze obvykle zadat i v konfiguraci extenze; je-li v proměnné PATH, není to třeba.

C/C++

…

Python

…

Fortran

Prvotřídní je extenze Modern Fortran. Bez dalšího s ní přijde barevná syntaxe (free & fixed source form). Větší sílu extenze nabere dodáním fortranského Language Serveru fortls (pythonovský balíček, tedy pip install fortls): přidá se Hover nápověda (myš nad jmény proměnných, datových typů, standardních i vlastních procedur), režim Intellisense (Ctrl+mezera nad čímkoliv, také automaticky při psaní), Goto/Peek aparát pro nahlížení a odskoky do jiných částí projektu na definice proměnných, typů a procedur (pravé tlačítko myši a klávesové zkratky) a podpora globálního Rename Symbol. Je-li dostupný fortranský překladač, s extenzí se aktivuje i linting (překlad editovaného souboru na pozadí po Ctrl+S, vyznačení chybových míst vlnovkou a poskytnutí chybové zprávy překladače). Fortranské soubory lze podrobit formátovačům findent nebo fprettify (po pip install findent fprettify). Samostatné programy lze rovnou spouštět pomocí extenze Code Runner (Ctrl+Alt+N), pro složitější projekty si člověk připraví skript nebo makefile, zkonfiguruje Code Runner a zůstane u něj, nebo vytěží extenzi Makefile Tools. Pro debugging je nabídnuta podpora gdb pomocí launch.json souborů.

Linting: Extenze podporuje linting pomocí GNU a Intel překladačů, neumí Nvidia a AMD (fortran.linter.compiler). GNU Compiler Collection (GCC), zahrnující gfortran, gcc, g++, gdb, make aj., se do Windows dobře instaluje z http://www.equation.com/. Cestu je při instalaci vhodné nastavit bez mezer, např. C:\gcc; adresář C:\gcc\bin by se měl do PATH dostat automaticky. Intel překladače ifort, ifx, icc, icx, icpc, icpx se distribuují jako součást volně dostupného Intel oneAPI HPC Toolkitu, závislého na Intel oneAPI Base Toolkitu; cesty k toolkitům se nastavují skriptem C:\Program Files (x86)\Intel\oneAPI\setvars.bat, pro linting pomocí ifort může být vhodné zadat do fortran.linter.compilerPath výstup z where ifort. V Settings (Ctrl+,) lze linteru dodat volby překladače (fortran.linter.extraArgs, default -Wall pro GNU a -warn all pro Intel) a cestu k include adresářům (fortran.linter.includePaths); pro udržení čistoty Workspace je vhodné doplnit adresář pro mod-soubory vytvářené během lintingu (fortran.linter.modOutput). Cestu k fortls.exe (fortran.fortls.path), findent.exe nebo fprettify.exe je nutné zadat, není-li poznačena v PATH; tyto soubory lze i někam na PATH zkopírovat.

Code Runner: Drobná extenze, která umožňuje navázat ke zdrojovým souborům podle typu (také podle přípony) sekvence příkazů uložené v code-runner.executorMap. Pro f90-soubory platí popis ve FortranFreeForm, příkazy mohou obsahovat proměnné $fileName, $fileNameWithoutExt, $dir (pokud nezahrnují, extenze doplní $fileName na konec sekvence automaticky), příkazy v sekvenci se oddělují symbolem && (společným ve Windows i Linuxu), nastavení specifické pro projekt se patří uložit do Workspace (nikoliv User). Sekvence také může odkázat specificky na překladový skript projektu nebo obecně na make a s pomocí makefile pak lze touto cestou překládat a spouštět i složité projekty. Konfigurovaná sekvence se z editoru aktivuje pomocí Ctrl+Alt+N. Z voleb extenze stojí za zmínku code-runner.runInTerminal a code-runner.saveFileBeforeRun.

Makefile Tools: Rozměrnější extenze, z níž lze do začátku vyzobnout schopnost aktivovat volání make v Command Palette (F1): Makefile: Build the current target (makefile.buildTarget), Build the target ALL, Build clean the current target ad. Frekventované volání bude vhodné definovat v Keyboard Shortcuts (Ctrl+K Ctrl+S), např. pro Build ALL se nabízí neobsazená Ctrl+Shift+A.

Pascal

Populární jsou extenze Pascal a OmniPascal – Open Preview s běžnými schopnostmi. Free Pascal bude vhodné vložit do PATH: C:\lazarus\fpc\3.2.2\bin\x86_64-win64.
Settings pro OmniPascal: omnipascal.defaultDevelopmentEnvironment (FreePascal), omnipascal.freePascalSourcePath (C:\lazarus\fpc\3.2.2), omnipascal.lazbuildPath (C:\lazarus\lazbuild.exe).

Matlab/Octave

Vhodná je extenze Matlab; extenze hlásící se k Octavu jsou slabé a mohou být v konfliktu s extenzí Matlab. Extenze Matlab navíc vytěžuje matlabovský linter mlint.
Pomocí extenze Code Runner lze konfigurovat řádkové spouštění skriptů pomocí příkazů matlab -batch myCode (bez přípony .m) nebo pomocí Octavu, octave myCode.m (takto v Linuxu; ve Windows s Octave v.7.2.0 nutný octave.bat s obsahem @C:\Octave\octave-launch %1).
Settings: matlab.matlabpath (C:\\Program Files\\MATLAB\\R2022a\\bin\\),
matlab.mlintpath (C:\\Program Files\\MATLAB\\R2022a\\bin\\win64\\mlint.exe)

LaTeX a spol.

Vhodné je instalovat extenzi LaTeX Workshop. Spolupracuje s LaTeXem předinstalovaným v systému (lze menší MiKTeX doplněný Strawberry Perlem i objemný TeX Live, pozor na případnou přednost starých verzí Perlu v PATH: where perl). Extenze poskytuje barevnou syntaxi, hover nápovědu a linting (raději jen onSave, při Ctrl+S). Vykresluje vzorce (Hover Preview) hned při editaci (engine MathJax). Ikona vpravo nahoře (též Ctrl+Alt+V) zobrazí přeložené PDF (Ctrl+myš pro zoom) s podporou synchronizace SyncTex (Ctrl+Alt+J v editoru, zpětně Ctrl+click v PDF). Užitečný je LaTeX panel vlevo (Ctrl+Alt+X) se Snippet View s možností výběru symbolů kliknutím.
Settings (100+): latex-workshop.latex.autoBuild.run (onSave), latex-workshop.linting.run (onSave),
latex-workshop.latex.autoClean.run (onBuilt), latex-workshop.hover.preview.scale,
latex-workshop.synctex.afterBuild.enabled, latex-workshop.view.pdf.invert, latex-workshop.view.pdf.scrollMode
a „latex view pdf"

Bitmapy (png, jpg), PostScript

Bitmapy se zobrazují out-of-the-box, pro PostScript jsou vhodné extenze PostScript Language (barevná syntaxe) a PostScript Preview (požadované ps2pdf a pdftocairo se naleznou samy v TeXovské distribuci). Postscriptový soubor se otevře v editoru a ikonou vpravo nahoře se získá okno Preview, dobře reagující na operace myší.

HTML

Barevná syntaxe, automatické vkládání párových tagů a řada dalších pomůcek (zkratky Emmet). S extenzí Live Preview (Microsoft), případně Live Server (populárnější), se připraví lokální web server pro rendering stránky během editace (raději jen při Ctrl+S), ikona vpravo nahoře pro otevření panelu. S extenzí Remote – SSH lze editovat rovnou na vzdáleném serveru (u nás přímo na karlovi až po jeho updatu, zatím třeba přes vaclava v /nfs1/www/users). Pozor na kódování HTML zdrojů, není-li defaultní UTF-8, lze kliknout vpravo dole na liště na zkratce Encoding, nahoře zvolit akci Reopen with Encoding a vybrat správné kódování; brzy pak své soubory raději konvertovat na UTF-8 (včetně údajů v jejich hlavičce). Prospějí četné extenze, jako HTML CSS Support, CSS Peek aj.
Settings (~10): livePreview.autoRefreshPreview (On Changes to Saved Files)

ReStructuredText (Sphinx)

Extenze reStructuredText & reStructuredText Syntax highlighting (potřeba pip install docutils esbonio) poskytují hlavně barevnou syntaxi. Jejich Live Preview není dokonalé, vhodnější je sphinxovské dokumenty překládat postaru v terminálu (make html) a zobrazovat v browseru.

Peacock

Extenze pro barevné odlišení jednotlivých workspace, vhodná, bývá-li jich otevřeno více najednou. V okně Command Palette (F1) se zadá Peacock a vybere se barva nebo další volby.

Polacode

Populární extenze autorů P & P (nikoliv Polacode-2022 aj.). Vytvoří snapshot vybraného textu v exkluzivním designu. V okně Command Palette (F1) se zadá Polacode, (podle zobrazeného návodu) se text přenese přes clipboard do okénka Polacode a klikne se na kolečko pod okénkem.

Print

…

WSL (Remote – WSL)

Extenze Microsoftu WSL (Windows Subsystem for Linux) umožňuje pomocí Windows VS Code (nikoliv linuxového VS Code) pracovat v prostředí WSL, tedy editovat soubory v souborovém systému WSL a vytvářet z nich aplikace spustitelné v prostředí WSL nebo v obecném Linuxu. Doporučeno je WSL 2; verzi lze zjistit (v terminálu VS Code, nikoliv v cmd.exe) příkazem wsl -l -v, povýšit WSL 1 na WSL 2 příkazem wsl --set-default-version 2. Na našich strojích s Linuxem je Ubuntu 22.04, je tedy rozumné instalovat tuto distribuci i do WSL.

Editor ve WSL se otevírá obdobně jako ve Windows, např. code . v adresáři projektu. Do takto otevřeného editoru se znovu instalují potřebné extenze (výzvou Install in WSL v nabídce extenzí). V Settings se mezi sadami User a Workspace objeví nová sada Remote [WSL: Ubuntu 22.04], vhodná pro obecné volby nastavené ve WSL jinak než ve Windows (např. cesty); volby nastavené ve Workspace budou mít ovšem přednost. Např. pro Modern Fortran není potřeba cesty explicitně nastavovat, nejsou-li ovšem změněny v nadřazené User konfiguraci; v takovém případě by se do Remote konfigurace měly vložit volby fortran.linter.compilerPath (/usr/bin/gfortran), fortran.formatting.path (/home/geo/.local/bin), fortran.fortls.path (/home/geo/.local/bin/fortls).

Remote – SSH

check OpenSSH Client in Windows Optional features
open in cmd, vscode Terminal, Altap 64b Command Shell: ssh-keygen -t rsa
cat c:\users\nobody\.ssh\id_rsa.pub do geof:~\.ssh\authorized_keys
optional: add geofxx do c:\windows\system32\drivers\etc\hosts
vscode Remote Explorer: Add (+) SSH Target
new vscode windows
geof: pip install fortls findent fprettify
Extensions: Modern Fortran, Code Runner, …
Settings: "fortran path", edit (in Remote!) formatting path, if needed:
    /home/lh/.local/bin, edit fortls path: /home/lh/.local/bin/fortls
    Command Palette (F1) & Reload Window
Explorer with remote dirs, open nebo code .
Settings: Workspace, "fortran mod" a Linter: Mod Output ${workspaceFolder}/mod
open ^ ctrl-s all
linter compiler gfortran, linter compiler path /usr/bin/gfortran
linter compiler ifort, linter compiler path /opt2/intel-2022.3/oneapi/compiler/2022.2.0/linux/bin/intel64/ifort

Remote X11

…

SSH FS

…

Etc

Code Spell Checker, Docker, Regex Previewer, Keymaps: Atom/Jupyter/Notepad++/Sublime/Vim
…

Odkazy: https://code.visualstudio.com/, https://en.wikipedia.org/wiki/Visual_Studio_Code
https://marketplace.visualstudio.com/
http://www.equation.com/servlet/equation.cmd?fa=fortran

SyncTex, https://github.com/James-Yu/LaTeX-Workshop/wiki/View

L. H., 1. 10. 2022