Real-time C++ Error Detection in NVChad

Param Matharoo

Introduction

This guide provides a streamlined process for integrating `clangd` into your NVChad setup, enabling real-time C++ error detection. This approach offers a lightweight and powerful alternative to full IDEs, providing immediate inline diagnostics and warnings without the overhead of heavy autocomplete features.

Step 1: Install clangd

First, ensure that `clangd` is installed on your system. Open your terminal and execute the following command to check its version:

```
clangd --version
```

If it is not installed, you can easily add it using your system's package manager:

```

For Debian/Ubuntu-based systems
sudo apt install clangd

For Arch-based systems
sudo pacman -S clang
```

Step 2: Enable clangd in NVChad LSP

Next, you need to configure NVChad to use the `clangd` language server. Open your LSP configuration file:

```
nvim ~/.config/nvim/lua/configs/
lspconfig.lua
```

Locate the `local servers` list and add `"clangd"` to the array:

```
local servers = { "pyright", "tsserver
", "clangd" }
```

After saving the file, restart NVChad to activate the new configuration. NVChad will now automatically recognize and provide diagnostics for your `.c` and `.cpp` files.

Step 3: Custom Key Mappings

For enhanced productivity, we can add a few custom key mappings. This guide will focus on a crucial mapping for copying text to your system's clipboard. You will add these to your `init.lua` file:

```
nvim ~/.config/nvim/init.lua
```

Include the following lines to enable clipboard-based copying with `<leader>y` in both normal and visual modes:

```
-- Yank (copy) to system clipboard
vim.keymap.set("v", "<leader>y", '"+y')
vim.keymap.set("n", "<leader>y", '"+y')
```

These commands tell Vim to use the system clipboard register (+) for yanking (copying) text, making it accessible outside of your Neovim session.