Accelerating Dotfiles Debugging

After managing our Dotfiles with Home Manager, one issue we may encounter is that each time we modify our Dotfiles, we need to run `sudo nixos-rebuild switch` (or `home-manager switch` if you are using Home Manager standalone) for the changes to take effect. However, running this command recalculates the entire system state each time, which is painful despite the many caching mechanisms already in place within Nix to accelerate this computation.

Taking my Neovim/Emacs configuration as an example, I make frequent modifications to them, sometimes dozens or even hundreds of times a day. Having to wait for `nixos-rebuild` to run for tens of seconds every time is a sheer waste of time.

Fortunately, Home Manager provides a mkOutOfStoreSymlink function, which can create a symlink pointing to the absolute path of your Dotfiles, thereby bypassing Home Manager itself and allowing your modifications to take effect immediately.

This method is effective under the premise that your Dotfiles content is not generated by Nix. For instance, my Emacs/Neovim configurations are native and are only linked to the correct locations through Nix Home-Manager's `home.file` or `xdg.configFile` .

Below is a brief explanation of how to use this function to accelerate Dotfiles debugging.

Assuming you have placed your Neovim configuration under `~/nix-config/home/nvim` , add the following code to your Home Manager configuration (e.g., `~/nix-config/home/default.nix`):

```nix
{ config, pkgs, ... }: let
  # path to your nvim config directory
  nvimPath = "${config.home.homeDirectory}/nix-config/home/nvim";
  # path to your doom emacs config directory
  doomPath = "${config.home.homeDirectory}/nix-config/home/doom";
in
{
  xdg.configFile."nvim".source = config.lib.file.mkOutOfStoreSymlink nvimPath;
  xdg.configFile."doom".source = config.lib.file.mkOutOfStoreSymlink doomPath;


  # other configurations
}
```

After modifying the configuration, run `sudo nixos-rebuild switch` (or `home-manager switch` if you are using Home Manager standalone) to apply the changes. From then on, any modifications you make to `~/nix-config/home/nvim` or `~/nix-config/home/doom` will be immediately observed by Neovim/Emacs.

This way, you can manage all your Dotfiles using a single nix-config repository, while frequently modified non-Nix configurations can take effect quickly, unaffected by Nix.

Loading comments...