

Projet de base

Ce projet établit une architecture de base avec meson pour le développement C.

Vous êtes libres de modifier et renommer les variables, cibles de compilations et autres fichiers importants. Vous pouvez également utiliser d'autres fonctionnalités de meson

Installation

Selon votre environnement, il vous est possible d'utiliser le projet de plusieurs manières différentes.

DevEnv: Windows / Linux / macOS

Cette variante exécute Nix et l'environnement de dev dans un conteneur Docker. Elle fonctionne de la même manière sur Windows, Linux et macOS, et évite d'installer Nix sur l'hôte.

Pré-requis : Docker installé.

Depuis la racine du projet :

Linux / macOS :

```
docker run -it --rm \
-v "$PWD:/work" \
-w /work \
nixos/nix:latest \
sh
```

Windows (PowerShell) :

```
docker run -it --rm `
-v "${PWD}:/work" `
-w /work `
nixos/nix:latest `
sh
```

Windows (CMD) :

```
docker run -it --rm ^
-v "%cd%:/work" ^
-w /work ^
nixos/nix:latest ^
sh
```

Dans le conteneur, entrer dans l'environnement DevEnv :

```
nix-env --install --attr devenv -f https://github.com/NixOS/nixpkgs/tarball/nixpkgs-unstable
devenv shell
```

Utilisation sans DevEnv (installation locale Meson)

Cette variante n'utilise ni Nix ni Docker. Vous installez directement Meson (et un compilateur C) sur votre machine, puis vous construisez le projet avec Meson + Ninja.

Pré-requis communs :

- Un compilateur C (GCC/Clang ou MSVC)
- Meson
- Ninja (recommandé, Meson l'utilise par défaut)

Linux (Debian/Ubuntu) :

```
sudo apt update
sudo apt install -y build-essential meson ninja-build
```

macOS (Homebrew) :

```
brew install meson ninja
# Clang est fourni avec Xcode Command Line Tools
xcode-select --install
```

Windows (recommandé : MSYS2) :

```
# 1) Installer MSYS2, puis ouvrir "MSYS2 UCRT64" (ou MINGW64)
# 2) Mettre à jour et installer la toolchain + meson + ninja
pacman -Syu
pacman -S --needed base-devel mingw-w64-ucrt-x86_64-toolchain mingw-w64-ucrt-x86_64-meson m
```

Windows (Visual Studio / MSVC) :

- Installer “Visual Studio Build Tools” (ou Visual Studio) avec le workload “Desktop development with C++”.
- Installer Meson + Ninja via Python/pip (ou via Chocolatey).

Avec Python/pip :

```
py -m pip install --user meson ninja
```

Utilisation

```
meson setup build
meson compile -C build
meson test -C build
```