Introduction au WAT

Le WAT (WebAssembly Text Format) est un format de fichier qui permet de représenter du code WebAssembly sous forme de texte humainement lisible. Sa syntaxe est issue des assembleurs classiques et utilise des mnémoniques pour représenter les instructions de WebAssembly. (func, module, import ...)

Le WAT est principalement utilisé pour le développement et le débogage de code WebAssembly car il permet de lire et de modifier facilement le code source.

Il est aussi possible de se servir de WAT comme une destination de compilation pour pouvoir analyser les instructions utilisé.

Par exemple :

```
wasm-bindgen <source>.rs --out-dir <output_directory> --out-name <output_name> --target web
--no-modules
```

- `<source>.rs` est le fichier source Rust que vous souhaitez compiler.
- `<output_directory>` est le répertoire de sortie où le fichier WAT sera généré.
- `<output_name>` est le nom du fichier WAT généré.

La commande a été faite avec WASM bindgen mais il est tout à fait possible de le faire avec d'autres outils.

Le WebAssembly Text Format (WAT) est un format ouvert et indépendant de la plateforme qui peut être compilé en binaire WebAssembly (.wasm) pour être exécuté dans un navigateur ou un environnement d'exécution WebAssembly.

Il est important de noter que le WAT n'est pas un langage de programmation à part entière, mais plutôt un format de fichier pour représenter le code WebAssembly. Cependant, certaines librairies ou outils peuvent vous permettre d'écrire directement en WAT.

Le WAT présente des avantages pour le développement et le débogage, ainsi que pour l'apprentissage et la pédagogie. En utilisant le WAT, les étudiants et les développeurs débutants peuvent mieux comprendre comment fonctionne WebAssembly en étudiant le code source sous forme de texte.

Le WAT permet par d'ailleurs une meilleure compréhension des instructions de WebAssembly et de leur utilisation, ce qui peut aider à améliorer la compréhension des concepts de bas niveau tels que

la manipulation de la mémoire et les opérations de bas niveau.

Revision #2
Created 2023-05-05 14:02:59 UTC by Noé Larrieu-Lacoste
Updated 2023-05-05 14:04:10 UTC by Noé Larrieu-Lacoste