2017-04-15

Maxime Goyette et David Michel Donais

Université de Sherbrooke, Département d’informatique

Dans le cadre du cours IFT55555555

Manuel du programmeur

# Logiciels pour la compilation

## Essentiel

* Meson (version 0.36.0 ou supérieure)
* Python (version 3.4 ou supérieure)
* Ninja (version 1.5 ou supérieure)
* Système d’exploitation GNU/Linux

## Émulateur graphique

* Bibliothèque SFML-dev (version 2.4 ou supérieure)
* Compilateur avec support de c++14

## Émulateur Raspberry Pi

* gcc-arm-none-eabi (testé avec la version 6-2017-q1)
* wget

## Outils pratiques

* xxd
* git

# Hiérarchie du projet

* **Frankenstein**
  + **application** 
    - Contient un émulateur graphique, sfml\_emulator, et un émulateur console, term\_emulator, utilisés pour le développement dans un environnement graphique GNU/Linux.
  + **emulator**
    - Contient les fichiers de développement de la bibliothèque de l’émulateur.
    - **include**
      * Contient les fichiers d’entête de la bibliothèque de l’émulateur.
    - **test**
      * Contient quelques tests unitaires pour l’unité central de traitement (CPU), utilisant la bibliothèque gtest.
      * **roms**
        + Contient des images de cartouche NES de tests, utilisées pour des tests d’intégrations.
  + **kernel**
    - Contient les fichiers de développement pour l’image s’exécutant sur le Raspberry Pi.
  + **Subproject**
    - Contient les bibliothèques externes utilisés dans le projet.
    - **circle**
      * La bibliothèque Circle, de rsta2, est un mini système d’exploitation pour le Raspberry Pi.

# Environnement de développement

Le développement et la compilation du projet ont été réaliser sur des systèmes d’exploitation GNU/Linux, plus précisément la distribution archlinux. Le projet devrait pouvoir être configuré sur un environnement similaire, pour peu que les dépendances soient à jour.

Il est possible d’utiliser l’éditeur ou l’éditeur de développement intégré de son choix, tant que ce dernier supporte les projets «libres», où les commandes de compilation et d’exécution peuvent être spécifié, ou encore les projets Meson, directement. D’expérience, il était facile de configurer le projet avec l’IDE *Netbeans*, dans sa dernière version, ainsi que *Gnome Builder* et *KDevelop*.

# Compilation

L’outil utilisé pour la création des exécutables et de l’image pour Raspberri Pi 3 se nomme Meson. Meson est un moteur de production (« build management system ») relativement nouveau. Il est très semblable à l’outil CMake, mais avec une syntaxe DSL accessible et une documentation en ligne compréhensible (**burn**?). Il possède aussi la notion de compilation sur une plateforme différente de l’hôte, intégré directement au DSL. Ce trait permet de compiler l’image ARM pour le Raspberry Pi et les exécutable natifs, d’une seule commande.

La première étape pour configurer le projet Meson est de créer un dossier de compilation (build directory), ici nommé **build**, ainsi qu’un dossier d’installation, ici nommé **install**. Le premier est utilisé pour la compilation des objets et fichiers temporaires, alors que le second est le dossier de destination où se retrouveront tous les fichiers nécessaires pour exécuter le projet sur l’environnement GNU/Linux et le Raspberry Pi.

Bibliographies et source

Roms: blargg + https://github.com/christopherpow/nes-test-roms