

ESGI

Linter C

Bilan

Alexis Desjardins
Louis LAURENT
Lou ANGRAND
11/12/2018

Table des matières

INTRODUCTION2

ANALYSE DE L'APPLICATION2

LISTE DES STRUCTURES IMPORTANTES2

FONCTIONS PRINCIPALES2

INSTALLATION3

UTILISATION3

BILAN DU PROJET4

Introduction

L'objectif de ce projet est de construire un Linter.

Un Linter est un outil d'analyse de code source. Il doit permettre de détecter des erreurs de syntaxe mais aussi du non-respect de convention de codage pour les programmes C uniquement.

Ce projet a été réalisé par :

- **Louis LAURENT**
- **Lou ANGRAND**
- **Alexis DESJARDINS**

Analyse de l'application

Liste des structures importantes

Structures de configurations :

- ConfigKey
Structure contenant une clé ainsi que ses paramètres.
- BValue
Paramètre de key sous forme :
`true`
- CValue
Paramètre de key sous forme :
`- Rule1 = 20`
- Value
Paramètre de key sous forme :
`- Rule2`

Fonctions principales

Fonctions de récupération de la configuration :

- loadConfig (nom)
Renvoie une liste chaînée ConfigKey contenant l'intégralité des clés contenues dans le fichier nommé *nom*.
- fusionKey (conf1, conf2)
Intègre une liste ConfigKey 2 dans une autre.

Fonctions de la partie 2 :

- functionTestIndent(path)
Affiche les erreurs dans le cas où l'indentation est incorrecte.
- functionStartComment(path)
Affiche une erreur dans le cas où il n'y a pas de commentaire en début de fichier.

- `functionTestTrailingSpace(path)`
Affiche des erreurs dans le cas où il y a un espace en fin de ligne.
- `processOperators(path)`
Affiche les erreurs s'il n'y a pas d'espace entre les opérateurs.
- `functionTestCommat(path)`
Affiche des erreurs quand il n'y a pas d'espace après une virgule.
- `checkCharactersNumbers(path, numberLimit)`
Affiche les erreurs quand le nombre de caractères d'une ligne dépasse la limite
- `checkLinesNumbers(path, numberLimit)`
Affiche une erreur quand le nombre de lignes dans le fichier dépasse la limite.

Fonctions de la partie 3 :

- `noMultiDeclarations(filePath)`
Affiche une liste d'erreurs des lignes sur lesquelles plusieurs variables sont déclarées.
Renvoie le nombre d'erreurs.
- `unusedVariables(filePath)`
Affiche une liste d'erreurs des variables déclarées mais non utilisées. Renvoie le nombre d'erreurs.
- `undeclaredVariables(filePath)`
Affiche une liste d'erreurs des variables utilisées mais non déclarées. Renvoie le nombre d'erreurs.

Installation

Un makefile est présent au sein du dossier INSTALL. Pour le lancer il faut effectuer : « make create » .
Le binaire sera généré dans le dossier bin au nom de « LinterESGI ».

Utilisation

Le fichier *main.lconf* est le fichier de configuration par défaut. Les différents paramètres de configuration de celui-ci sont présents dans le sujet. Vous pouvez étendre d'autres fichiers de configuration, les paramètres du *main.lconf* sont prioritaires sur ceux hérités.

Bilan du projet

Nous avons été surpris par le rythme de l'alternance et nous sommes dans l'ensemble mis au travail trop tard et avons mal géré notre temps. Nous avons chacun plus ou moins aimé le sujet. Nous sommes globalement satisfaits du résultat final.

Le groupe n'a pas eu de problème interne, nous nous sommes réparti les fonctions selon les préférences de chacun, rien n'a été imposé.