

Auteurs: -Florian DAJON - | [ROLE A AJOUTER]
-Pierre-Alexandre GROSSET | [ROLE A AJOUTER]
Destinataires: MEMBRES GROUPE COREWAR
-Amdjad AHMOD-ALI | [ROLE A AJOUTER]
-Romain NACOUÉLE | [ROLE A AJOUTER]
Date: -26 Avril 2023

MEMO N°001

Charte de rédaction des Tests Unitaires et Fonctionnels

Ce mémo a pour objectif de définir la politique de tests qui sera tenue à travers le projet "Corewar" qui a débuté le Lundi 23 Avril 2023 et se terminera le 28 Mai 2023.

Définitions des Tests Unitaires et Fonctionnels

-Test Unitaire : Un test permettant de tester les actions d'une fonction individuelle. D'identifier les failles et les cas d'erreurs qui pourraient porter atteinte au bon fonctionnement du programme. La plus petite unité de code (ligne par ligne)

-Test Fonctionnel : Un test permettant de tester les actions d'un ensemble de fonctions. Assurer que le système fonctionne de manière automatisée. Tester un ensemble de fonctions.

Définition d'un standard de macro de Tests

-Afin de faciliter la rédaction de tests et de rendre ce processus plus efficient nous mettrons tout les tests dans un format standardisé :

"Test([FONCTION À TESTER], tests_[FONCTION], .init=redirect, .timeout=1)"

- "[FONCTION À TESTER]" : Indiquer la fonction que vous testez. 1 Test = 1 Fonction
- "tests_[FONCTION]" : Permet d'identifier le Test. Utile lorsqu'un test échoue.
- ".init=redirect" : Permet de rediriger la STDIN et STDOUT. TOUJOURS LE METTRE.
- ".timeout=1" : Permet de limiter dans le temps un test (empêche de boucler à l'infini)

Procédure de création de tests

-Afin de réaliser les tests et de couvrir le maximum de lignes de programme et de branches conditionnels voici la technique à suivre :

- 1) Effectuer les tests à partir de la plus petite unité de code : tester les fonctions en profondeur d'abord
- 2) Tester les fonctions lignes par lignes dans l'ordre
- 3) Assurez vous de créer des tests utiles qui testent toutes les possibilités existantes
- 4) Si vous identifiez une faille ajouter le code nécessaire pour boucher la faille
- 5) Si une fonction présente une faille nécessitant une refonte de la fonction, signalez-le à son responsable

Cas triviaux

- Afin de tester les failles du programme, assurez-vous que les pointeurs ne sont jamais égaux à NULL.
- Testez au maximum les malloc et allocations de mémoires à l'aides des MALL_VS[0-10]
- Vérifiez si les pointeurs que vous envoyez en paramètre sont différents de NULL

Contact

- Pour toutes questions concernant ce mémo merci de vous adresser à l'auteur de ce mémo via l'adresse email suivante :

florian.dajon@epitech.eu

(Florian DAJON, [ROLE A AJOUTER])

pierre-alexandre.grosset@epitech.eu

(Pierre-Alexandre GROSSET, [ROLE A AJOUTER])

- Le prochain mémo (Mémo N°[002]) sera à définir lors de la prochaine réunion qui se déroulera le 27 avril 2023.