

Compte rendu AN

Audit normatif sur le code C / ProSE

Appel à l'ordre :

Date de la réunion : 08/06

Heure début et fin : 10h → 12h15

Lieu : Amérique

Réunion organisée par : Frédéric JOUAULT Type de réunion : Audit normatif Rédacteur : Camille LENNE Point météo : Soleil, quelques modifications à faire	Participants : Théo BÉNARD, Elisa DECLERCK, Thomas ROCHER, et Frédéric JOUAULT
---	--

Objectif de la réunion :

Audit normatif sur le code source C réalisé par Elisa et Thomas.

Sujets abordés et actions menées :

- Ajouter des mq_unlink après les mq_open pour éviter de créer une BAL si elle existe déjà.
- Placer le sémaphore avant la fonction receive_data dans Dispatcher et Postman pour permettre un sem_post lors de l'appel à stop et ainsi nettoyer correctement.
- Dans Dispatcher, utiliser le code de retour de recv en cas de déconnexion plutôt qu'un caractère spécial.
- Vérifier les erreurs en premier dans Dispatcher et effectuer la vérification avant tout le reste du décodage.
- Dans Postman, calculer la longueur de la chaîne avec strlen avant la boucle for dans postmanSendMessage() et utiliser strcpy plutôt qu'une simple affectation.
- Dans ProxyGUI, remplacer le nombre "18" en dur dans le paramètre par une macro appropriée.
- Dans ProxyLogger, ne pas accepter de valeurs incorrectes dans les fonctions snprintf en utilisant des asserts.

Commentaires :

Pastille Verte Pale / 85 % d'avancement

Prise de note détaillée de l'AN

TOUS

- ajouter des mq_unlik après les mq_open pour éviter de créer la BAL si elle existe encore.
- dans le run pthread_exit((void*)ERROR_STATE_FAILURE); marche pas, je capte pas demande explication à Elisa
- enlever le "&=" par -->
- dans les free ne pas faire de return par ce que sinon tu peux pas free la suite s'il y a une erreur
- enlever le close de ma mq dans tous les MQ_send
- mettre le mq_attr en local

Dispatcher

#.h

#.c

- mettre le semaphore avant (dans) le receive data comme ça tu peux faire un sem_post quand tu appelles stop et tu peux déblayer oklm
 - dans le receive_data, au lieu de mettre un caractère spécial met le code de retour de recv en cas de déco (l'erreur).
 - s'il y a une deconnexion, receiveData
- dispatcher toujours en receiveData -> s'arrete quand receiveData dit qu'il y a une erreur.
- s'il y a une deconnexion -> receiveData direct en attente sur le semaphore.
- postman
- faire un while dans le run qui receive en boucle.
- faire la vérification de l'erreur en tout premier et pas après tout le reste du décodage.

Postman

#.h

#.c

- mettre le semaphore avant le receive data comme ça tu peux faire un sem_post quand tu appelles stop et tu peux déblayer oklm

- postmanSendMessage() -> faire un strlen avant la boucle for et faire strcpy au lieu d'un simple "="

ProxyGUI

#.c

- enlever le 18 en dur dans le param et mettre la macro à la place
- faire en sorte de ne pas prendre en compte un sendingState faux dans le snprintf -> mettre un assert

ProxyLogger

#.c

- faire en sorte de ne pas avoir du faux dans le snprintf -> mettre un assert
- plutôt que de diviser par /16 l'id tu peux faire un décalage de bit
- prendre en compte le ID=0
- mettre BUFFERSIZE en param du snprintf plutôt que d'adapter une taille