

Rapport Projet 2

réalisé par Fréjoux Gaëtan et Niord Mathieu

Table des matières

Organisation de notre code	1
Les fichiers ajoutés	1
utils.c/h	1
Les fichiers déjà présents	2
Les fichiers principaux :	2
client.c	2
master.c	2
worker.c	2
Les fichiers secondaires :	2
master_client.c/h	2
master_worker.c/h	2
myassert.c/h	2
Protocoles de communication	2
Sémaphore	2
Les tubes	3
Nommés	3
Anonymes	3

Organisation de notre code

Les fichiers ajoutés

- utils.c/h

Nous avons décidé d'ajouter aux fichiers déjà existants un couple c/h nommé "utils", nous avons utilisé ces fichiers pour y intégrer les "includes" nécessaires ainsi que nos fonctions pour que les fonctions dans les autres fichiers ne soient pas trop longues.

Les fichiers déjà présents

Les fichiers principaux :

- client.c
 - ce fichier contient le code permettant le bon déroulement d'un client pour chaque ordre demandé.
- master.c
 - ce fichier contient le code permettant le bon déroulement d'un master pour chaque ordre demandé.
- worker.c
 - ce fichier contient le code permettant le bon déroulement d'un worker pour chaque ordre demandé.

Les fichiers secondaires :

- master_client.c/.h
 - ces fichiers contiennent :
 - les structures :
 - mS (pour le master)
 - ThreadData (pour les threads)
 - des fonctions pour le bon fonctionnement du couple master/client
- master_worker.c/.h
 - ces fichiers contiennent :
 - la structure pour wS pour les workers
 - les fonctions pour le bon fonctionnement du couple master/worker
- myassert.c/.h
 - ces fichiers contiennent une fonction améliorée de assert

Protocoles de communication

Sémaphore

Nous avons décidé d'utiliser seulement un sémaphore permettant de gérer le cas où il y aurait plusieurs clients voulant communiquer avec le master pour qu'ils ne puissent pas s'interrompre.

Les tubes

Nommés

Nous avons utilisé 2 tubes nommés, l'un permettant au client d'écrire pour le master et l'autre permettant au master d'écrire pour le client.

Anonymes

Nous avons aussi utilisé des tubes anonymes entre le master et le worker, un pour que le master puisse écrire au premier worker, un entre chaque worker pour que le précédent envoie l'information aux suivants et enfin un même tube pour chaque worker permettant d'écrire au master.