

Rapport projet de système

Leroy Florent et Salles Théo

Décembre 2023

Table des matières

1	File synchronisée	2
1.1	file_sync.h	2
1.2	file_sync.c	2
2	Parseur	2
3	Lanceur de commande	2
4	Client	3

1 File synchronisée

La file synchronisée est construite comme une bibliothèque avec un fichier interface *file_sync.h* et un fichier implémentations. *file_sync.c*.

1.1 file_sync.h

Le module est composé de 4 fonctions. Une fonction de création de la file synchronisée, une qui libère les ressources allouées à la file et 2 autres fonctions pour ajouter ou retirer des éléments de cette file.

1.2 file_sync.c

Toutes les informations de la file tel que les sémaphores, le buffer de données et les têtes de lecture et d'écriture sont conservées dans un segment de mémoire partagé qui est créé dans la fonction *create_file_sync* avant l'initialisation des sémaphores, des têtes de lectures et du buffer. Chaque action sur la file commence par l'ouverture du segment de mémoire partagée puis pour la fonction *destroy_file* on détruit toutes les sémaphores puis on libère le segment de mémoire partagée. Pour les fonctions *defiler* et *enfiler* c'est un simple problème consommateur/producteur avec les sémaphores.

2 Parseur

Le module parseur est une réutilisation du module *analyse* donné lors du tp 2 avec des tests rajoutées notamment sur les malloc.

3 Lanceur de commande

Le nom des tubes est défini par une norme choisie qui s'applique entre le lanceur et le client. Cette norme veut que le nom d'un tube soit une chaîne de caractère qui définit l'utilité du tube suivie du pid du client afin que chaque client possède des tubes qui lui sont propres.

- TUBE_CLIENT_ : est le tube dans lequel le client écrit les informations pour le lanceur, il est créé par le client.
- TUBE_RES_CLIENT_ : est le tube dans lequel le résultat de la commande sera écrit et envoyé au client, il est créé par le lanceur.

- `TUBE_ERR_CLIENT_` : est le tube dans lequel une possible erreur lors de l'exécution de la commande sera écrit et envoyé au client, il est créé par le lanceur.
- `BUF_SIZE` : Taille maximale en nombre de caractère des commandes avec leurs options.
- `CMD_SIZE` : Taille maximale en nombre de du nom de la commande (taille déterminer à partir du nombre de caractère de la plus grande commande linux avec une marge de sécurité).
- `PID_SIZE` : Taille maximale en nombre de caractère du pid du client.

4 Client