Rapport du projet Systèmes d'Exploitation Avé Cesar

CARBONNIER Nicolas, SOTTAS Jean-Charles

29 avril 2017

1 Description du programme

1.1 Structure INFO

La structure sauvegarde le nombre de lignes dans la variable taille, et sauvegarde chaque ligne dans le tableau ligne.

1.2 Structure ENCODAGE

La structure sauvegarde un caractère, le numéro de décalage qui doit être fait et le thread qui emploie cette structure.

1.3 Processus Directeur

Le processus directeur crée deux tubes, un pour que le père puisse écrire et le fils lire et l'autre pour que le fils puisse écrire et que le père lise, pour chaque processus fils créé. Le père va ainsi envoyer une ligne du fichier principal dans chaque tube. Il va recevoir dans l'autre tube soit le message en clair, soit "C" correspondant au fichier qui a été crypté.

1.4 Processus Chef-Equipe

Le processus chef d'équipe découpe le message reçu dans le tube, et obtiendra le nom du fichier avec son chemin, soit c pour crypter soit d pour décrypter. Il ouvre le fichier et commence à envoyer aux threads un caractère ; il enverra aux threads décrypter sinon encrypter. Il stockera leurs retours dans un buffer qui sera ensuite écrit dans un fichier ou envoyé dans le tube pour le processus directeur.

1.5 Processus Fils

Le processus fils reçoit un caractère et le numéro de décalage. Il vérifie si le caractère est une lettre ; si c'est une lettre il vérifie si celle-ci est une majuscule,

puis lui applique la formule. Si celle-ci n'est pas une lettre alors le caractère est retourné.

1.6 Ecrire fichier

La fonction reçoit le buffer ainsi que le nom du fichier avec son chemin. La fonction va alors chercher à modifier le nom du fichier pour permettre le rajout de l'extention "crypter". Une fois ajouté on crée le fichier avec les mêmes droits et la même destination puis on écrit le message codé.

2 Les difficultés

Pour le tube qui retourne au directeur, la difficulté a été le choix du message de retour lorsque le retour doit être vide : il est impossible de lire dans un tube vide.
L'autre difficulté a été lors du test du programme avec valgrind où il détecte des erreurs car les buffers n'étaient pas initialisés. Une initialisation de chaque buffer a permis de résoudre le problème. Valgrind détecte de la mémoire non "free" qui est "steal reachable" ; après vérification le programme libère cette partie lorsque le programme se termine : cela ne pose plus de problème pour la mémoire.