Projet Systèmes d'Exploitation
Avé Cesar

S. Gougeaud

Licence Informatique (2016/2017) – UVSQ

1 Objectif

Le projet consiste en l'implémentation d'un mécanisme de chiffrement/déchiffrement, basé sur le chiffre de Cesar (`fr.wikipedia.org/wiki/Chiffrement_par_décalage`). Un ensemble de fichiers d'entrées, appelés messages, contiendront soit un clair, soit un chiffré, qu'il va donc falloir traiter. Chaque fichier est donné avec un entier positif, indiquant le paramètre du chiffrement, et un sens de traitement : chiffrement ou déchiffrement. Le programme principal créé un processus par fichier à traiter, et un thread sera affecté à chaque mot du message, pour sa traduction.

Chaque projet doit se faire en **binôme**. Le rendu du projet doit se faire avant le **30 Avril 2017 à 23h59** (heure d'été, France métropolitaine) par mail à l'adresse suivante, en fonction de votre chargé de TD :

`arthur.loussert@inria.fr`
`sebastien.gougeaud@uvsq.fr`

Le mail doit contenir une archive tar.gz composée :

- des sources du projet faites en langage C ;
- des directives de compilation pour le projet (script, Makefile) ;
- d'un rapport de deux/trois pages décrivant brièvement votre implémentation ainsi que les problèmes que vous avez rencontrés.

Chaque binôme sera soumis à une soutenance de 15 à 20 minutes durant laquelle il devra présenter son code, faire une démonstration et répondre aux questions qui lui seront posées.

2 Pré-requis

- Gestion des fichiers.
- Gestion des processus/threads.
- Communication inter/intra processus.
- Gestion des mutex/conditions.

3 Fichiers d'entrée du programme

Le programme possède deux types de fichiers d'entrée :

1. le fichier principal – possède sur chaque ligne, le chemin d'un fichier message, le pas du chiffrement et son sens (c pour chiffrement et d pour déchiffrement), chacun étant séparé par un ';' ;

```
path/msg1.txt;3;c
path/msg2.txt;5;d
path/msg3.txt;12;c
```

2. le fichier message – possède un message composé de mots/chiffres/symboles séparés par des caractères blancs (espaces, sauts de lignes, etc.).

```
Bonjour ! Ici un message simple pour
tester le chiffrement.
```

4 Consignes d'implémentation

Le programme doit être composé de trois groupes d'acteurs :
— le processus directeur – envoi des fichiers messages aux chefs d'équipes ;
— les processus chefs d'équipe – communication entre le directeur et les employés, découpage du message et transmission aux employés ;
— les threads employés – exécution de la tâche.

Le processus directeur devra donc créer n processus chefs d'équipe, 1 par fichier message renseigné dans le fichier principal, puis transmettre le chemin vers le fichier message ainsi que les paramètres liés à son traitement. Le processus chef d'équipe créera à son tour m threads, et donnera à chacun un mot du message. En fonction de l'opération demandée, le résultat sera retourné sous deux formes différentes :

Seuls les lettres sont concernées par le chiffrement/déchiffrement, tous les autres symboles sont conservés tels quels. La casse des lettres (majuscule/minuscule) est également conservée.
— chiffrement – un nouveau fichier message est créé par le chef d'équipe, avec comme chemin, celui du clair auquel est ajouté `_cypher` ;
— déchiffrement – le contenu du message déchiffré est redonné au directeur, qui l'affiche sur la sortie standard.

La gestion des communications entre les différents acteurs est libre, **hormis celle allant des employés aux chefs d'équipe** : une chaîne de caractère 'buffer', allouée par le chef d'équipe, servira à stocker les mots traités par les employés. **Attention** à l'ordre de remplissage du buffer.

Les consignes suivantes doivent être respectées :
- l'appel au programme doit se faire comme suit :
  `./cesar fichier`
- les appels systèmes doivent être utilisés pour les fonctionnalités suivantes :
  — la gestion des fichiers d'entrée/sortie ;
  — la gestion des processus ;
  — la gestion des tubes.
- les constantes sont les seules variables globales autorisées ;
- chaque allocation doit être libérée avant la fin du programme ;
- chaque création de processus/thread doit être 'attendue'.

Rappel : Pour savoir si une fonction est un appel système ou non, il suffit de regarder le numéro d'indexage de cette dernière dans `man`. 2 signifie que c'est un appel système, 3 une fonction issue d'une bibliothèque.