Sistemi Operativi
prova di laboratorio
– 29 luglio 2024 –

Creare un programma **`decryptor.c`** in linguaggio C che accetti invocazioni sulla riga di comando del tipo:

`decryptor <keys-file> <ciphertext-input-file> [plaintext-output-file]`

Il programma dovrà decifrare un testo a cui è stato applicato un "cifrario per sostituzione" utilizzando le chiavi riportate nel file specificato. In un cifrario per sostituzione le lettere dell'alfabeto nel messaggio in chiaro originale vengono sostituite con quelle di un alfabeto in cui le lettere sono permutate in un ordine casuale. A titolo d'esempio la prima delle chiavi fornite, `HWEPOUQXLTZFDVYISKNGRCAMJB`, indica che per decifrare sono necessari i seguenti scambi:

```
H W E P O U Q X L T Z F D V Y I S K N G R C A M J B
Δ Δ Δ Δ Δ Δ Δ Δ Δ Δ Δ Δ Δ Δ Δ Δ Δ Δ Δ Δ Δ Δ Δ Δ
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

Ogni riga del file cifrato ha una struttura del tipo "`3:ur yqur yqf lfpb buif`", dove "`3`", in questo caso, indica l'impiego della terza chiave.

Il file con le chiavi conterrà *n* righe con *n≥2*: ogni riga conterrà esattamente una permutazione delle 26 lettere dell'alfabeto maiuscolo (la "chiave"). Il thread principale dovrà creare *n* thread `K-`*i*: ognuno con il compito di gestire unicamente una delle chiavi fornite passata in una struttura dati privata. Tutti i thread condivideranno una struttura dati (rigorosamente non definita come globale!) con i seguenti elementi:

- un buffer di 100 byte utile a contenere una riga di testo;
- uno o più semafori.

Sempre il thread principale si occuperà di leggere riga-per-riga il file con il ciphertext: la riga da decifrare verrà caricata nel buffer condiviso e il thread `K-`*i* corrispondente verrà risvegliato all'occorrenza. Il generico thread `K-`*i*, quando richiesto, si occuperà di decifrare la riga passata sostituendola con la versione in chiaro. Il thread principale, non appena la riga attuale è stata decifrata, si occuperà di visualizzarla sullo standard output e di scriverla sull'eventuale file di output. Se il file di output viene specificato questo dovrà essere creato/sovrascritto.

I thread dovranno usare unicamente semafori generici della libreria phtread per coordinarsi e terminare spontaneamente alla fine dei lavori. Non si dovranno usare strutture dati definite globalmente. E' necessario rispettare, quanto più fedelmente possibile, la struttura dell'output suggerito nell'esempio a seguire.

Tempo: 2 ore

L'output atteso di una esecuzione sui file di esempio forniti (keys.txt, ciphertext.txt) potrebbe essere il seguente:

```
$ ./decryptor keys.txt ciphertext.txt plaintext.txt

[M] trovate 3 chiavi: creo i thread K-i necessari
[K1] chiave assegnata: HWEPOUQXLTZFDVYISKNGRCAMJB
...
[M] la riga 'ur yqur yqf lfpb buif ?' deve essere decifrat  n la chiave n.3
[K3] sto decifrando la frase di 22 caratteri passa
[M] la riga è stata decifrata in: IS THIS TH
[M] la riga 'iy moiy clym wrkmryg ?'  la chiave n.2
[K2] sto decifrando la frase di 
[M] la riga è stata decifrata in:  NTASY ?
...
```