

PR6 – Programmation réseaux

TP n° 6 : Un client dict

I) Les longs messages

Exercice 1 :

Contrairement aux messages courts échangés dans les TPs précédents, il n'est pas garanti que quand on appelle la fonction `recv`, on reçoive tout le message envoyé par le serveur. Ni qu'on recevra le contenu d'un seul `send`. En pratique, `send` écrit dans la socket, et `recv` lira au plus `size` octets dans la socket.

Il faudra donc stocker les messages reçus dans un tampon, et chercher dans ce tampon le symbole de fin qui nous intéresse.

Écrivez une fonction qui va :

- chercher le symbole de fin voulu
- si ce symbole est dans le tampon, extraire la chaîne correspondante
- sinon, sauvegarder la fin du tampon, puis appeler `recv` pour récupérer la suite des messages (qui devraient contenir le symbole de fin, sinon les tampons sont trop petits).

Suggestion : définir un type structuré pour le tampon, et la tête de lecture, ainsi que les fonctions suivantes :

```
typedef struct t {
    char buf[SIZE_MSG+1];
    int cur;
    int size;
} buf_t;

int find_token(buf_t* buf, char token);
void read_line(int socket_fd, buf_t* buf, buf_t* line_buf);
```

Exercice 2 : Requête de traduction

Modifiez votre programme pour qu'il demande les traductions en français du mot passé par la ligne de commande et les affiche si le mot est dans le dictionnaire, et un message d'erreur sinon.

Indications :

- la base de données qui a la réponse est `fd-eng-fra`;
- la réponse est constituée d'une suite de lignes :
 - 552_ si le mot est absent du dictionnaire,
 - 150_ si le mot est présent puis 151_ indiquant dans quel dictionnaire la définition se trouve puis des lignes de la forme `i_XXX` où `i` est un entier et `XXX` les `i`^e traductions et enfin `.` indiquant la fin de la liste,
 - un entier indiquant le statut de la réponse.

Exercice 3 : Les expressions régulières

La commande `MATCH` permet entre autres d'obtenir une liste de mots correspondants à une expression régulière.

Écrivez un programme qui prend en argument une expression régulière et affiche un message de la forme « Les N correspondances sont x_1, x_2, x_3 .`\n` » ou alors « Aucune correspondance trouvée.`\n` ».