



Projet Programmation et Système

Thème : Jeu de devinette d'un mot

Réalisé par :

- Fatima-Ezzahra AKHTEN
- Imene OUAHRANI

Présentation générale:

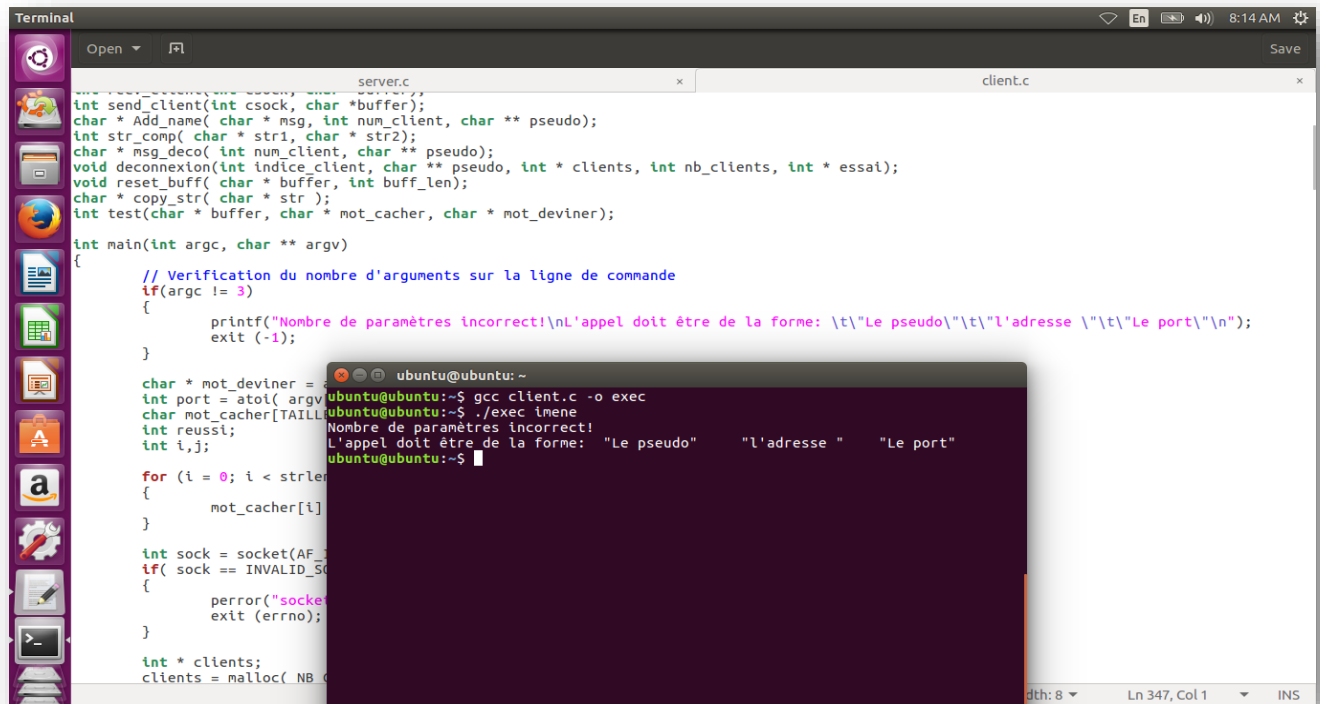
Nous avons choisi comme thème: **un jeu de devinette** dont le concept consiste à proposer des lettres (par les clients connectés) pour pouvoir deviner le mot lancé par le serveur (multi-clients).

Les clients (ou les joueurs) doivent lancer leurs propositions lettre par lettre (et non pas un caractère) sans dépasser le nombre de tentatives donné.

Le client qui devine la dernière lettre du mot caché en respectant le nombre d'essais, sera le gagnant. Une fois il épuise ses tentatives, ses propositions seront ignorées par le serveur.

1.Client

- L'exécutable doit accepter les paramètres suivant : **le pseudo/l'adresse du serveur/le port.**
- Pour nous c'est au niveau local:
Exemple : **./exec JEAN 127.0.0.1 5000**
- Un message d'erreur s'affiche si les paramètres ne sont pas respectés.



```
server.c
client.c

int send_client(int csock, char *buffer);
char * Add_name( char * msg, int num_client, char ** pseudo);
int str_comp( char * str1, char * str2);
char * msg_deco( int num_client, char ** pseudo);
void deconnexion(int indice_client, char ** pseudo, int * clients, int nb_clients, int * essai);
void reset_buff( char * buffer, int buff_len);
char * copy_str( char * str );
int test(char * buffer, char * mot_cacher, char * mot_deviner);

int main(int argc, char ** argv)
{
    // Verification du nombre d'arguments sur la ligne de commande
    if(argc != 3)
    {
        printf("Nombre de paramètres incorrect!\nL'appel doit être de la forme: \t\"Le pseudo\" \t\"l'adresse\" \t\"Le port\"\n");
        exit (-1);
    }

    char * mot_deviner = argv[2];
    int port = atoi( argv[3] );
    char mot_cacher[TAILL];
    int reussi;
    int i,j;

    for (i = 0; i < strlen(mot_deviner); i++)
    {
        mot_cacher[i] = '*';
    }

    int sock = socket(AF_INET, SOCK_STREAM, 0);
    if( sock == INVALID_SOCKET )
    {
        perror("socket");
        exit (errno);
    }

    int * clients;
    clients = malloc( NB_CLIENTS * sizeof(int));
}
```

```
ubuntu@ubuntu:~$ gcc client.c -o exec
ubuntu@ubuntu:~$ ./exec imene
Nombre de paramètres incorrect!
L'appel doit être de la forme: "Le pseudo" "l'adresse " "Le port"
ubuntu@ubuntu:~$
```

- La connexion du client s'effectue après l'exécution du serveur. Après, le jeu commence.

2.Serveur

- L'exécutable doit accepter : **mot à deviner/port**
Exemple : **./exec vœux 5000**
- Un message d'erreur s'affiche si les paramètres ne sont pas respectés.

```

server.c
client.c

int send_client(int csock, char *buffer);
char * Add_name( char * msg, int num_client, char ** pseudo);
int str_comp( char * str1, char * str2);
char * msg_deco( int num_client, char ** pseudo);
void deconnexion(int indice_client, char ** pseudo, int * clients, int nb_clients, int * essai);
void reset_buff( char * buffer, int buff_len);
char * copy_str( char * str );
int test(char * buffer, char * mot_cacher, char * mot_deviner);

int main(int argc, char ** argv)
{
    // Verification du nombre d'arguments sur la ligne de commande
    if(argc != 3)
    {
        printf("Nombre de paramètres incorrect!\nL'appel doit être de la forme: \"mot à deviner\" \"port du serveur\" \n");
        exit (-1);
    }

    char * mot_deviner = argv[1];
    int port = atoi( argv[2]);
    char mot_cacher[TAILL];
    int reussi;
    int i,j;

    for (i = 0; i < strlen(mot_deviner); i++)
    {
        mot_cacher[i] = '*';
    }

    int sock = socket(AF_INET, SOCK_STREAM, 0);
    if( sock == INVALID_SOCKET)
    {
        perror("socket");
        exit (errno);
    }

    int * clients;
    clients = malloc( NB_CLIENTS * sizeof(int));

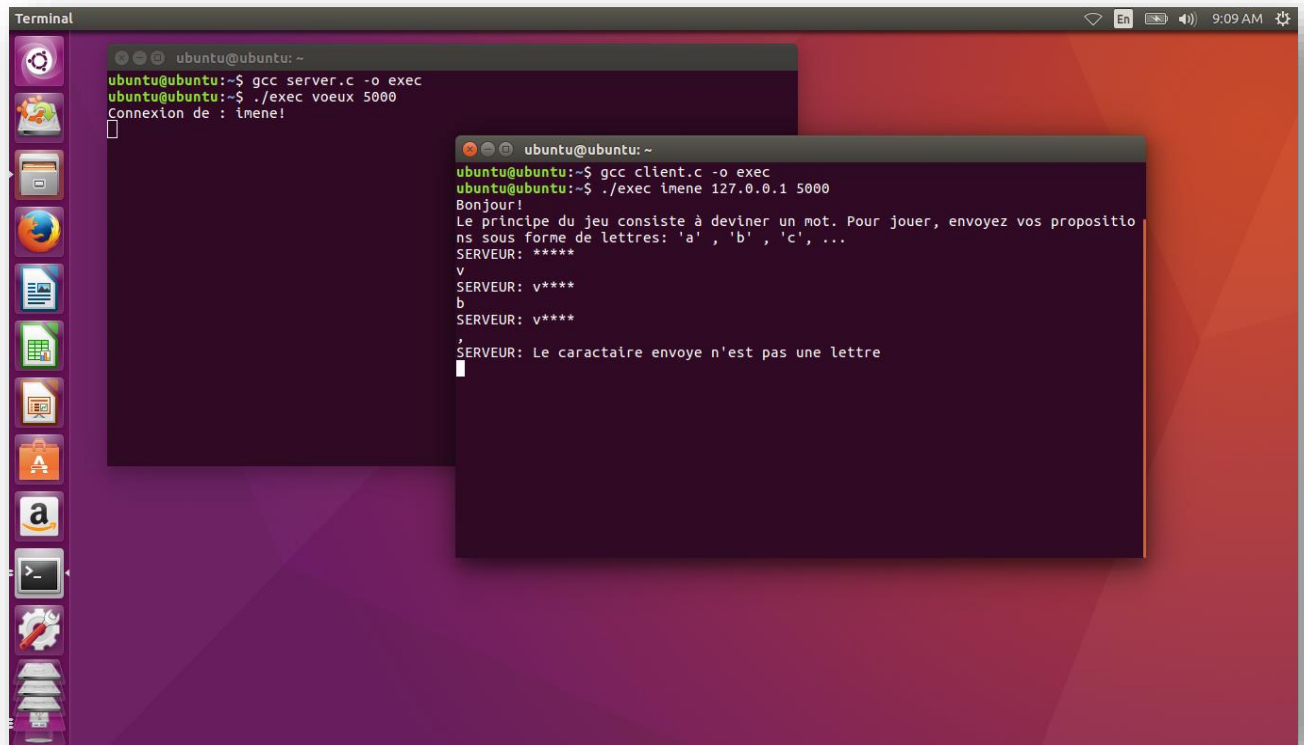
```

```

ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ gedit server.c
ubuntu@ubuntu:~$ gcc server.c -o exec
ubuntu@ubuntu:~$ ./exec voeux
Nombre de paramètres incorrect!
L'appel doit être de la forme: "mot à deviner" "port du serveur"
ubuntu@ubuntu:~$

```

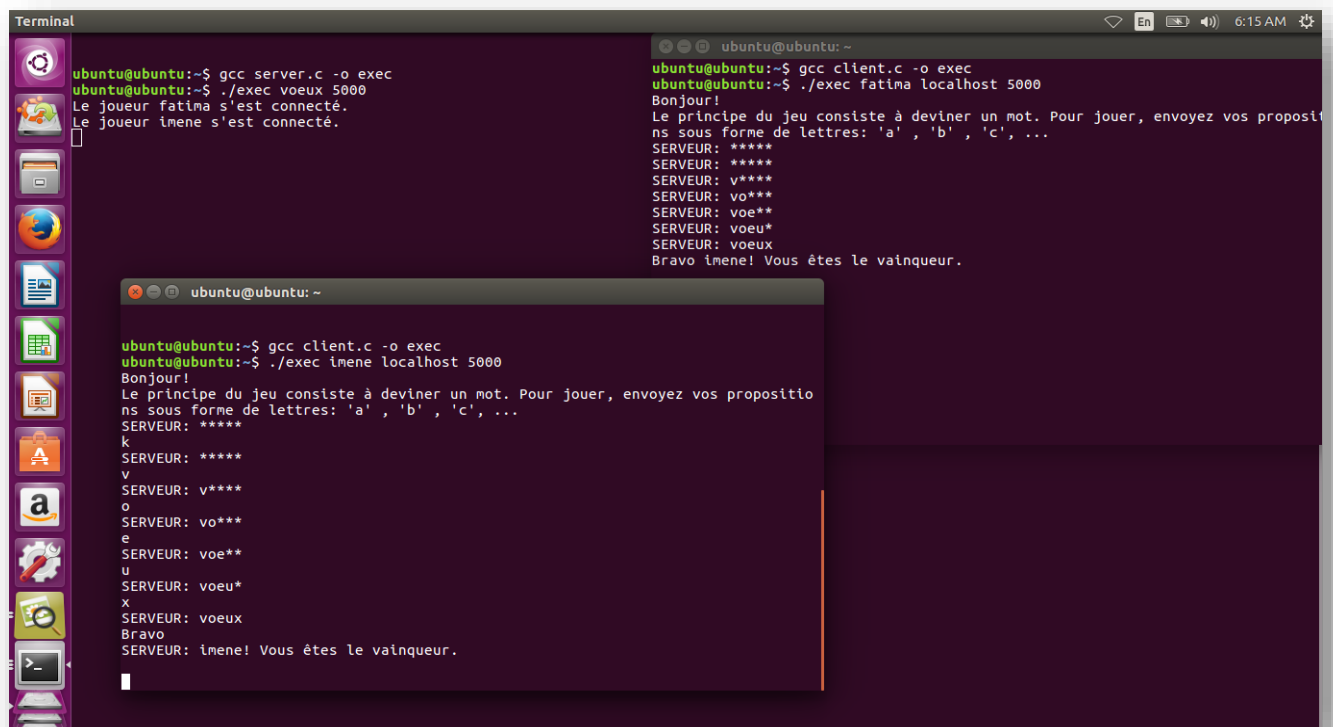
- La connexion d'un nouveau client est indiquée sur le terminal du serveur.
- Quand un client se connecte, le serveur lui envoie un message :
« Bonjour ! »
Le principe du jeu consiste à deviner un mot. Pour jouer, envoyez vos propositions sous forme de lettres : 'a', 'b', 'c',... »
- Quand le serveur se déconnecte, les clients seront informés par un message du type:
« Le serveur s'est déconnecté! »
- Lorsqu'un client dépasse le nombre de tentatives, le serveur lui envoie un message:
« Oups ! *Pseudo du client* a épuisé ces coups » ainsi pour les autres clients participants.



```
Terminal
ubuntu@ubuntu:~$ gcc server.c -o exec
ubuntu@ubuntu:~$ ./exec voeux 5000
Connexion de : imene!

ubuntu@ubuntu:~$ gcc client.c -o exec
ubuntu@ubuntu:~$ ./exec imene 127.0.0.1 5000
Bonjour!
Le principe du jeu consiste à deviner un mot. Pour jouer, envoyez vos propositions sous forme de lettres: 'a', 'b', 'c', ...
SERVEUR: *****
v
SERVEUR: v****
b
SERVEUR: b****
,
SERVEUR: Le caractere envoye n'est pas une lettre
```

- Si le nombre de tentatives est respecté et le mot à deviner est trouvé, le serveur envoie un message aux clients :
« **Bravo *pseudo du client* ! Vous êtes le vainqueur** »



```
Terminal
ubuntu@ubuntu:~$ gcc server.c -o exec
ubuntu@ubuntu:~$ ./exec voeux 5000
Le joueur fatima s'est connecté.
Le joueur imene s'est connecté.

ubuntu@ubuntu:~$ gcc client.c -o exec
ubuntu@ubuntu:~$ ./exec fatima localhost 5000
Bonjour!
Le principe du jeu consiste à deviner un mot. Pour jouer, envoyez vos propositions sous forme de lettres: 'a', 'b', 'c', ...
SERVEUR: *****
SERVEUR: *****
SERVEUR: v****
SERVEUR: vo***
SERVEUR: voe**
SERVEUR: voeu*
SERVEUR: voeux
Bravo imene! Vous êtes le vainqueur.

ubuntu@ubuntu:~$ gcc client.c -o exec
ubuntu@ubuntu:~$ ./exec imene localhost 5000
Bonjour!
Le principe du jeu consiste à deviner un mot. Pour jouer, envoyez vos propositions sous forme de lettres: 'a', 'b', 'c', ...
SERVEUR: *****
k
SERVEUR: *****
v
SERVEUR: v*****
o
SERVEUR: o*****
e
SERVEUR: e*****
u
SERVEUR: u*****
x
SERVEUR: x*****
Bravo
SERVEUR: Bravo imene! Vous êtes le vainqueur.
```

Quand le serveur se déconnecte, un message sera affiché au client : « **Le serveur s'est déconnecté** »

```
ubuntu@ubuntu: ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
ubuntu@ubuntu:~$ clear  
ubuntu@ubuntu:~$ gcc client.c -o exec  
ubuntu@ubuntu:~$ ./exec fatima 127.0.0.1 3000  
Bonjour!  
Le principe du jeu consiste à deviner un mot. Pour jouer, envoyez vos propositions sous forme de lettres: 'a' , 'b' , 'c' , ...  
SERVEUR: *****  
v  
SERVEUR: v****  
o  
SERVEUR: vo***  
e  
SERVEUR: voe**  
u  
SERVEUR: voeu*  
x  
SERVEUR: voeuxBravo fatima  
SERVEUR: ! Vous êtes le vainqueur.  
Le serveur s'est déconnecté.  
ubuntu@ubuntu:~$
```

Voici le résumé de l'exécution

```
ubuntu@ubuntu: ~  
Bonjour!  
Le principe du jeu consiste à deviner un mot. Pour jouer, envoyez vos propositions sous forme de lettres: 'a' , 'b' , 'c' , ...  
SERVEUR: *****  
v  
SERVEUR: v****  
o  
SERVEUR: vo***  
e  
SERVEUR: voe**  
u  
SERVEUR: voeu*  
x  
SERVEUR: voeuxBravo imene! Vous êtes le vainqueur.  
Le serveur s'est déconnecté.  
ubuntu@ubuntu:~$ clear  
ubuntu@ubuntu:~$ gcc client.c -o exec  
ubuntu@ubuntu:~$ ./exec imene 127.0.0.1 2018  
Bonjour!  
Le principe du jeu consiste à deviner un mot. Pour jouer, envoyez vos propositions sous forme de lettres: 'a' , 'b' , 'c' , ...  
SERVEUR: *****  
v  
SERVEUR: v****  
o  
SERVEUR: vo***  
k  
SERVEUR: vo***  
p  
SERVEUR: vo***  
l  
SERVEUR: vo***  
;  
SERVEUR: Le caractère envoyé n'est pas une lettre.  
g  
SERVEUR: vo***  
y  
SERVEUR: vo***  
t  
SERVEUR: vo***  
SERVEUR:  
Oops! imene  
SERVEUR: a épuisé ces coups.
```