

TD 3 – Programmation Socket en C

1 – Familiarisation avec la programmation

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <sys/types.h>

#define PORT 12345
int sock, socket2, lg;
char mess[80];
struct sockaddr_in local; // ???
struct sockaddr_in distant; // ???

creer_socket()
{
    // preparation des champs d'entete
    bzero(&local, sizeof(local)); // ???
    local.sin_family = AF_INET; // ???
    local.sin_port = htons(PORT); // ???
    local.sin_addr.s_addr = INADDR_ANY; // ???
    bzero(&(local.sin_zero), 8); // ???

    lg = sizeof(struct sockaddr_in);
    // ???
    if((sock=socket(AF_INET, SOCK_STREAM, 0)) == -1){perror("socket"); exit(1);}
}

main()
{
    // ???
    creer_socket();
    // ???
    if(bind(sock, (struct sockaddr *)&local, sizeof(struct sockaddr)) == -1)
        {perror("bind");exit(1);}
    // ???
    if(listen(sock, 5) == -1){perror("listen");exit(1);}

    // ???
    while(1)
    { // ???
        if((socket2=accept(sock, (struct sockaddr *)&distant, &lg)) == -1)
            {perror("accept");exit(1);} // ???
        printf ("client connecte \n");
        read(socket2,mess,80); // ???
        printf ("le client me dit %s \n",mess);
        close(socket2); // ???
    }
}
```

Découverte

Commentez les parties notées ???

Test en local:

Saisir, compilez et exécutez le programme ci-dessus.

Ouvrez une autre console et lancez : **telnet 127.0.0.1 12345**. Que se passe-t-il ?

2 – Familiarisation avec la programmation

Complétez le programme ci-dessous pour qu'il puisse communiquer avec le programme précédent.

```
#include <stdio.h>
#include <netdb.h>
#include <fcntl.h>
#include <string.h>
#include <sys/socket.h>
#include <netinet/in.h> // gestion adresses i
#include <sys/types.h>

#define SERV ???          // adresse IP = boucle locale
#define PORT ???          // port d'ecoute serveur
int port, sock;           // n°port et socket
char mess[80];            // chaine de caracteres

struct sockaddr_in serv_addr; // zone adresse
struct hostent *server;       // nom serveur

creer_socket()
{
    port = PORT;
    server = gethostbyname(SERV); // verification existence adresse
    if (!server){fprintf(stderr, "Problème serveur \"%s\"\n", SERV);exit(1);}
    // creation socket locale
    ??? // creation socket
    bzero(&serv_addr, sizeof(serv_addr)); // preparation champs entete
    serv_addr.sin_family = AF_INET; // Type d'adresses
    bcopy(server->h_addr, &serv_addr.sin_addr.s_addr, server->h_length);
    serv_addr.sin_port = htons(port); // port de connexion du serveur
}

main()
{
    // creation socket
    creer_socket();
    // connexion au serveur
    ??? // connexion à l'application du dessus
    ??? // saisie d'une chaine de caractères au clavier
    ??? // envoi de cette chaine à l'application du dessus
    printf ("connexion avec serveur ok\n");
    close (sock);
}
```

Si tout fonctionne, BRAVO, vous venez de créer votre première application Client/Serveur !!!
Mais au fait qui est client, qui est le serveur ? Justifiez votre réponse .

3 – Création d'un petit "chat"

Apporter les modifications nécessaires aux deux programmes précédents afin de mettre en oeuvre un dialogue.

4 – Du multi-tâches

Comment modifier l'application "serveur", pour que ce dernier puisse répondre à plusieurs clients en même temps ?

Faites la modification et testez.

Remarque : Pour cela vous pouvez utiliser l'application cliente ou telnet.