

1 Exemple de client

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

void error(char *msg) {
    perror(msg);
    exit(0);
}

int main(int argc, char *argv[]) {
    int portno;
    int sockfd;
    struct sockaddr_in serv_addr;
    struct hostent *server;
    char *msg="Message d'exemple";

    portno = 7;
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");
    server = gethostbyname("localhost");
    if (server == NULL)
        error("ERROR, no such host");

    serv_addr.sin_family = AF_INET;
    memcpy(&serv_addr.sin_addr, server->h_addr_list[0], server->h_length);
    serv_addr.sin_port = htons(portno);

    if (connect(sockfd,(struct sockaddr*)&serv_addr,sizeof(serv_addr)) < 0)
        error("ERROR connecting");

    char *buffer=malloc(sizeof(char)*(strlen(msg)+1));

    write(sockfd,msg,strlen(msg));
    read(sockfd,buffer,strlen(msg));

    // we receive an array of characters, so it must be converted
    // to a string by adding a \0 at the end
    buffer[strlen(msg)]='\0';
    printf("%s\n", buffer);

    return 0;
}
```

2 Exemple de server

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

void error(char *msg) {
    perror(msg);
    exit(1);
}

int main(int argc, char *argv[]) {
    int sockfd, newsockfd, portno, clilen;
    char buffer;
    struct sockaddr_in serv_addr, cli_addr;
    int n;

    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");

    portno = 7;
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_addr.s_addr = INADDR_ANY;
    serv_addr.sin_port = htons(portno);
    if (bind(sockfd, (struct sockaddr *) &serv_addr,
        sizeof(serv_addr)) < 0)
        error("ERROR on binding");
    listen(sockfd,5);
    clilen = sizeof(cli_addr);

    newsockfd = accept(sockfd,
        (struct sockaddr *) &cli_addr,
        (socklen_t*) &clilen);
    if (newsockfd < 0)
        error("ERROR on accept");

    while( n = read(newsockfd,&buffer,1))
        write(newsockfd,&buffer, 1);

    return 0;
}
```

3 Exercice: Partie 1

- Lire et comprendre les deux exemples
- Lancer le client en utilisant le service *Echo* de `marine.edu.ups-tlse.fr`
- Modifier le client et le serveur pour prendre un port/nom de serveur en ligne de commande

- Modifier le client pour qu'il accepte une chaîne au clavier (**man scanf**)
- Modifier le serveur pour pouvoir supporter plusieurs clients successifs
- Modifier le serveur pour qu'il puisse lire plus de 1 caractère en même temps.

4 Exercice: Partie 2

On veut créer un service qui donne le nombre de **a** dans des lignes

- réutilisez la partie 1 pour renvoyer le résultat sous la forme d'une chaîne de caractères
- modifier les codes pour que la valeur soit renvoyée en tant qu'entier.