

## TD5 Client/serveur UDP

### M1 IRCOMS

#### Exercice 1 :

- 1- Compléter le code suivant pour réaliser une communication client serveur avec le protocole UDP (mode non connecté).

#### client.c

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

int main()
{
    int sock;
    struct sockaddr_in server_addr;
    struct hostent *host;
    char send_data[1024];

    host= ...

    if ((sock =...) == -1)
    {
        perror("socket");
        exit(1);
    }

    server_addr.sin_family = ...
    server_addr.sin_port = ...
    server_addr.sin_addr =...
    bzero(&(server_addr.sin_zero),8);

    while (1)
    {
        printf("Type Something (q or Q to quit):");
        gets(send_data);

        if ((strcmp(send_data , "q") == 0) || strcmp(send_data , "Q") == 0)
            break;

        else
            ...
    }
}
```

```
}
}
```

### server.c

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

int main()
{
    int sock;
    int addr_len, bytes_read;
    char recv_data[1024];
    struct hostent *host;
    struct sockaddr_in server_addr , client_addr;

    if ((sock = ...) == -1) {
        perror("Socket");
        exit(1);
    }

    server_addr.sin_family = ...
    server_addr.sin_port = ...
    server_addr.sin_addr.s_addr = ...
    bzero(&(server_addr.sin_zero),8);

    if (...== -1)
    {
        perror("Bind");
        exit(1);
    }

    addr_len = sizeof(struct sockaddr);

    printf("\nUDPServer Waiting for client on port 5000");
    fflush(stdout);

    while (1)
    {
        bytes_read = ...

        recv_data[bytes_read] = '\0';

        printf("\n(%s , %d) said : ",inet_ntoa(client_addr.sin_addr),
                ntohs(client_addr.sin_port));
        printf("%s", recv_data);
        fflush(stdout);
    }
}
```

```
}  
    }  
    return 0;  
}
```