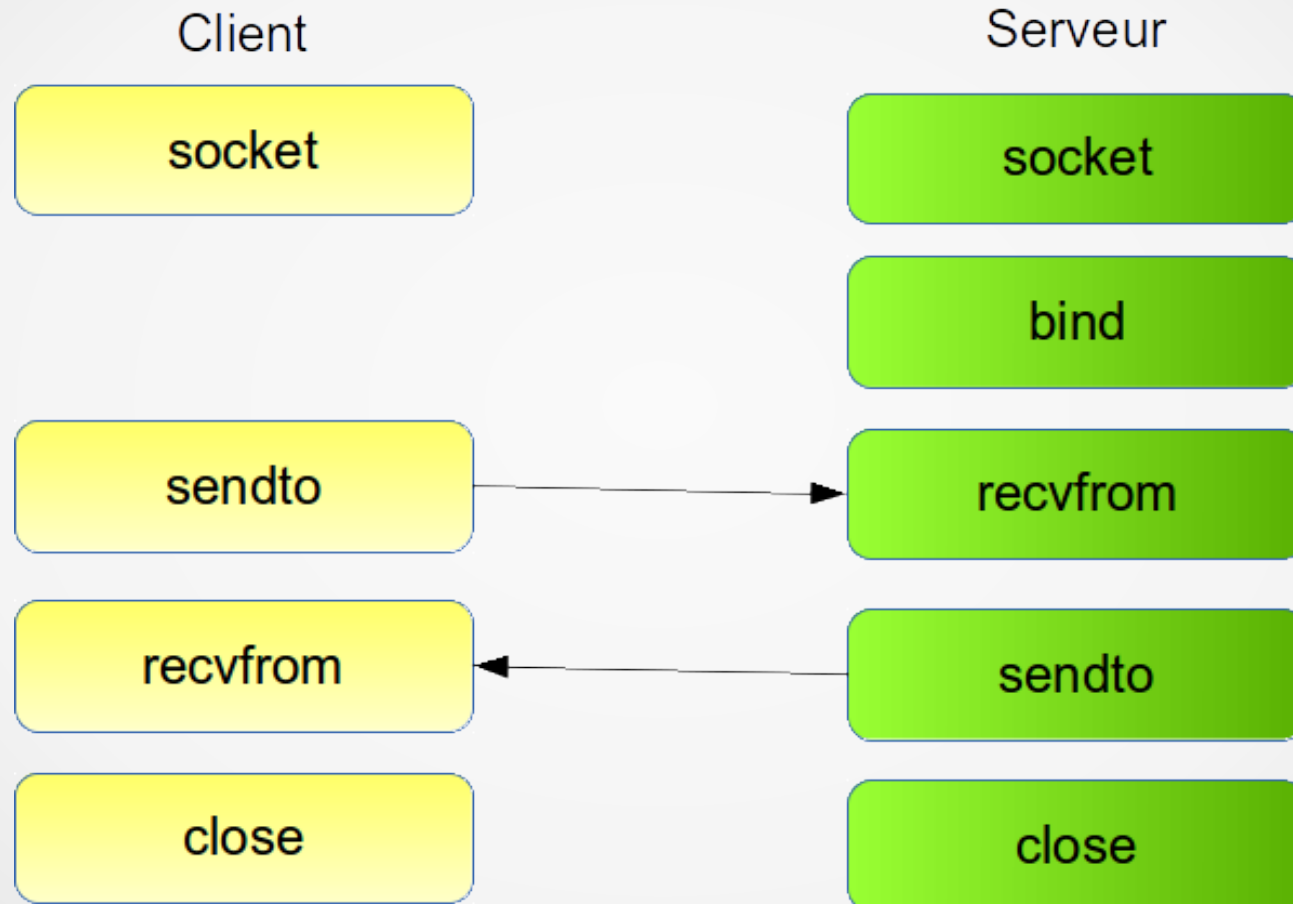




# Synthèse Socket Serveur UDP en C

# Mode UDP (non-connecté)



# Serveur UDP

```
int fdSocket;
struct sockaddr_in infoServeur;
struct sockaddr_in infoClient;
int retour;
int tailleClient;
int valRec;
fdSocket = socket(PF_INET, SOCK_DGRAM, IPPROTO_UDP);
if (fdSocket == -1)
{
    printf("pb socket : %s\n", strerror(errno));
    exit(errno);
}

infoServeur.sin_family = AF_INET;
infoServeur.sin_port = htons(2222);
infoServeur.sin_addr.s_addr = htonl(INADDR_ANY); //IP du serveur dans l'ordre des octets du reseau

retour = bind(fdSocket, (struct sockaddr*) &infoServeur, sizeof(infoServeur));
if (retour == -1)
{
    printf("pb bind : %s\n", strerror(errno));
    exit(errno);
}

tailleClient = sizeof(infoClient);
```

```
int bind(int sockfd, const struct sockaddr *addr, socklen_t addrlen);
```

# Serveur UDP

```
int fdSocket;
struct sockaddr_in infoServeur;
struct sockaddr_in infoClient;
int retour;
int tailleClient;
int valRec;
fdSocket = socket(PF_INET, SOCK_DGRAM, IPPROTO_UDP);
...
tailleClient = sizeof(infoClient);
while (1==1){

    retour = recvfrom(fdSocket, &valRec, sizeof(valRec), 0,
                     (struct sockaddr *) &infoClient, &tailleClient);
    if (retour == -1)
    {
        printf("pb recvfrom : %s\n", strerror(errno));
        exit(errno);
    }
    printf("message du client %s/%d: %d\n", inet_ntoa(infoClient.sin_addr),
                                             infoClient.sin_port, valRec);
    valRec=-valRec;
    retour = sendto(fdSocket, &valRec, sizeof(valRec), 0,
                   (struct sockaddr *) &infoClient, sizeof(infoClient));
    if (retour == -1)
    {
        printf("pb sendto : %s\n", strerror(errno));
        exit(0);
    }
}
```

# Mode UDP

- A retenir :
  - Les informations concernant celui à qui l'on envoie doivent être passées en paramètre de la fonction **sendto**.
  - Les informations concernant celui qui a envoyé sont mises à jour via la fonction **recvfrom**.
  - Dans les deux cas, il s'agit d'une variable de type :
    - **struct sockaddr\_in**
  - Attention à la fonction **recvfrom** qui a des passages de paramètres par adresse.