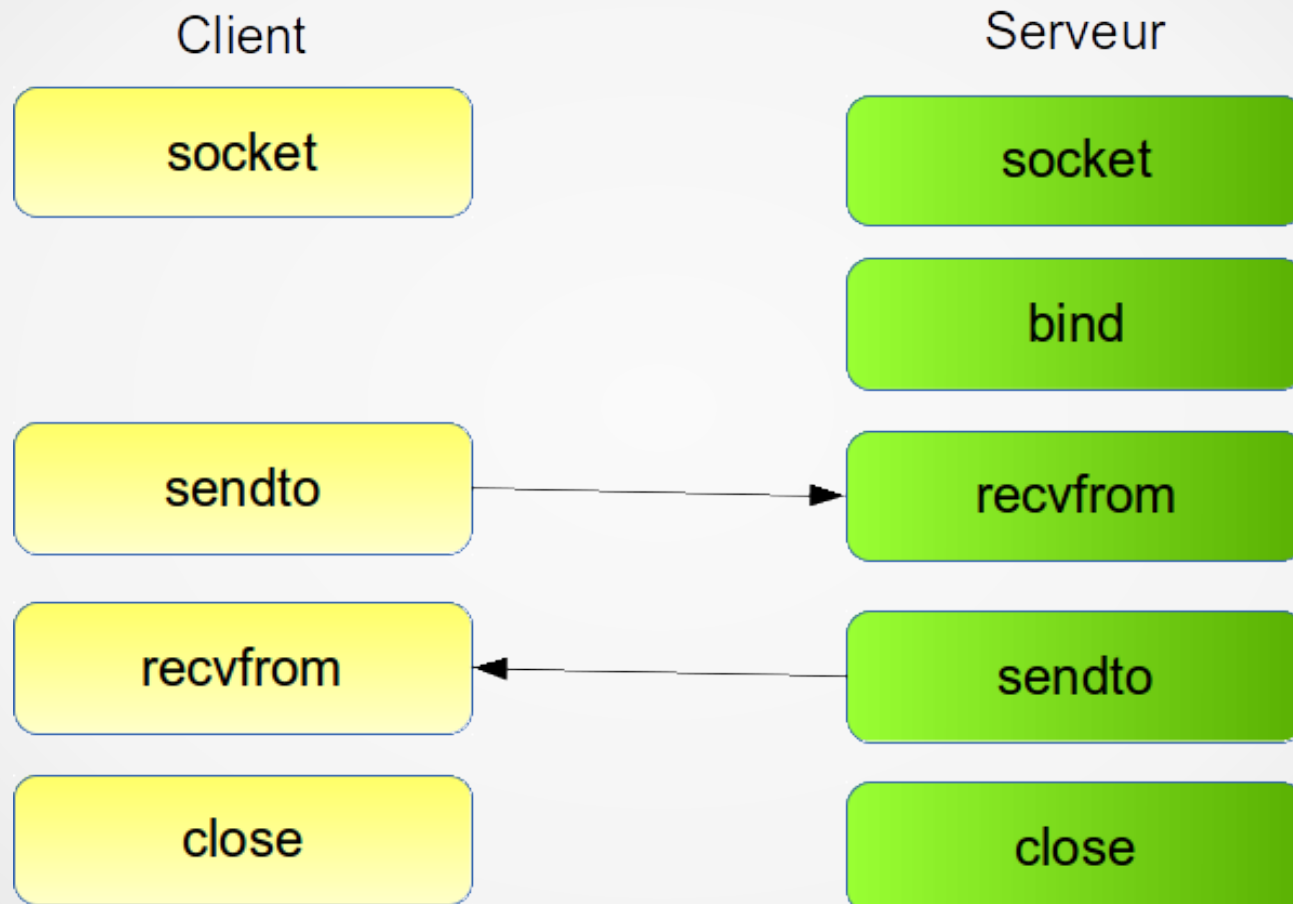




# Synthèse Socket en C

## Client UDP

# Mode UDP (non-connecté)



# Client UDP

```
#include <errno.h>
#include <string.h>
```

```
int socketUdp ;
socketUdp = socket(PF_INET, SOCK_DGRAM, IPPROTO_UDP);
if (socketUdp == -1)
{
    printf("pb socket : %s\n", strerror(errno));
    exit(errno);
}
```

```
#include <sys/types.h>
#include <sys/socket.h>
```

# Client UDP

La socket

Adresse mémoire  
de la donnée  
à envoyer

Taille de la  
donnée à envoyer

```
ssize_t sendto(int sockfd, const void *buf, size_t len,  
int flags, const struct sockaddr *dest_addr,  
socklen_t addrlen);
```

0

Taille de la structure  
struct sockaddr\_in

Adresse mémoire d'une variable  
de type **struct sockaddr\_in**  
contenant les données  
destinées au destinataire :

**Si le type attendu dans le prototype est un pointeur,  
il ne faut pas pour autant en déclarer un.  
Donner l'adresse d'une variable statique est bien plus judicieux !**

# Client UDP

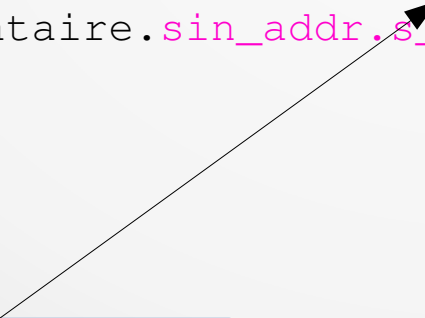
```
struct sockaddr_in {
    sa_family_t    sin_family; // famille d'adresses : AF_INET
    uint16_t       sin_port;   // port dans l'ordre des octets réseau
    struct in_addr sin_addr;   // adresse Internet
};

struct in_addr {
    uint32_t       s_addr;     // adresse dans l'ordre des octets réseau
};
```

```
int maDonnee ;
struct sockaddr_in informationsDestinataire ;
int tailleStructure ;
int retourSend ;
int socketUdp ;
socketUdp = socket(PF_INET, SOCK_DGRAM, IPPROTO_UDP);
...
```

```
informationsDestinataire.sin_family = AF_INET ;
informationsDestinataire.sin_port = htons(2222) ;
informationsDestinataire.sin_addr.s_addr = inet_addr("172.17.83.110") ;
```

`#include <arpa/inet.h>`



`#include <netinet/in.h>`



# Client UDP

```
struct sockaddr_in {
    sa_family_t    sin_family; // famille d'adresses : AF_INET
    uint16_t       sin_port;   // port dans l'ordre des octets réseau
    struct in_addr sin_addr;   // adresse Internet
};

struct in_addr {
    uint32_t       s_addr;     // adresse dans l'ordre des octets réseau
};
```

```
int maDonnee ;
struct sockaddr_in informationsDestinataire ;
int tailleStructure ;
int retourSend ;
int socketUdp ;
socketUdp = socket(PF_INET, SOCK_DGRAM, IPPROTO_UDP);
...
```

```
informationsDestinataire.sin_family = AF_INET ;
informationsDestinataire.sin_port = htons(2222) ;
informationsDestinataire.sin_addr.s_addr = inet_addr("172.17.83.110") ;
maDonnee = 32 ;
tailleStructure = sizeof(informationsDestinataire) ;
```

# Client UDP

```
struct sockaddr_in {
    sa_family_t    sin_family; // famille d'adresses : AF_INET
    uint16_t       sin_port;   // port dans l'ordre des octets réseau
    struct in_addr sin_addr;   // adresse Internet
};

struct in_addr {
    uint32_t       s_addr;     // adresse dans l'ordre des octets réseau
};
```

```
int maDonnee ;
struct sockaddr_in informationsDestinataire ;
int tailleStructure ;
int retourSend ;
int socketUdp ;
socketUdp = socket(PF_INET, SOCK_DGRAM, IPPROTO_UDP);
...

informationsDestinataire.sin_family = AF_INET ;
informationsDestinataire.sin_port = htons(2222) ;
informationsDestinataire.sin_addr.s_addr = inet_addr("172.17.83.110") ;
maDonnee = 32 ;
tailleStructure = sizeof(informationsDestinataire) ;
retourSend = sendto(socketUdp, &maDonnee, sizeof(maDonnee), 0,
(struct sockaddr *)&informationsDestinataire, tailleStructure );
if (retourSend == -1)
{
    printf("pb sendto : %s\n", strerror(errno));
    exit(errno);
}
```

# Client UDP

Adresse mémoire  
de la variable qui  
prendra la valeur  
de la donnée reçu

Taille du type de  
la donnée à recevoir

La socket

```
ssize_t recvfrom(int sockfd, void *buf, size_t len,  
int flags, struct sockaddr *src_addr,  
socklen_t *addrlen);
```

0

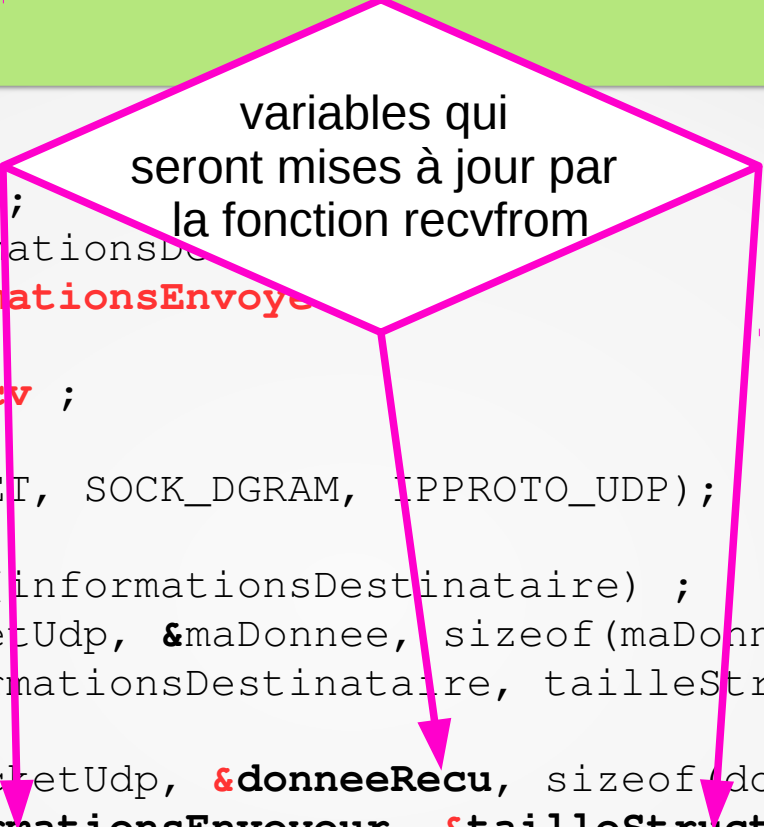
Adresse mémoire d'une  
variable ayant été  
initialisée avec la  
taille de la structure  
`struct sockaddr_in`

Adresse mémoire d'une variable  
de type **struct sockaddr\_in**  
qui sera mise à jour avec les  
données de celui qui a envoyé  
les données, par la fonction **recvfrom**



# Client UDP

variables qui  
seront mises à jour par  
la fonction recvfrom



```
int maDonnee, donneeRecu ;  
struct sockaddr_in informationsDestinataire ;  
struct sockaddr_in informationsEnvoyeur ;  
int tailleStructure ;  
int retourSend, retourRecv ;  
int socketUdp ;  
socketUdp = socket (PF_INET, SOCK_DGRAM, IPPROTO_UDP) ;  
...  
tailleStructure = sizeof (informationsDestinataire) ;  
retourSend = sendto (socketUdp, &maDonnee, sizeof (maDonnee), 0,  
(struct sockaddr *) &informationsDestinataire, tailleStructure) ;  
...  
retourRecv = recvfrom (socketUdp, &donneeRecu, sizeof (donneeRecu), 0,  
(struct sockaddr *) &informationsEnvoyeur, &tailleStructure) ;  
if (retourRecv == -1)  
{  
    printf ("pb recvfrom : %s\n", strerror (errno)) ;  
    exit (errno) ;  
}  
  
printf ("donnee recu : %d \n", donneeRecu) ;  
close (socketUdp) ;
```