

SOCKETS

I. Sockets en C

1. Création du socket

<code>socket(domain, type, protocole)</code>	Création du socket
<code>bind(socket, adresse, lgAdr)</code>	Défini adresse et port local (côté serveur)

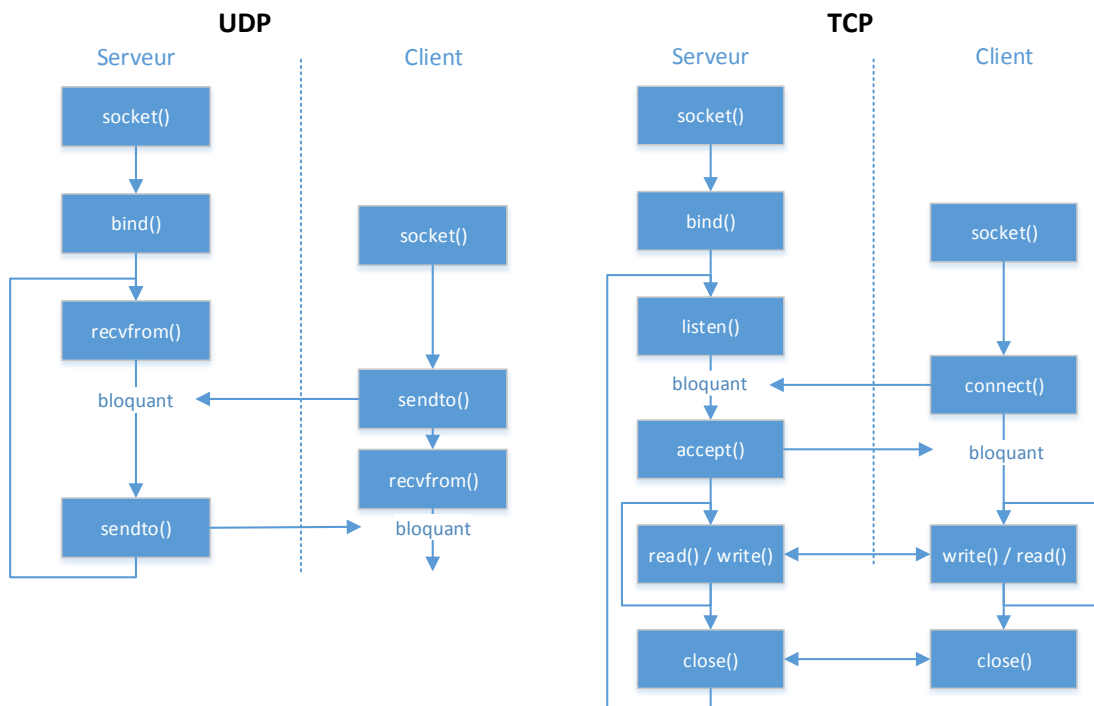
2. Envoi du message en UDP

<code>sendto(socket, message, long, flags, adresseDest, lgAdr)</code>	Envoi de message
<code>recvfrom(socket, *buffer, lgMax, flags, *adresseSrc, *lgAdr)</code>	Reception de message

3. Envoi de message en TCP

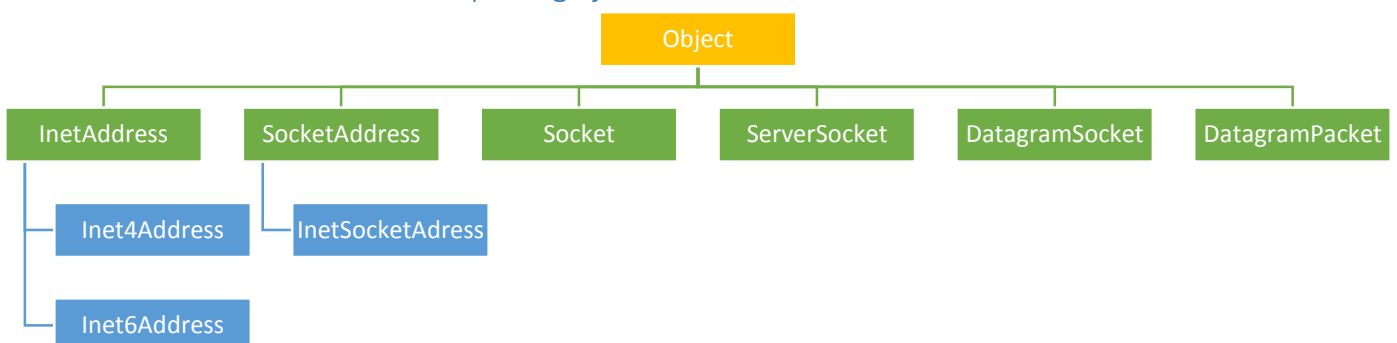
<code>connect(socket, adresse, lgAdr)</code>	Connexion client au serveur
<code>listen(socket, lgFileAttente)</code>	Ecoute les connexion serveur
<code>accept(socket, * adresseSrc, *lgAdr)</code>	Accepte une connexion serveur
<code>read(socket, *buffer, lgMax)</code>	Lit un message
<code>write(socket, message, long)</code>	Envoie un message
<code>close(socket)</code>	Ferme la connexion

4. Schéma des appels de fonctions



II. Sockets en Java

1. Arbre de classes du package java.net



SOCKETS

InfoRep – TDM1

2. InetAddress

- static InetAddress **getLocalHost()**
- static InetAddress **getByName(String host)**
- static InetAddress[] **getAllByName(String host)**

3. Socket : connexion TCP active entre un client et un serveur

- **Socket(String host, int port)** // connexion
- **Socket(InetAddress address, int port)** // connexion
- *InetAddress* **getInetAddress()** // adresse IP distante
- *InetAddress* **getLocalAddress()** // adresse IP locale
- *int* **getPort()** // port distant
- *int* **getLocalPort()** // port local
- *InputStream* **getInputStream()**
- *OutputStream* **getOutputStream()**
- *void* **close()**

4. SocketServer : ouverture d'un port côté serveur en attente de client

- **ServerSocket(int port, int nbClientMax, InetAddress adresse)**
- *Socket* **accept()** // récupérer le socket de communication avec client

5. DatagramSocket : socket de communication sans connexion

- **DatagramSocket()**
- **DatagramSocket(int port)**
- *void* **close()**
- *InetAddress* **getInetAddress()**
- *int* **getPort()**
- *InetAddress* **getLocalAddress()**
- *int* **getLocalPort()**
- *void* **send(DatagramPacket p)**
- *void* **receive(DatagramPacket p)**

6. DatagramPacket : packet sans connexion à envoyer ou à recevoir

- **DatagramPacket(byte[] buf, int length)** // packet reçu
- **DatagramPacket(byte[] buf, int length, InetAddress addr, int port)** // envoyé
- *InetAddress* **getAddress()**
- *byte[]* **getData()**
- *int* **getLength()**
- *int* **getPort()**
- *void* **setAddress(InetAddress iaddr)**
- *void* **setData(byte[] buf)**
- *void* **setLength(int length)**
- *void* **setPort(int iport)**

III. Fonctions utiles en C

sscanf(calcul,"%d%c%d", &e1, &operation, &e2); // scan de la chaîne "calcul" pour en extraire infos.

sprintf(message, "%d", resultat); // écriture dans une chaîne