I. Sockets en C

1. Création du socket

`socket(domaine, type, protocole)`	Création du socket
`bind(socket, adresse, lgAdr)`	Défini adresse et port local (côté serveur)

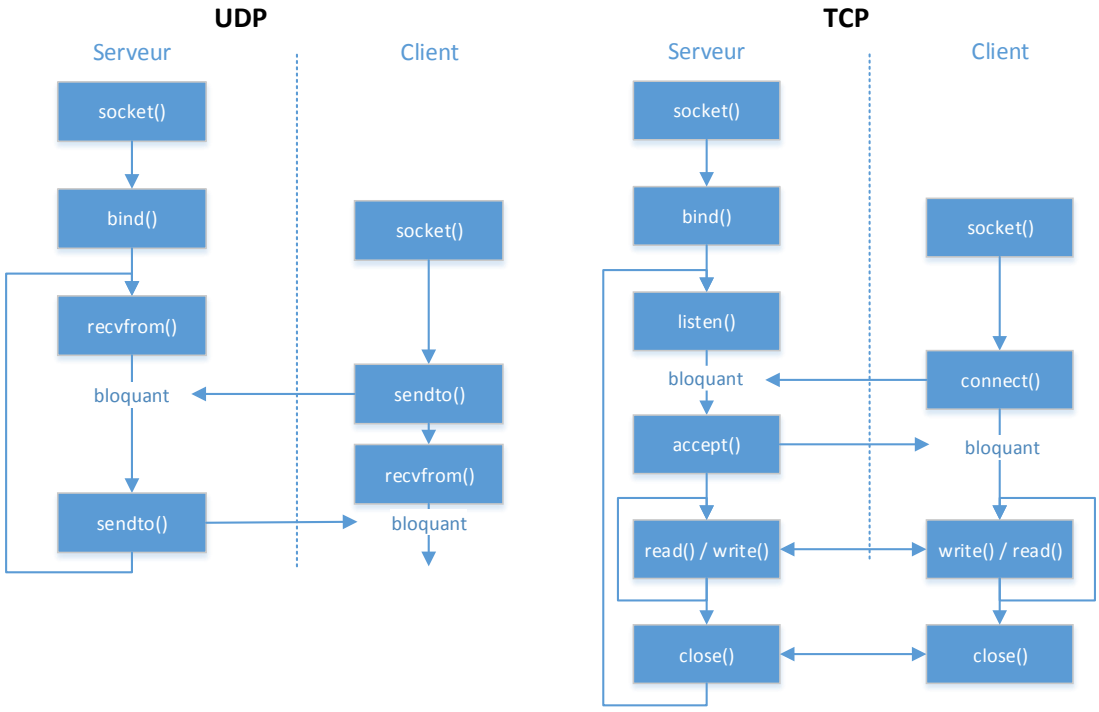
2. Envoi du message en UDP

`sendto(socket, message, long, flags, adressDest, lgAdr)`	Envoi de message
`recvfrom(socket, *buffer, lgMax, flags, *adresseSrc, *lgAdr)`	Reception de message

3. Envoi de message en TCP

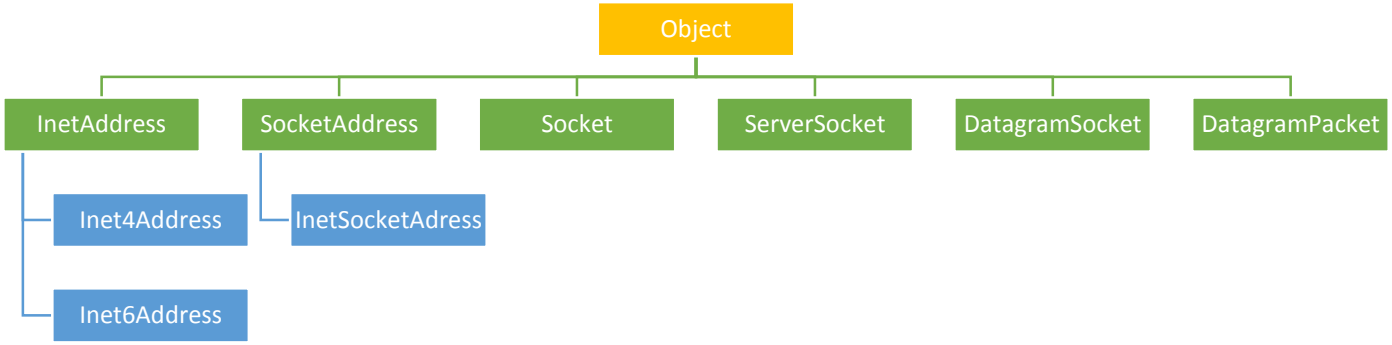
`connect(socket, adresse, lgAdr)`	Connexion client au serveur
`listen(socket, lgFileAttente)`	Ecoute les connexion serveur
`accept(socket, * adresseSrc, *lgAdr)`	Accepte une connexion serveur
`read(socket, *buffer, lgMax)`	Lit un message
`write(socket, message, long)`	Envoie un message
`close(socket)`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

4. Schéma des appels de fonctions



II. Sockets en Java

1. Arbre de classes du package java.net



2. InetAdress

- 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. DatagramPaacket : 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 chaine "calcul" pour en extraire infos.

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