

TP n°8 – Les Sockets

Exercice 1

Récupérer les sources des programmes **serveur_UDP.c** et **client_UDP.c**.

Modifiez ces deux programmes pour qu'ils prennent [II'@IP](#) et le port en paramètre de la ligne de commande.

Exercice 2

Modifier les programmes de l'exercice 1 pour utiliser le protocole **TCP** au lieu d'**UDP**.

Exercice 3

Modifier votre programme serveur TCP de l'**exercice 2** pour que celui-ci gère la connexion multiple de client en utilisant la création de processus fils pour gérer le dialogue avec un client.

La connexion Client-Serveur est interrompu quand le client envoie **EXIT**.

Vous pouvez tester votre serveur avec la commande **nc**.

Exercice 4

Modifier votre programme serveur TCP de l'**exercice 2** pour que celui-ci gère la connexion multiple de client en utilisant l'appel système ***select()*** pour gérer le dialogue avec un client. La connexion Client-Serveur est interrompu quand le client envoie **EXIT**.

Vous pouvez tester votre serveur avec la commande **nc**.

Mémento

UDP

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

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

```
ssize_t recvfrom(int sockfd, void *buf, size_t len, MSG_NOWAIT, NULL, NULL);
```

TCP

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

```
int listen(int sockfd, int backlog);
```

```
int accept(int sockfd, struct sockaddr *adresse, socklen_t *longueur);
```

```
ssize_t send(int s, const void *buf, size_t len, int flags);
```

```
ssize_t recv(int s, void *buf, ssize_t len, int flags);
```

```
int close(int fildes);
```

Fonctionnement de l'appel système *select()*

```
fd_set readfds;           // déclaration de la structure des descripteurs à  
                           surveiller  
  
FD_ZERO(&readfds);        // initialisation de la structure des descripteurs  
                           à surveiller  
  
FD_SET(server_fd, &readfds); // ajout d'un descripteur  
  
max_sd = server_fd;       // définition du descripteur MAX  
  
activity = select(max_sd + 1, &readfds, NULL, NULL, NULL); // surveillance des  
                                                           descripteurs  
  
FD_ISSET(server_fd, &read_fds) // test si un descripteur est prêt à  
                               être lu
```