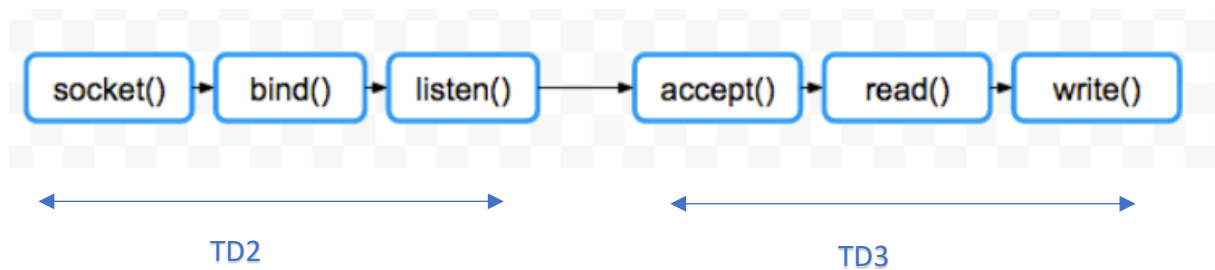


**TD3**  
**Serveur d'écho à partir du TD précédent**  
 Christian Toinard



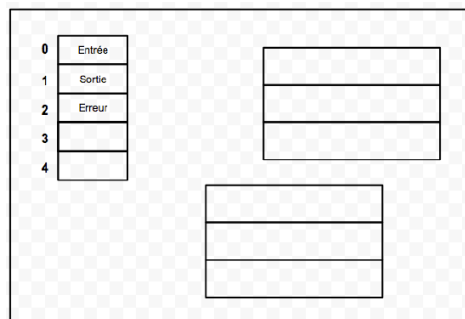
### Question 1

Vous reprenez le code du TD corrigé précédent.

- 1.1 Déclarez une structure `sockaddr_in` distantExtremity qui permettra de récupérer l'extrémité distante du nouveau client.
- 1.2 Quel est l'appel système pour récupérer le numéro de descripteur `nDescClientConnection` de la connexion établie avec le client.
- 1.3 Quels paramètres sont nécessaires pour cet appel ? Que retourne-t-il ? Les paramètres sont-ils des entrées ou sorties par rapport au noyau ?
- 1.4 Donnez le code de l'appel.

### Question 2

- 2.1 Votre appel système vous retourne l'extrémité distante, vous allez afficher cette extrémité en vous aidant des fonctions `ntohs()` et `inet_ntop()`.
- 2.2 Complétez le schéma suivant afin de faire apparaître ce qui "correspond par analogie" aux deux prises, numéros de prise, câbles et combinés d'une communication téléphonique.



### Question 3

3.1 Sur la connexion établie avec le client telnet, nous allons pouvoir réceptionner et envoyer des données. Pour ce faire vous utilisez les appels `read()` et `write()`. On souhaite mettre en place une réponse automatique du serveur "Received:" suivi de la chaîne de caractère du client.

Pour la lecture vous utilisez quelque chose du type, en expliquant la dernière ligne :

```
char receivedBuffer[BUFFER_LEN] = {0};
int nBytesRead = read(nDescClientConnection, receivedBuffer, BUFFER_LEN);
receivedBuffer[nBytesRead]='\0'; /* A quoi sert cette ligne */
```

Pour l'écriture, vous utilisez quelque chose du type, en justifiant les trois écritures :

```
char begin[] = "Received: ";
int writeResult = write(nDescClientConnection, begin, sizeof(begin));
int writeResult = write(nDescClientConnection, receivedBuffer, sizeof(receivedBuffer));
char end = '\n';
int writeResult = write(nDescClientConnection, end, sizeof(end));
```

3.2 Réalisez une boucle infinie de lecture.

## Question 4

4.1 Pour l'instant, votre programme ne détecte pas les ruptures de connexion. Mettez en place une détection des ruptures de connexion qui permet de sortir de la boucle infinie de lecture. Aidez-vous des informations données pour l'appel `read()` de la page 23 du PDF de cours.

4.2 Testez le traitement de la rupture de connexion grâce à `telnet`. Comment faites-vous ?

## Bonus

Il est possible que votre code ne vérifie pas les erreurs systèmes. Mettez en place les vérifications d'erreur des appels systèmes.