

TYPES DES SERVEURS

- **SERVEURS ITÉRATIFS:** C'EST LA FORME LA PLUS SIMPLE. LE SERVEUR GÈRE UN CLIENT À LA FOIS ET LES CLIENTS ATTENDENT POUR S'Y CONNECTER.
- UNE SOCKET SERVEUR EST EN ATTENTE DE DEMANDES DE CONNEXION.
- SI UNE DEMANDE ARRIVE PENDANT QU'UNE AUTRE EST EN COURS DE TRAITEMENT, ELLE EST PLACÉE DANS UNE FILE D'ATTENTE.
- SI UNE DEMANDE ARRIVE ALORS QUE LA FILE EST PLEINE, ELLE EST REJETÉE;
- **SERVEURS CONCURRENTS :** CRÉER PLUSIEURS PROCESSUS EN PARALLÈLE ET PERMET DE SERVIR PLUSIEURS REQUÊTES À LA FOIS.
- LA FORME LA PLUS SIMPLE POUR IMPLÉMENTER UN SERVEUR CONCURRENT SOUS UNIX EST DE CRÉER UN PROCESSUS ENFANT POUR PRENDRE EN CHARGE CHAQUE CLIENT.

SERVEUR EN MODE CONCURRENT

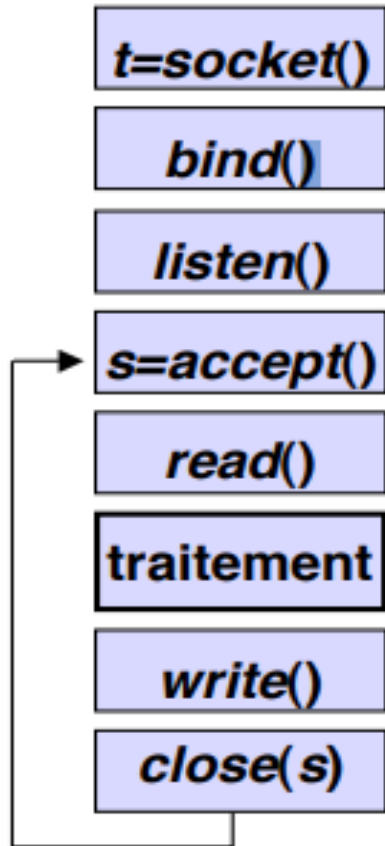
- NÉCESSITÉ DE PLUSIEURS PROCESSUS OU THREADS
- NOUVELLE CONNEXION : UN NOUVEAU PROCESSUS EST CRÉÉ POUR GÉRER LA COMMUNICATION AVEC LE NOUVEAU CLIENT

SERVEUR EN MODE CONCURRENT

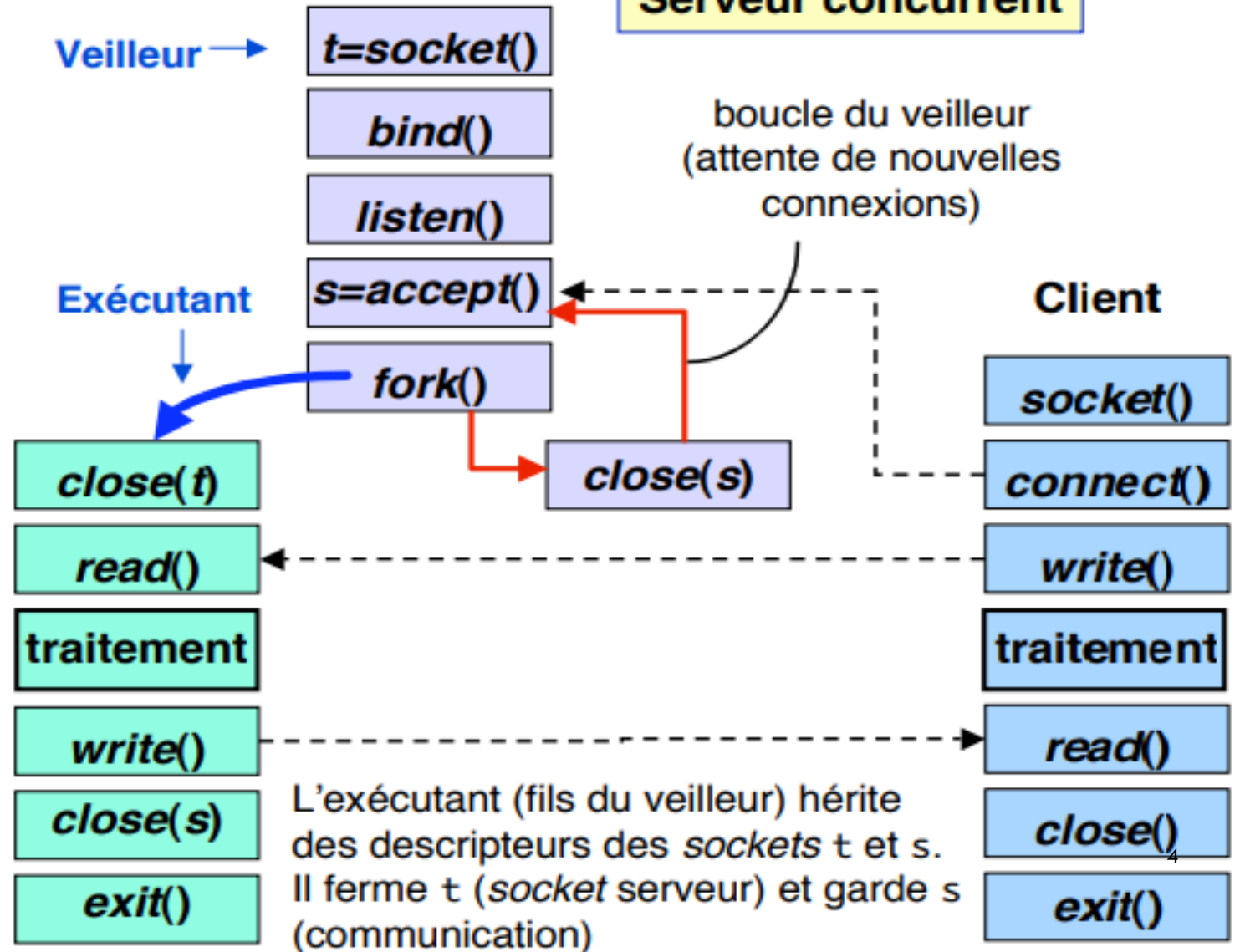
- POUR RÉALISER UN SERVEUR EN MODE CONCURRENT, LA SOLUTION CONSISTE À:
- CRÉER UN NOUVEAU PROCESSUS POUR SERVIR CHAQUE DEMANDE DE CONNEXION.
- LE PROGRAMME PRINCIPAL DU SERVEUR NE FAISANT QUE LA BOUCLE D'ATTENTE SUR LES DEMANDES DE CONNEXION. (PROCESSUS PRINCIPAL (APPELÉ VEILLEUR) QUI ATTEND SUR ACCEPT()).
- LORSQUE LE VEILLEUR REÇOIT UNE DEMANDE DE CONNEXION, IL CRÉE UN PROCESSUS FILS (APPELÉ EXÉCUTANT) QUI VA INTERAGIR AVEC LE CLIENT.
- LE VEILLEUR REVIENT ET SE METTRE EN ATTENTE SUR ACCEPT(). PLUSIEURS EXÉCUTANTS PEUVENT EXISTER SIMULTANÉMENT.
- IL EXISTE D AUTRES SOLUTIONS (THREADS,...).

SOCKETS TCP – GESTION PLUSIEURS CLIENTS

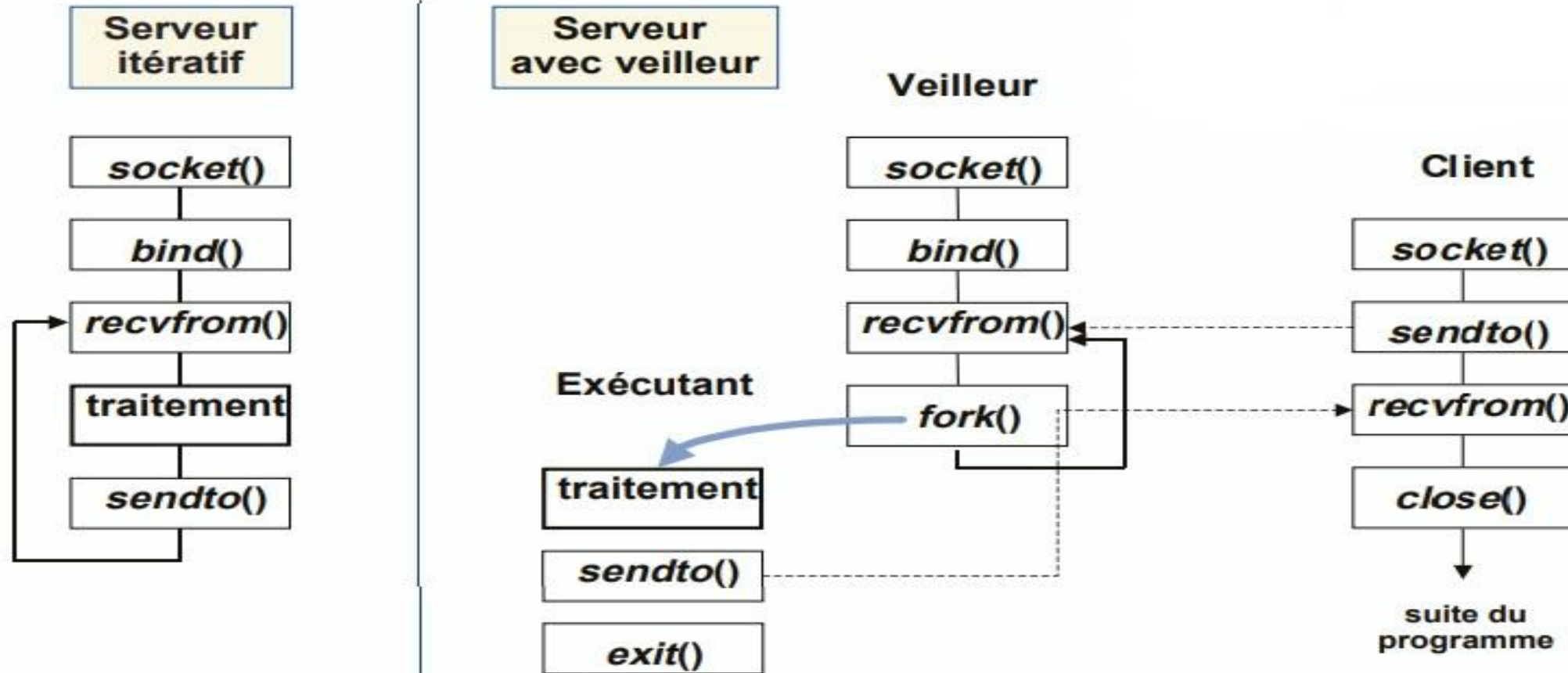
Serveur itératif



Serveur concurrent



SOCKETS UDP– GESTION PLUSIEURS CLIENTS



SERVEUR TCP SÉQUENTIEL

- LE SERVEUR NE PEUT GÉRER QU'UNE CONNEXION À LA FOIS :

```
s = socket (...);
```

```
bind (...);
```

```
listen (s, ...);
```

```
for (;;) 
```

```
{
```

```
ns = accept (s, ...);
```

```
executer_service (ns, ...);
```

```
close (ns);
```

```
}
```

SERVEUR TCP CONCURRENT

- LE SERVEUR PEUT GÉRER PLUSIEURS CONNEXIONS EN PARALLÈLE :

```
S = SOCKET (...); BIND (...); LISTEN (S, ...);
```

```
FOR (;;) 
```

```
{      NS = ACCEPT (S, ...);
```

```
      IF (FORK() == 0)
```

```
      {
```

```
      CLOSE (S);
```

```
      EXECUTER_SERVICE (NS, ...);
```

```
      CLOSE (NS);
```

```
      }
```

```
      ELSE {
```

```
      CLOSE (NS);
```

```
      }
```

```
}
```