

Servidores multi-cliente (draft)

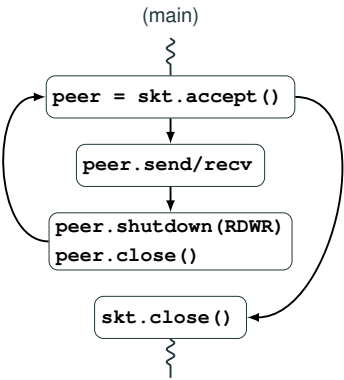
Di Paola Martín
martinp.dipaola <at> gmail.com
Facultad de Ingeniería
Universidad de Buenos Aires

1

Servidores multi-cliente (draft)

2

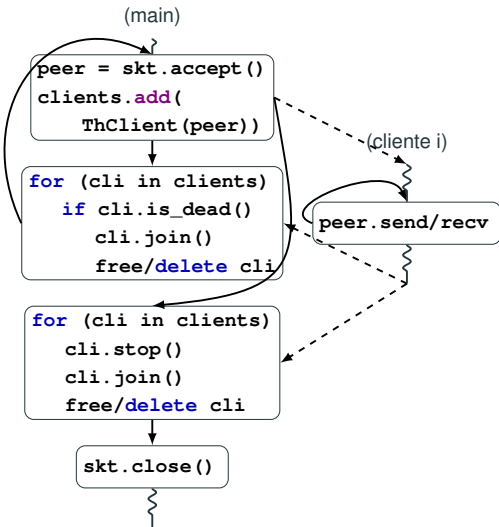
Simple: servidor para un solo cliente a la vez



3

- El programa es un bucle simple, se espera una conexión remota, se la procesa y se repite el proceso
- Solo cuando `accept` falla por recibir una señal, el bucle finaliza.
- Este esquema solo soporta un cliente a la vez y no permite hacer cosas en paralelo mientras se espera a un nuevo cliente ni mientras se habla con uno ya conectado.
- Es muy simple. Usado en servidores RPC y servidores Web simples (o dummies).

Servidor multi-cliente (draft)



4

Frenar un hilo

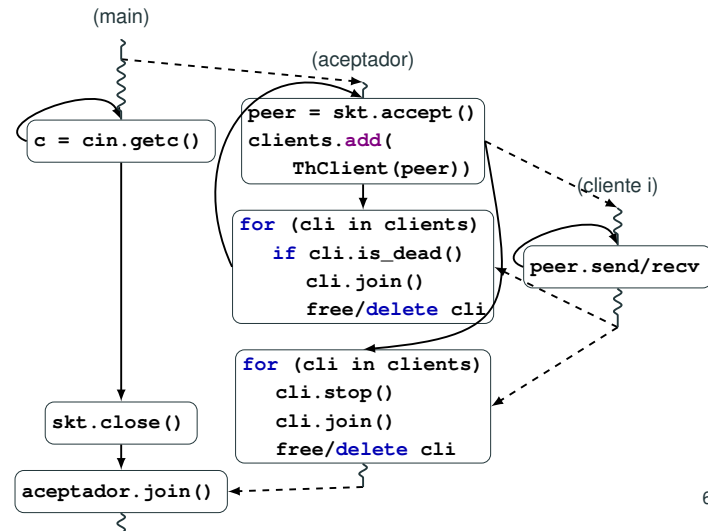
```
1 class ThClient:public Thread {
2     std::atomic<bool> keep_talking;
3     std::atomic<bool> is_running;
4     Socket peer;
5
6     public:
7     ThClient():keep_talking(true),
8             is_running(true) {}
9
10    virtual void run() {
11        while (keep_talking) {
12            ...
13            peer.send(...);
14            ...
15            peer.rcv(...);
16            ...
17        }
18        is_running = false;
19    }
```

```
1 // Violento pero efectivo
2 void ThClient::stop() {
3     keep_talking = false;
4     peer.shutdown();
5     peer.close();
6 }
7
8 // Polite pero peligroso
9 void ThClient::stop() {
10    keep_talking = false;
```

5

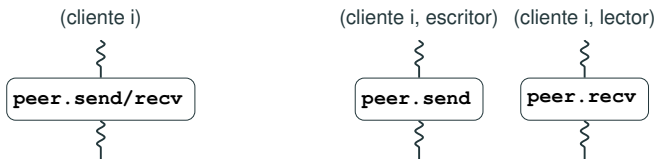
- Como frenar un hilo? La librería `pthread` ofrece una manera genérica de frenar o matar a un hilo (stop/kill) pero deja **los recursos sin finalizar. No usar.**
- Frenar un hilo correctamente dependerá de la naturaleza del hilo (depende de la aplicación en cuestión, no hay una solución general).
- Hay dos variantes posibles: forzar un cierre o decirle al hilo que cuando pueda él mismo finalice.
- Si el hilo esta bloqueado en una operación de sockets, se puede hacer un `shutdown/close` del socket para forzar un cierre. Obviamente si el hilo estaba en el medio de una comunicación, el trabajo puede quedar trunco o corrupto.
- Usar un `bool` para decirle al hilo que finalice. El hilo puede terminar su conversación y cerrar ordenadamente: es más seguro pero si el hilo está bloqueado **no** se desbloqueará.
- Y si el hilo esta bloqueado en otra operación? Como en un `queue_safe.pull()`? No hay una solución genérica.

Servidor multi-cliente (final): cierre ordenado



6

Hilos de comunicación



- Un socket puede ser leído (`recv`) por un hilo y escrito (`send`) por otro. La lectura no entra en conflicto con la escritura.
- Pero hacer un `send` (o un `recv`) sobre un mismo socket desde múltiples hilos trae problemas de concurrencia!

7