

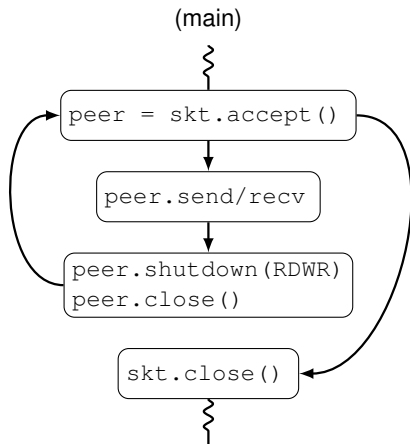
Sockets avanzado

(Ultra DRAFT)

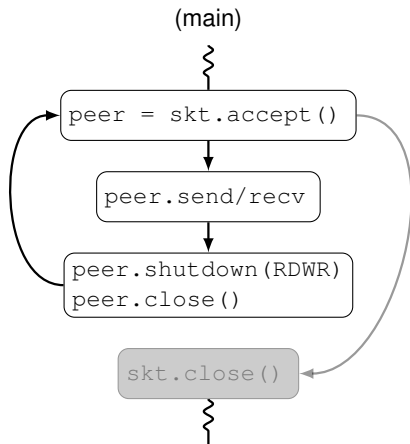
¹Facultad de Ingeniería
Universidad de Buenos Aires



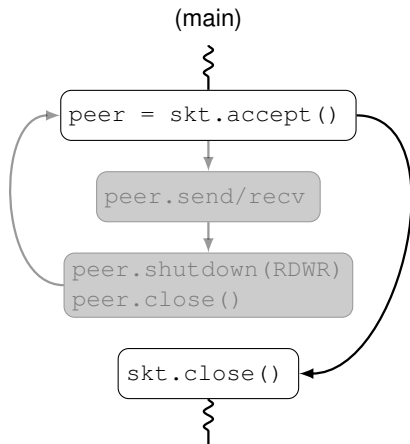
Server mono hilo



Server mono hilo

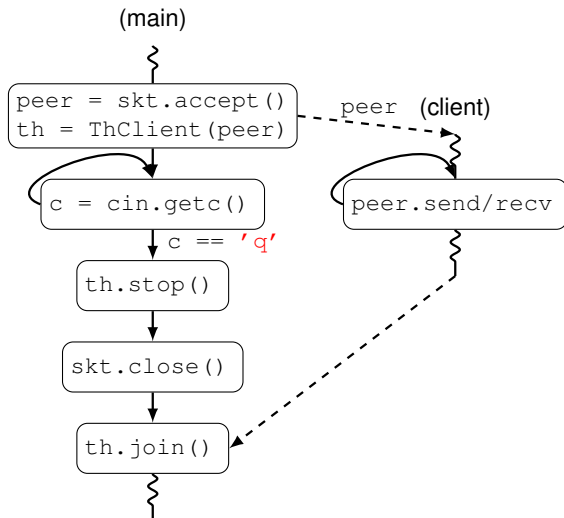


Server mono hilo



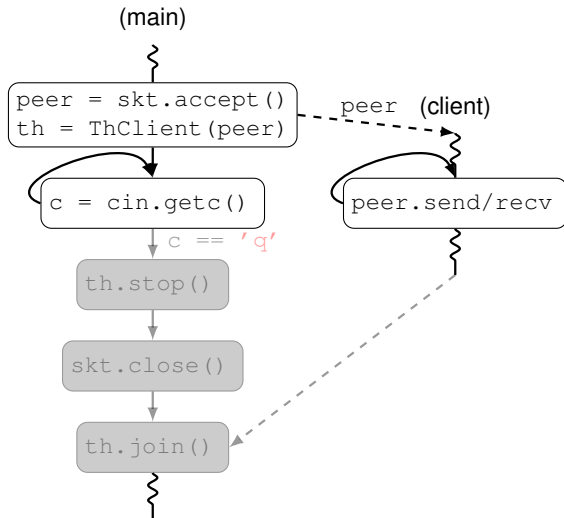
Server con 2 hilos

Maneja un cliente a la vez



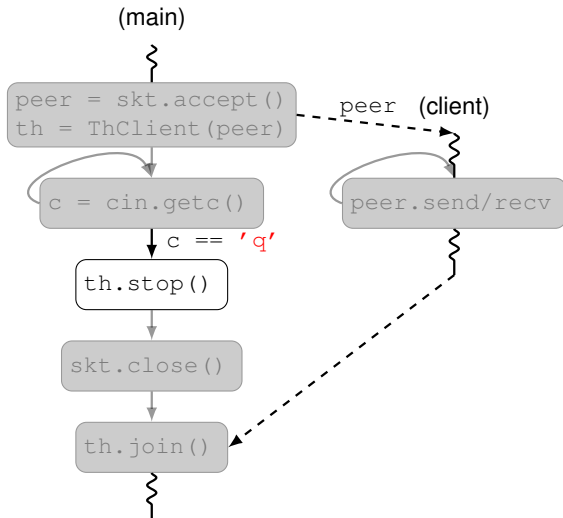
Server con 2 hilos

Maneja un cliente a la vez



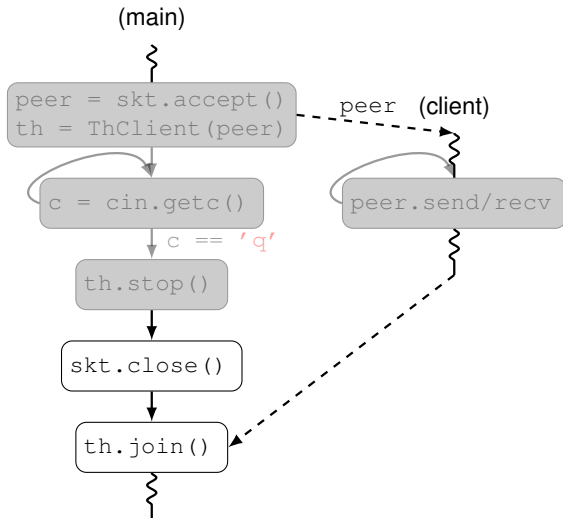
Server con 2 hilos

Maneja un cliente a la vez



Server con 2 hilos

Maneja un cliente a la vez



Frenado de un hilo

Dos alternativas para frenar un hilo con un socket asociado

```

1  class ThClient:public Thread {
2      bool keep_talking;
3      Socket peer;
4
5      public:
6      virtual void run() {
7          while (keep_talking) {
8              ...
9              peer.send(...);
10             ...
11             peer.recv(...);
12             ...
13         }
14     }
15
16     void stop();
17 };

```

```

1  // Violento pero efectivo
2  void ThClient::stop() {
3      keep_talking = false;
4      peer.shutdown();
5      peer.close();
6  };

```

```

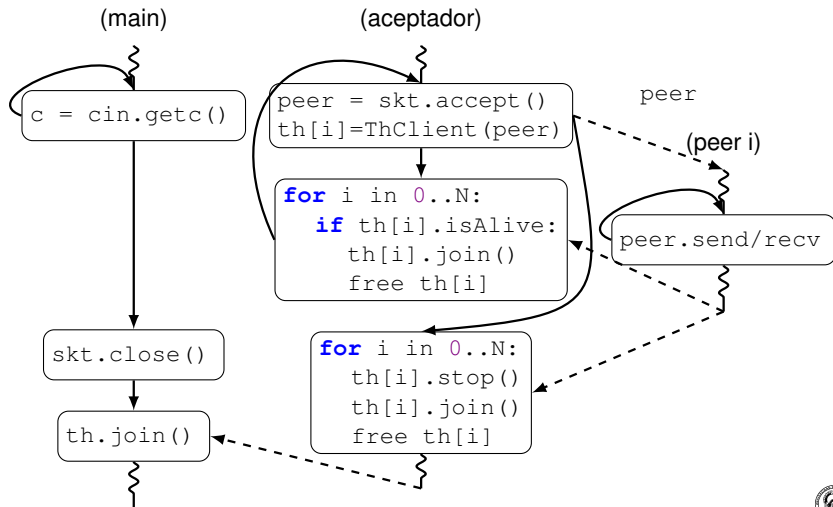
1  // Polite pero peligroso
2  void ThClient::stop() {
3      keep_talking = false;
4  };

```



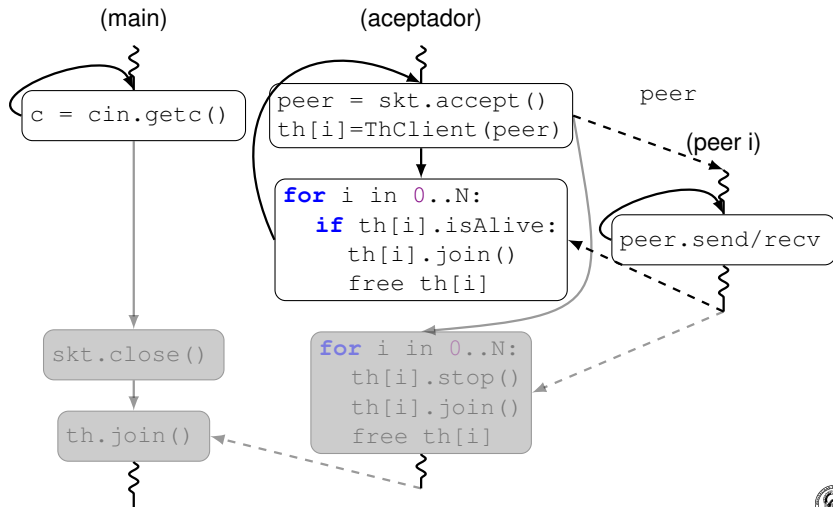
Server multi hilo

Un hilo por cliente (al menos)



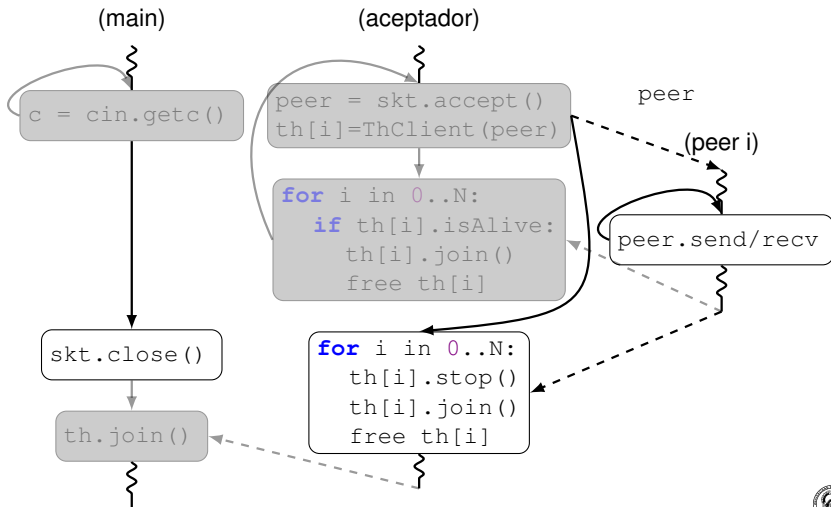
Server multi hilo

Un hilo por cliente (al menos)



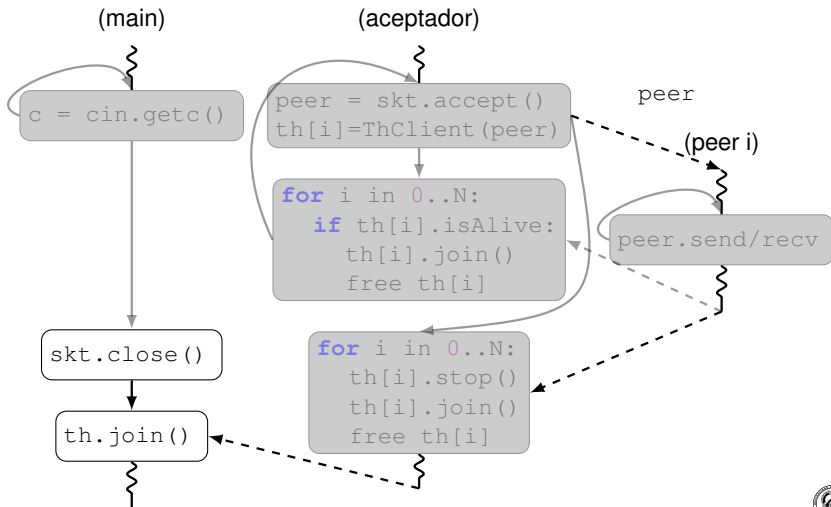
Server multi hilo

Un hilo por cliente (al menos)



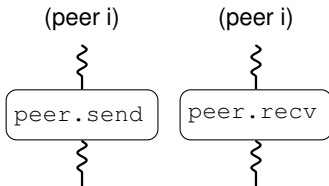
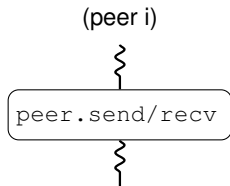
Server multi hilo

Un hilo por cliente (al menos)



Hilos de comunicacion

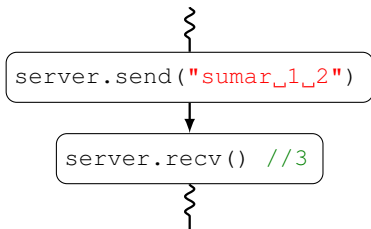
Un unico hilo para enviar/recibir o dos hilos separados



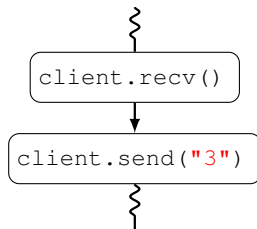
Comunicacion cliente servidor

Abstracciones ClienteProxy y ServidorProxy

(client i)



(client i, server side)



Resumen

- Detectar que acciones son bloqueantes: accept, send, recv
No solo las acciones de los socket son bloqueantes: join, fwrite, fread
- Por cada accion bloqueante preguntarse, si el hilo se bloqueara, se deberia poder hacer otra cosa en simultaneo? Si es asi, separarlos en 2 hilos.
Ej: hacer un request (send) a <http://fi.uba.ar> y esperar (recv) a recibir la pagina web para renderizarla (un hilo).
Ej: mientras se hace se procesa un request a una pagina, se procesan en paralelo otros requests (multiples hilos).



Resumen

- Un socket puede ser leído (recv) por un hilo y escrito (send) por otro. La lectura no entra en conflicto con la escritura.
- Hacer un send sobre un mismo socket desde multiples hilos trae problemas de concurrencia! Lo mismo para multiples concurrentes recv.

