|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Server file | msg | Client One | msg | Client Two | msg |
| ./chat\_server 9000 |  |  |  |  |  |
| asks for incoming connections permission. |  |  |  |  |  |
| do\_accept() |  |  |  |  |  |
|  |  | chat\_client 127.0.0.1 9000 |  |  |  |
|  |  | do\_connect() |  |  |  |
| (!ec) |  | (!ec) |  |  |  |
| start() |  | do\_read\_header() |  |  |  |
| join() |  | —async read, buffer— | data() |  |  |
| do\_read\_header() |  |  |  |  |  |
| —async read, buffer— | data() |  |  |  |  |
| second do\_accept() |  |  |  |  |  |
|  |  |  |  | ./chat\_client 127.0.0.1 9000 |  |
|  |  |  |  | do\_connect() |  |
| (!ec) |  |  |  | (!ec) |  |
| start() |  |  |  | do\_read\_header() |  |
| join() |  |  |  | —async read, buffer— | data() |
| do\_read\_header() |  |  |  |  |  |
| —async read, buffer— | data() |  |  |  |  |
| do\_accept() |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  | \*HELLO\* |  |  |  |
|  |  |  | body\_length() |  |  |
|  |  |  | body() |  |  |
|  |  |  | const body\_length() |  |  |
|  |  |  | encode\_header() |  |  |
|  |  | write() |  |  |  |
|  |  | (!write\_in\_progress) in write |  |  |  |
|  |  | do\_write() |  |  |  |
|  |  |  | data() |  |  |
|  |  |  | length() |  |  |
| —async read connects— from do\_read\_header | decode\_header() | —async write, buffer—(!ec) in do\_write |  |  |  |
| (!ec) in do\_read\_header |  |  |  |  |  |
| do\_read\_body() —async read, buffer— |  |  |  |  |  |
|  | body() |  |  |  |  |
|  | const body\_length() |  |  |  |  |
| (!ec) in do\_read\_body |  |  |  |  |  |
| room deliver() |  |  |  |  |  |
| participant->deliver(msg) |  |  |  |  |  |
| deliver() from session |  |  |  |  |  |
| (!write\_in\_session) in deliver |  |  |  |  |  |
| do\_write() —async write, buffer— |  |  |  |  |  |
|  | data() |  |  |  |  |
|  | length() |  |  |  |  |
| participant->deliver(msg) |  | —async read connects— in do\_read\_header | decode\_header() |  |  |
| deliver() to every participant |  |  | (!ec && decode\_header) in do\_read\_header |  |  |
| (!write\_in\_progress) |  | do\_ready\_body() |  |  |  |
| do\_write() |  |  | body() |  |  |
|  | data() | —async read, buffer—from do\_read\_body | const body\_length() |  |  |
| —async write, buffer— from do\_write | length() | (!ec) from do\_read\_body |  |  |  |
| do\_read\_header() |  |  | body() |  | decode\_header() |
| —async read, buffer— | data() |  | body\_length() & \*HELLO\* | (!ec & decode\_header() in do\_read\_header |  |
| (!ec) in do\_write |  | do\_read\_header() |  | do\_read\_body() |  |
| (!ec) in do\_write |  |  | data() |  | body() |
|  |  |  |  |  | body\_length() |
|  |  |  |  | (!ec) in do\_read\_body) |  |
|  |  |  |  |  | body() |
|  |  |  |  | HELLO (received) | const body\_length() |
|  |  |  |  | do\_read\_header() |  |
|  |  |  |  | —async read, buffer— | data() |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

async connect and async accept

both do\_read\_header() async read wait

when client do\_write() async\_write works

do\_read\_header() async\_read connects

server do\_read\_body() async\_read waits

while do\_write() async\_write connects to client do\_read\_header() async\_read