MODULE *Client_Server*

EXTENDS *Server*, *Client*
VARIABLE *locked*, *held*

$Init \triangleq$
$\quad \land locked = [i \in Server \mapsto \text{TRUE}]$
$\quad \land held = [i \in Client \mapsto \{\}]$

$Connect(client, server) \triangleq$
$\quad \land \quad locked[server] = \text{TRUE}$
$\quad \land \quad held' = [held \text{ EXCEPT } ![client] = held[client] \cup \{server\}]$
$\quad \land \quad locked' = [locked \text{ EXCEPT } ![server] = \text{FALSE}]$

$Disconnect(client, server) \triangleq$
$\quad \land server \in held[client]$
$\quad \land held' = [held \text{ EXCEPT } ![client] = held[client] \setminus \{server\}]$
$\quad \land locked' = [locked \text{ EXCEPT } ![server] = \text{TRUE}]$

$Next \triangleq$
$\quad \lor \exists\, client \in Client,\ server \in Server : Connect(client, server)$
$\quad \lor \exists\, client \in Client,\ server \in Server : Disconnect(client, server)$

$Spec \triangleq Init \land \Box[Next]_{\langle locked,\, held \rangle}$

$Safe \triangleq$
$\quad \forall\, client\_ < i >,\ client\_j \in Client :$
$\quad (held[client\_i] \cap held[client\_j] \neq \{\}) \lor (client\_i = client\_j)$