```cpp
#include "Server.h"

// Vecteur donnant le numero du ferry présent dans chaque terminal
// (0 si aucun ferry)
int docked_ferries[] = { 1, 4, 0, 3, 7, 0 };

// Ferrys en attente de traitement dans l'un des terminaux
queue<int> waiting_ferries;
pthread_mutex_t mutex_waiting;

// Ferrys en cours de départ
list<int> leaving_ferries;
pthread_mutex_t mutex_leaving;

// Liste des clients connectés au serveur (numero du terminal)
list<int> connected_clients;
pthread_mutex_t mutex_connected;

// Liste des sockets des clients connectés sur le serveur de status
list<ClientSocket> status_clients;
pthread_mutex_t mutex_status;

int main(void)
{
    pthread_mutex_init(&mutex_waiting, NULL);
    pthread_mutex_init(&mutex_leaving, NULL);
    pthread_mutex_init(&mutex_connected, NULL);
    pthread_mutex_init(&mutex_status, NULL);

    pthread_mutex_init(&mutex_pause, NULL);

    waiting_ferries.push(5);
    waiting_ferries.push(2);
    waiting_ferries.push(6);

    printf("Occupation des terminaux: \n");
    for (int i = 0; i < 6; i++) {
        printf("    Terminal %d: %d\n", i+1, docked_ferries[i]);
    }

    async_call(status_server, NULL);
    async_call(terminal_server, NULL);
    async_call(admin_server, NULL);
    inout_server(NULL);

    pthread_mutex_destroy(&mutex_leaving);
    pthread_mutex_destroy(&mutex_waiting);
    pthread_mutex_destroy(&mutex_connected);
    pthread_mutex_destroy(&mutex_status);

    pthread_mutex_destroy(&mutex_pause);

    return 0;
}
```