```cpp
#include "StatusServer.h"

void _add_client(ClientSocket sock);

// Serveur assurant la transmission des status du serveur aux clients
void *status_server(void* arg)
{
    IniParser properties("status_server.ini");

    int port = atoi(properties.get_value("port").c_str());
    int n_clients = atoi(properties.get_value("n_clients").c_str());

    with_server_socket(port, n_clients, _add_client);
    return NULL;
}

// Ajoute le client à la liste des clients qui suivent le status du serveur
void _add_client(ClientSocket sock)
{
    pthread_mutex_lock(&mutex_status);
    status_clients.push_front(sock);
    pthread_mutex_unlock(&mutex_status);
}

// Signale le status à tous les clients connectés
void signal_status(char status)
{
    pthread_mutex_lock(&mutex_status);
    list<ClientSocket>::iterator it;
    for (it = status_clients.begin(); it != status_clients.end(); it++) {
        try {
            it->send<char>(&status);
        } catch (SocketException e) {
            status_clients.erase(it);
        }
    }
    pthread_mutex_unlock(&mutex_status);
}
```