

```
#include "InOutServer.h"

void _inout(ClientSocket sock);

// Serveur assurant la gestion des entrées et des sorties des ferrys.
void *inout_server(void* arg)
{
    IniParser properties("inout_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, _inout);
    return NULL;
}

// Enregistre les entrées et les sorties
void _inout(ClientSocket sock)
{
    inout_protocol packet;
    sock.receive<char>((char *) &packet.type);
    sock.receive_string(packet.ferry_id);
    sock.receive_string(packet.time);

    if (packet.type == inout_protocol::SHIP IN) {
        int ferry_id = atoi(packet.ferry_id);
        waiting_ferries.push(ferry_id);
        printf("%s: Ajout du ferry numero %d\n", packet.time, ferry_id);
        return _inout(sock);
    } else if (packet.type == inout_protocol::SHIP OUT) {
        int ferry_id = atoi(packet.ferry_id);
        leaving_ferries.remove(ferry_id);
        printf("%s: Sortie du du ferry numero %d\n", packet.time, ferry_id);
        return _inout(sock);
    }
    sock.close();
}
```