Exercise Sheet 4

Deadline: June 24, 2024 – 04:00 am CEST

Exercise 1 Project "‘Room Manager with Sun RPC"’

As you have experienced in the previous exercise sheet the manual development of the RPC protocol via sockets is cumbersome and error-prone. Hence, the university management decides to switch to a standardized RPC system: *Sun RPC*. Sun RPC is a widely used solution providing a good efficiency and allows for a good integration with the room manager library because its C interface.

Sun RPC

Once again you will find a *git submodule* in the subdirectory `04` of your repository which links the room manager library into the `libsrc` subdirectory. The subdirectory `src` includes a file called `roomman_cli.c` which contains the source of a simple menu program using the API of the roomman library. In order to compile it, simple use `make` from the main directory and execute the resulting application `roomman_climenu`.

Your task at hand is to convert this application into a distributed one. Hence, a **server** (`roommanager_server`) and a **client** (`roommanager_client`) using the Sun RPC for communication need to be implemented.

The following steps are required to achieve this goal:

1. Create an RPC **interface specification** `roommanager.x` in the `src` subdirectory.

2. Autogenerate the **RPC stubs** `roommanager_clnt.c`, `roommanager_svc.c`, the header file `roommanager.h`, and the **templates** `roommanager_server.c` and `roommanager_client.c`. In order to do so, you need to call:

   ```
   $ rpcgen -a roommanager.x
   ```

3. Manually adapt the client and server side templates `roommanager_server.c` and `roommanager_client.c` in a way that the server is calling interface of the roommanager library and the client offers an interface identical to this interface using the RPC interface.

4. Build the client and server side of the application `roommanager_cliclient` und `roommanager_server` by calling `make all` from the main directory.

Sun RPC Tests

For testing purposes the directory `tests` contains two files `refinput.txt` and `refoutput.txt`. `refinput.txt` contains a recorded sequence of inputs for the menu application and `refoutput.txt` contains the corresponding output from the reference implementation. For running the test you first need to start the server and then feed the reference input via *stdin* into the client:

```
$ ./roommanager_server &
$ cat tests/refinput.txt | ./roommanager_client >myoutput.txt
```

The generated output file `myoutput.txt` should look similar to the the reference output. In order to compare the differences, you could use the tools `diff` or `meld`.

Hints

At first, familiarize yourself with SunRPC using a simple example. There are plenty of Sun RPC tutorials in the Internet, like the one at `https://www.cs.rutgers.edu/~pxk/417/notes/rpc/index.html`. Try the demonstrated example in order to get to know the usage of the tools.

- In order to use Sun RPC on you own (private) computer you many need some additional packages. On Debian based systems (like Ubuntu) the packages are called `rpcbind` and `libntirpc-dev`.
- Sun RPC requires the portmap service (cf. the slides from the lecture). Potentially you first need to start this service. Depending on your system it may be called `portmapper`, `rpcbind`, `portmapd` …