barkhausen institut

IPC

Microkernel-based Operating Systems—Exercise

Barkhausen Institut, Dresden (Germany)

Viktor Reusch, Martin Küttler

WiSe 26/27

IPC overview



- External resources are accessed through capabilities.
- Applications have a namespace containing capabilities.
- Some magically appear at startup, others can be added in startup script.
- Communication requires capability to an IPC gate.

Lua Config Scripts



Example Ned config scripts can be found in

- src/l4/conf/examples
- src/l4/pkg/l4re-core/ned/doc
- src/l4/pkg/examples/clntsrv/configs

General idea for our simple usecase:

- Programs are started with `L4.default_loader.start`
- They take capabilities to channels (aka IPC gates) that are created with `L4.default_loader.new_channel`
- Programs can get a client or a server (`:svr()`) capability via the `caps` parameter to the `start` function.

IPC

In the program, these capabilities can be used to send and receive messages.

Server side:

- Register Server object (that implements handler) with the named cap.
- Run server loop
- Handle requests

Client side:

- Query IPC gate capability at name service.
- Invoke cap with arguments.

Sending data works with the UTCB (lecture). We use a high-level interface on top.

IPC interface



- Define a class that inherits from `L4::Kobject_t` that has a member function for each message type.

```
struct MyClass : public L4::Kobject_t<MyClass, L4::Kobject, MY_PROTO_NUM>
{
  L4_INLINE_RPC(int, foo, (int arg1, int arg2));
  typedef L4::Typeid::Rpcs<foo_t> Rpcs;
};
```

- Functions return error code.
- All functions are listed in type Rpcs.

IPC interface



- The server implements a class inheriting `L4::Epiface` that implements the methods.
- Types are converted between interface and implementation; special types for inout/out parameters, capabilities, arrays, etc. are available.
- If you need to send caps, the `L4::Kobject_t` needs a fourth template argument of type `L4::Type_info::Demand_t<NUM>`, with NUM being the maximum number of caps per call.

Get help



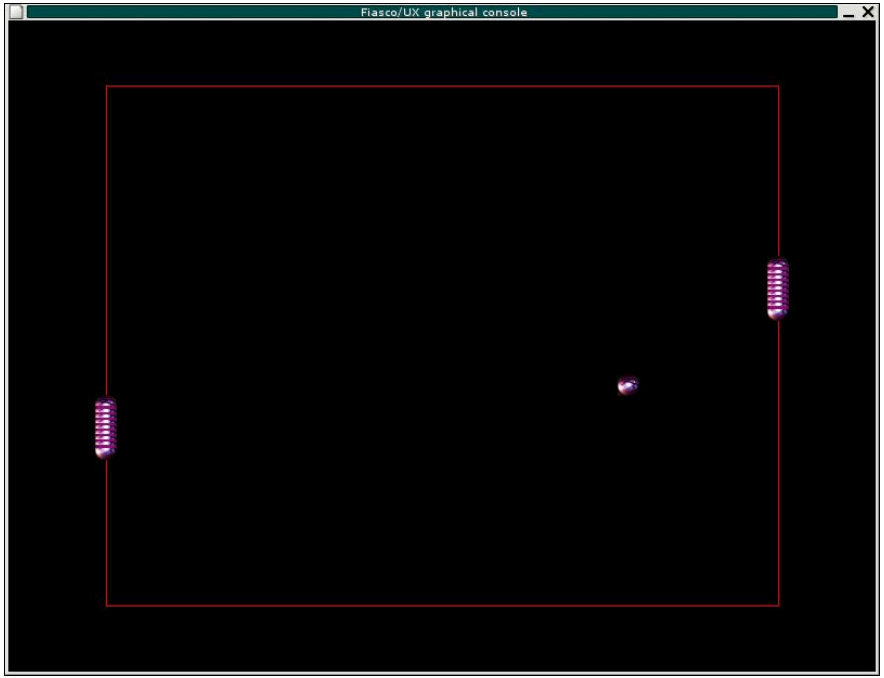
- See `src/l4/pkg/examples/clntsrv` for an example.
- See documentation `doc/source/html/index.html` (specifically, e.g., `doc/source/html/l4_cxx_ipc_iface.html`)
- The source is the ultimate documentation (and it includes all the normal documentation).
- When you have questions, use the mailing list (mos2026), matrix room, or send us an email directly.

Assignment 3



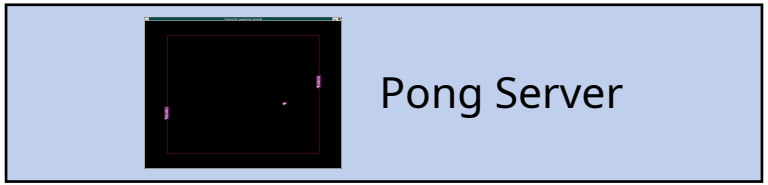
- Try to run and understand the `clntsrv` example.
- Build a client-server version of hello: The client should send a string to the server via IPC.
- As a second step, also consider large strings.
- In case of problems or for discussing solutions, write in the matrix room.

Goal of These Assignments



Multi-user Pong Game

Goal of These Assignments



Paddle Client 1

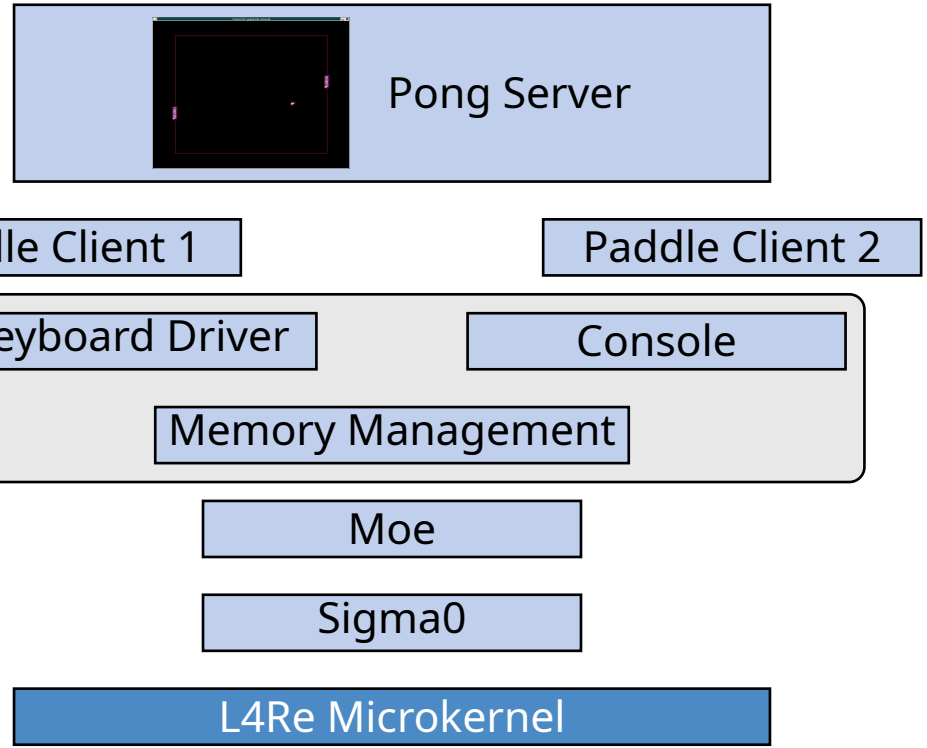
Paddle Client 2

Moe

Sigma0

L4Re Microkernel

Goal of These Assignments



Live Demo

Let’s have a look at the clntsrv example together.

Next meeting

- We will discuss your solution next time we meet for this lab.
- Then we start the real project.