



Universität Karlsruhe (TH)
Institut für
Betriebs- und Dialogsysteme
Lehrstuhl Systemarchitektur

System Design and Implementation Assignment 2

Lecturer: Jan Stoess
Tutors: Marcel Noe

1 Introduction

In assignment 1 you have learned how to create threads and address spaces, and how to manually send IPC messages between threads. The following assignment will teach you to generate client and server communication code using the IDL4 stub code compiler. IDL stands for Interface Definition Language; the IDL4 compiler allows you to automatically create communication stub code for L4 clients and servers, based on an interface defined in CORBA IDL.

2 Building a Syscall Server

In L4, privileged system calls can only be successfully executed by privileged threads. In the current L4 implementation, all threads residing in the root task's address space are defined to be privileged.

In this assignment you will develop a simple *privileged syscall server* that runs within the root task's address space and exports an IPC interface to clients allowing them to invoke privileged system calls. You will define the interface in IDL, and leverage the IDL4 compiler to generate the interface code from your interface definition. The following steps outline the procedure; although the steps are specific to the syscall server, they may still serve as a rough guideline for implementing new servers or interfaces.

1. Identify the privileged system calls.
Hint You can safely ignore `MemoryControl` and `ProcessorControl`; you won't need them in SDI.
2. Identify the parameters and results of each privileged system call.
3. Create a new IDL file in `sdios/if/` and place your interface description in it.
4. Create a server template in `sdios/src/root` using `idl4` with the `-t` option. See the SDI Wiki for details.
5. Fill the functions in the server template as appropriate.
6. Start the privileged system call server as another thread in the root task.