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

System Design and Implementation

Assignment 1

Lecturer: Jan Stoess
Tutors: Marcel Noe

1 Introduction

This assignment will teach you how to build and set up a rudimentary OS skeleton running on top of the L4 μkernel. The OS stub will serve as the starting point for each of your SDI projects. The assigment should further demonstrate you how to use the L4 primitives in practice. You will setup a test and build environment on the machines in room 149. You will extend the provided environment by a simple "Hello World" program. Some introductional remarks:

The Wiki [Aut06c] is the main source for documentation. The Wiki outlines how to setup the development and boot environment, how to boot L4, and how to build and boot applications on top of it. In case you still have questions after browsing the wiki, feel free to ask the tutors. They can be found in Room 154, 50.34; the best time to visit is their regular consultation time on tuesday afternoon.

Documentation Find SDI-related documentation in [Aut06b]; the site references further books and publications related to the lecture, as well as specifications, manuals, and references important for developing SDI servers.

Publications A good starting point for L4-related research is [Aut06a]. The papers are worth reading for people interested in L4 kernel research and development. The most important publications about L4 probably is [Lie95].

2 Setup your build environment

You should have received valid credentials for the machines in room 149. Browse to [Aut06c] follow the links `Installation` and `Quickstart on the machines in room 149`. Follow the instructions to setup a build environment for your account.

Revision: 1.5
April 29, 2009, 16:47 CEST

Common mistakes:

- When downloading the boot image you need to replace the X in sdiX with the number of your account. If VMware gets a valid IP via DHCP but no boot menu is displayed you probably missed something during this step.
- You have to tick the *Connect at power on* checkbox during configuration of your virtual machine. Failure to do so may result in VMware getting an IP via DHCP but not displaying any boot menu.
- Start minicom from the directory which you created your virtual machine in. Whenever your virtual machine is started, a file named vmware-s0 will show up in this directory; it represents the file-mapped terminal minicom uses to communicate with your virtual machine. The correct command to connect with the virtual machine is `minicom vmware`. If you do not start minicom in the correct directory it will display `Offline` in the lower end of your minicom console.

Using minicom You can get help using minicom via `CTRL-a z`. You will find a command reference using `man minicom`. Here are some helpful commands:

- `CTRL-A X` Exit minicom.
- `CTRL-A B` Opens a scroll back buffer. Useful if you have lots of output.
- `CTRL-A Z` Popup a help screen.
- `CTRL-A W` Toggle linewrapping. Lines which are too large to display are wrapped to the next line.

3 Reading Session

Download a copy of the L4 API manual from `http://l4hq.org`. Read and *understand* the following chapters and sections. If you have difficulties in understanding a topic [Aut06c] will provide you with many useful hints:

- Chapter 1 Basic Kernel Interface
- Chapter 2 Threads
- Section 4.3 SpaceControl
- Chapter 5 IPC (focus on 5.1, 5.4, 5.5 and 5.6)

Afterwards you should be able to answer these questions:

- What is the Kernel Interface Page? Who creates it? Can it be mapped? How does one know where it is mapped ?
- What types of thread ids exist? What's the difference between them? Who is allowed to invoke `ThreadControl` and `SpaceControl`? What parameters must be passed to `ThreadControl` to create i) a thread within an existing address space; ii) a thread in a new address space. How are address spaces created? How are they destroyed?
- What types of messages are supported? How are they constructed?

L4 API functions The naming convention is written down in the L4 API Manual (About the Manual-Using the API). In short, every function name that relates to a L4 system-call is prepended by "`L4_`".

4 Thread Creation

The locator service and the logging server are created as a thread within the roottask address space. Create two new threads within the roottask that display 'Hello World Thread #no' using the logging server.

The testclient is created and started in a new address space. Build a second testclient binary. Keep in mind that this involves adding a new directory and replacing the link address of the new binary appropriate. The two testclients shall display "Hello testclient #no" using the logging server.

Hint: The SDI Faq in the Wiki gives an introduction into building a new application.

5 Sending and Receiving IPC

Now we go back to the two threads created in the roottask. Send a message with two DWORDs from the first thread (sender) to the second thread (receiver). The receiver adds the DWORDs together and sends the result back to the sender (reply). Both threads display the message via the logserver, the receiver the two received DWORDs and the sender the result.

Convenience functions Please use the convenience functions for preparing and sending the message item.

References

[Aut06a] Various Authors. L4ka publications. *http://i30www.ira.uka.de/research/publications/l4ka*, 2006.

[Aut06b] Various Authors. SDI documentation. *http://i30www.ira.uka.de/~sdi/doc/*, 2006.

[Aut06c] Various Authors. SDI wiki. *http://i30www.ira.uka.de/~sdi/wiki*, 2006.

[Lie95] Jochen Liedtke. On μkernel construction. In *Proceedings of the 15th Symposium on Operating System Principles*, 1995.