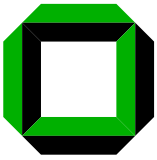

University of Karlsruhe
System Architecture Group
Gerd Liefländer

Teaching Assistant:
Philipp Kupferschmied

System Architecture 2008/09
Assignment 8 Update 1

Question 8.1: Design Parameters of IPC

What are the benefits and drawbacks of each of the following? Explain the terms and consider both the system's and the programmer's levels!

1. unidirectional vs. bidirectional communication
2. send by copy vs. send by reference
3. fixed-sized vs. variable-sized messages

Question 8.2: Problems with IPC

1. Why does unsynchronized send in conjunction with non-blocking receive *not* make sense?
2. Asynchronous `send` operations require a buffer for sent but not yet received messages. Discuss possible locations for this message buffer and evaluate them.
3. Consider a system that uses synchronous message passing and timeouts to detect/recover from non-responding communication partners. Discuss why the system designers might choose to provide an atomic `send-and-receive` system call in addition to separate `send` and `receive` calls.

Question 8.3: Emulation using IPC

1. How can you perform asynchronous inter-process communication (IPC) if your operating system only provides synchronous IPC mechanisms?
2. How can you provide synchronous IPC if your operating system only offers asynchronous IPC mechanisms?
3. Find at least three different ways to implement a mutex using synchronous IPC primitives (synchronous send, blocking receive).

Question 8.4: L4 IPC Revisited

L4's fast IPC path allows to transfer short messages in registers, i.e., without copying the message to memory, thus avoiding TLB- and cache-misses.

1. Can you use the fast path on SMP systems? If not, try to find a simple constraint that, when fulfilled, allows to use the fast path on such systems. State the conditions under which slowing down the fast path by checking the additional constraint can be tolerated.
2. L4 IPC implies a thread switch to the receiver, bypassing the scheduler. Point out possible policy violations introduced by this approach and try to justify (or condemn) them.
3. The fast IPC path contains a code fragment as follows in its `send` routine:

```
if (TCB(receiver)->state == WAITING) {
    TCB(receiver)->state = RUNNING;
    SWITCH_TO(receiver);
} else { slowpath(); }
```

On a single-processor system, this code runs atomically with interrupts disabled. Are there any race conditions on SMP systems? How can you avoid them?

Question 8.5: Prerequisites for Deadlocks

What are the necessary conditions for deadlocks? For each condition, give an example of how deadlocks can be prevented by breaking the condition.

Question 8.6: Scheduling Basics

1. What is the purpose of *scheduling*?
2. Enumerate and explain the different scheduling "levels" that were introduced in the lecture.
3. What *quantitative* metrics can be used to estimate the quality of a scheduling policy?