Advanced Operating Systems - Report

Fabian Wüthrich Janis Peyer Kristina Martšenko
Phillipe Mazenauer

Spring 2020

Contents

List of Figures

List of Tables

Chapter 1

Memory management and capabilities

123 [1]

Chapter 2

Processes, threads, and dispatch

TODO: Reason about why we don't handle unaligned addresses in paging. (We want the client to actively think about what they actually map.)

TODO: Explain the changes in the paging interface to the given interface

TODO: Maybe document our difficulties that address to params has to be passed in x0 register

Chapter 3

Message prassing

3.1 channel setup init ¡-¿ child

1. in `create_child_channel` (spawn.c) register `recv_setup_closure` from init (used to receive initial ep from child)
2. in init.c (called before each thread):
   - create child endpoint i.e channel for this child (we are already in child)
   - register `receive_init_closure` (used to save ep received from init)
   - send child ep to init and wait until `receive_init_closure` get called
3. `recv_setup_closure` gets called:
   - save child ep in init channel struct (`lmp_chan`)
   - send message to child that channel is ready (in `rpc_send_setup_closure`
4. `barrelfish_recv_init_closure` gets called on the child side
   - child is now ready to communicate with init
   - on child side continue after recv init success
   - init is now dispatching events in `recv_regular_closure`

Chapter 4

Page fault handling

Chapter 5

Multicore

Chapter 6

User-level message passing

Chapter 7

Individual Project 1

Chapter 8

Individual Project 2

Chapter 9

Individual Project 3

Chapter 10

Individual Project 4

Chapter 11

Summary of overall system

Appendix A

User guide

Bibliography

[1] Timothy Roscoe and the ETH Barrelfish team. *AOS Book*. URL: https://www.systems.ethz.ch/sites/default/files/file/drupal6/courses/2020-spring/aos/book.pdf. (accessed: 14.03.2020).