CHERI
CheriOS Microkernel

Hadrien Barral

University of Cambridge, SRI International

CHERI Microkernel Workshop – 23 April 2016
Overview

• Microkernel based on the CHERI model
• Pure-capability ABI
  • no legacy pointers, no MMU
• Make full use of compartmentalization
• Kernel mode: Context switch
  • IPC
  • Exceptions
Compartments

- Boot
- Context switch
  - IPC
- User program
  - Other service
  - Other service
- Memory allocator
- UART
  - User program
- Scheduler
- Filesystem
  - Other service
- Core service
Boot process

- Context switch
- IPC
- Scheduler
- UART
- Core service
- Init
- Memory allocator
- Filesystem
- User program
- Other service
Inter Process Communication

CCaller « A »

Kernel

Callee « B »

Register

Create task

Call constructor

Context switch

‘O’, method number, method arguments

Context switch

‘O’, method number, method arguments

Get ‘B’ object

Call constructor

CCall

‘B’, method number, method arguments

Execute method

Return value

Return value
IPC (2)

• Callee might **not return** (bug, untrusted, …)
  ➔ Make the call asynchronous

  • **Return to caller**; caller polls the kernel to know if the callee is done

  • Caller gives a **time limit** to the kernel
    • Other kind of limit?

• Communication with the kernel
  • **CCall** or **syscall**?
Memory management

- Revoking a capability is hard

- How can we free memory?
  - Restrict sharing of pointers
  - Garbage-collect
  - MMU
Thank you!

https://github.com/CTSRD-CHERI/cherios