seL4 Libraries:Theory

14 August 2015

Adrian Danis

seL4 Libraries

- Implement standard activities in seL4
  - dealing with initial caps
  - allocating objects
  - managing CSpace, managing VSpace
  - creating and managing processes
- Interfaces vs Implementations
  - *Interface*
    - key datastructs
    - function definitions
    - generic code to facilitate use of interface
  - *Implementation*
    - adds implementation-specific parts to datastructs
    - implements interface functions

Key Interfaces and Libraries

- Key Interfaces
  - simple: access to initial caps
  - vka: virtual kernel allocator
  - vspace: VSpace management
- Key Libraries
  - allocman (vka): allocator manager
  - sel4utils (vspace, io operations): higher level concepts
- Other Libraries
  - muslc, libseL4
  - platsupport, sel4platsupport
  - utils, debug, benchmark

Simple

- Easy way to access initial caps
- Abstracts over spec of initial caps
  - root task: uses bootinfo
  - user-level task: can use bootinfo or some other format
- Key concepts
  - location of resources, caps to resources
  - acquiring resource without cap
- Interfaces
  - simple

Simple: API

- Files
  - libsel4simple, simple.h,
  - libse4simple-default, simple-default.h
- Datastructs
  - simple_t
- Functions
  - simple_default_init_bootinfo
  - simple_print
  - simple_get_*: pd, tcb, cnode, node_size
  - simple_get_nth_untyped
  - simple_get_frame_*: cap, info, vaddr

VKA (virtual kernel allocator)

- Interface for allocating kernel objects
  - abstracts away
    - creation of objects through retyping untypes
    - managing CSpace and book keeping
  - requires an underlying implementation
- Key concepts
  - vka: allocator
  - objects
  - CSpace slots
  - utspace: pool of untyped memory
  - cspace path: fully qualified capability address

VKA: API

- Files
  - libsel4vka, vka.h
- Datastructs
  - vka_t
  - vka_object_t
    - cptr, cookie, object type, size
- Functions
  - vka_alloc_*: pd, cnode, tcb, endpoint. returns vka_object_t
  - vka_cspace_alloc: allocate an empty slot in the CSpace
  - vka_cspace_free: doesn't delete object!
  - vka_utspace_alloc: given empty slot create an object and put the cap to it in the slot. returns cookie
  - vka_utspace_free: given cookie, free object

Allocman

- Allocator Manager
  - implements vka
  - framework combining independent CSpace and utspace allocators
  - solves difficult recursion problems in allocation: Black Magic!
- Key concepts
  - resources: need to provide underlying resources (e.g. untypeds)
  - memory pool: for internal allocations
- Interfaces
  - vka
  - allocman: to add resources after initialisation

Allocman: API <TODO>

- Files:
  - libsel4allocman, allocman.h, vka.h, bootstrap.h
- Datastructs
  - allocman_t
- Functions
  - bootstrap
    - bootstrap_use_current_simple
    - bootstrap_new_2level_simple
  - allocman_make_vka

VSpace

- Interface for managing VSpaces
  - manage current VSpace
  - manage other VSpaces
    - note: create is not part of vspace API
  - allocate frames and map them into a VSpace
- Key concepts
  - reservation: portion of VSpace, that will not be given to others
  - mapping: frame in a VSpace

VSpace: API

- Files
  - libsel4vspace, vspace.h
- Datastructs
  - vspace_t
  - reservation_t
- Functions
  - vspace_reserve_range, vspace_free_reservation
  - vspace_new_pages
  - vspace_map_pages
  - vspace_unmap_pages: different ways to free frame object
  - vspace_get_cap: get frame cap for virtual address
  - vspace_get_root

seL4utils

- Utility code to make life easier
  - create and manage threads and processes
  - create vspaces, implement vspace interface
  - load ELF code
- Key concepts
  - process: CSpace + VSpace + TCB
- Interfaces
  - vspace

seL4utils: API

- Files:
  - libseL4utils, vspace.h, process.h, mapping.h
- Datastructs
  - process_t
  - thread_t
- Functions
  - sel4utils_bootstrap_vspace_with_bootinfo
  - sel4utils_get_vspace: create new vspace
  - sel4utils_configure_process, sel4utils_spawn_process_v
  - sel4utils_*_cap_to_process: mint, copy
  - seL4_ARCH_*: abstract architecture dependent seL4 syscalls

Platsupport

- Hardware Access
  - seL4 independent hardware access
  - device driver code
  - I/O interfaces:
    - allocation MMIO, DMA memory
    - IOport operations
- Key concepts
  - arch (x86, arm), mach (common superset of platforms), plat (specific device)
  - device initialisation, handle interrupts, driver interface
- Interfaces
  - I/O interfaces
  - device-class specific interfaces

Platsupport: APIs

- Files
  - libplatsupport: io.h, timer.h,
  - includes in arch, mach, plat
- General Datastructs
  - pstimer: platsupport timer
  - clock_sys_t: for ARM device clocks
  - ps_io_ops: wrapper combining key platsupport interfaces
- Driver-specific Interfaces
  - look them up yourself :-)

seL4 Platsupport

- seL4 wrappers for Platsupport
  - uses simple, vka, vspace to access hardware
- Files
  - libsel4platsupport: platsupport.h, io.h, timer.h
- Datastructs
  - sel4_timer_t
- Functions
  - platsupport_serial_setup_simple
  - sel4platsupport_get_default_timer
  - sel4platsupport_new_io_*: mapper, ops

Dependencies

- simple: libsel4
- vka: libsel4, utils
- allocman: vka, libsel4, sel4utils, vspace, utils
- vspace: vka, libsel4, utils
- sel4utils: simple, vka, vspace, platsupport, sel4utils, utils, elf, cpio
- platsupport: utils
- sel4platsupport: simple, vka, vspace, platsupport, sel4utils, utils

CapDL

- Capability Distribution Language
  - describes of a *Capability Distribution*
    - all the kernel objects
    - how capabilities to those objects are distributed and mapped: e.g. in CNodes, PageTables, TCBs, etc.
  - specifies a desired or existing system
- CapDL loader
  - root task:
    - given a capDL spec, creates the desired objects and capability distribution