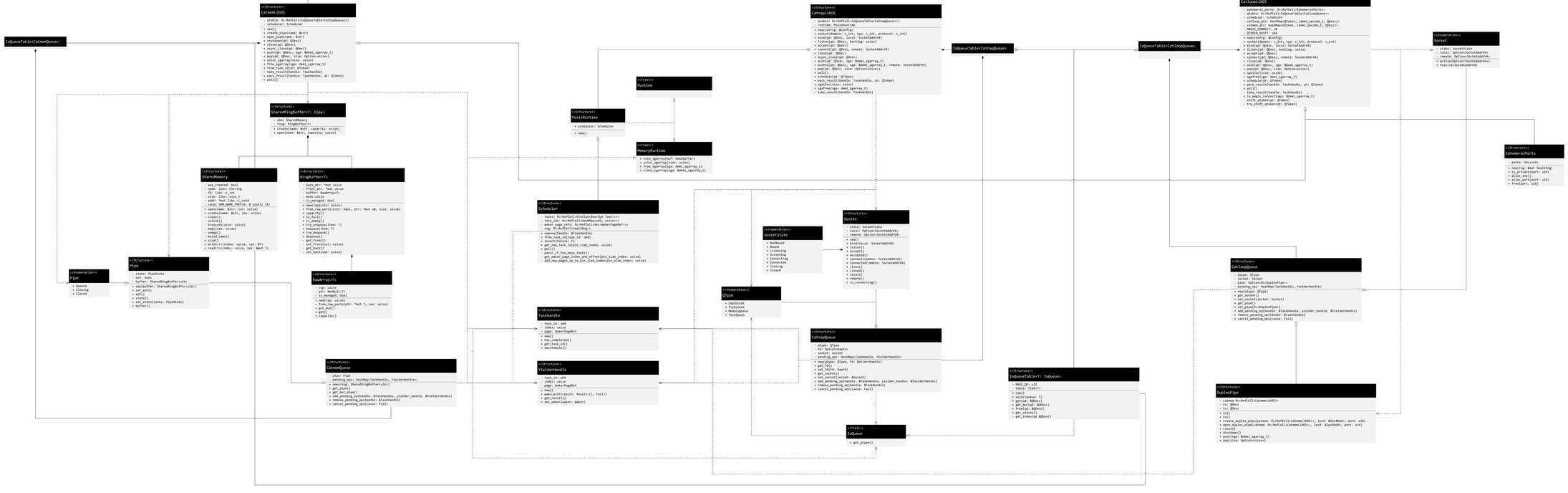
IoQueueTable<CatmemQueue>
<<Structure>>
CatmemLIOS
- global: RcRefCell<IoQueueTable<CatnapQueue>>
- scheduler: Scheduler
+ new()
+ create_pipe(name: &str)
+ open_pipe(name: &str)
+ shutdown(qd: QDesc)
+ close(qd: QDesc)
+ accept(qd: QDesc, qt: QToken)
+ push(qd: QDesc, sga: &dmi_sgarray_t)
+ pop(qd: QDesc, size: Option<usize>)
+ sgaalloc(size: usize)
+ free_sgarray(sga: dmi_sgarray_t)
+ from_task_id(qd: QToken)
- take_result(handle: TaskHandle)
+ pack_result(handle: TaskHandle, qt: QToken)
+ poll()
<<Structure>>
SharedRingBufferT: Copy>
- shm: SharedMemory
- ring: RingBufferT>
+ create(name: &str, capacity: usize)
+ open(name: &str, capacity: usize)
<<Structure>>
SharedMemory
- was_created: bool
- name: CStrRefOwning
- fd: libc::c_int
- size: libc::size_t
- ptr: *mut MemoryRegion
- const SHM_NAME_PREFIX: &'static str
+ open(name: &str, len: usize)
+ create(name: &str, len: usize)
+ close()
+ unlink()
+ truncate(size: usize)
+ mapsize: usize)
+ unmap()
+ build_name()
+ as_ptr()
+ write<T>(index: usize, val: &T)
+ read<T>(index: usize, val: &mut T)
<<Structure>>
RingBufferT>
- back_ptr: *mut usize
- front_ptr: *mut usize
- buffer: RawArray<T>
- mask:usize
- is_managed: bool
+ new(capacity: usize)
+ from_raw_parts(init: bool, ptr: *mut u8, size: usize)
+ capacity()
+ is_full()
+ is_empty()
+ try_enqueue(item: T)
+ enqueue(item: T)
+ try_dequeue()
+ dequeue()
+ peek_front()
- set_front(val: usize)
- get_back()
- set_back(val: usize)
<<Enumerations>>
Pipe
+ Queue
+ Closing
+ Closed
<<Structure>>
Pipe
- state: PipeState
- eof: bool
- buffer: SharedRingBufferu16>
+ new(buffer: SharedRingBufferu16>)
+ as_ptr()
+ eof()
+ state()
+ set_state(state: PipeState)
+ buffer()
<<Structure>>
RawArray<T>
- cap: usize
- ptr: NonNull<T>
- is_managed: bool
+ new(capacity: usize)
+ from_raw_parts(ptr: *mut T, len: usize)
+ get_mut()
+ get()
+ capacity()
<<Structure>>
CatmemQueue
- qtype: QType
- pending_ops: HashMap<TaskHandle, YielderHandle>
+ new(ring: SharedRingBufferu16>)
+ get_pipe()
+ get_mut_pipe()
+ add_pending_op(handle: &TaskHandle, yielder_handle: &YielderHandle)
+ remove_pending_op(handle: &TaskHandle)
+ cancel_pending_ops(cause: Fail)
<<Structure>>
Scheduler
- tasks: RcRefCell<InlinedArrayBoxed<dyn Task>>
- waker_page_refs: RcRefCell<HashMap<usize, usize>>
- waker_page_pool: RcRefCell<crossbeam/PagePool>>
- rng: RcRefCell<SmallRng>>
+ remove(handle: &TaskHandle)
+ from_task_id(task_id: u64)
+ insert(future: F)
+ get_new_task_id(pin_slab_index: usize)
+ poll()
- panic!_if_too_many_tasks()
- get_waker_page_index_and_offset(pin_slab_index: usize)
- add_new_pages_up_to_pin_slab_index(pin_slab_index: usize)
<<Structure>>
TaskHandle
- task_id: u64
- index: usize
- page: WakerPageRef
+ new()
+ has_completed()
+ get_task_id()
+ deschedule()
<<Structure>>
YielderHandle
- task_id: u64
- index: usize
- page: WakerPageRef
+ new()
+ wake_with(result: Result<(), Fail>)
+ get_result()
+ set_waker(waker: Waker)
<<Structure>>
CatnapLIOS
- global: RcRefCell<IoQueueTable<CatnapQueue>>
- runtime: PosixRuntime
- scheduler: Scheduler
+ socket(domain: c_int, typ: c_int, protocol: c_int)
+ bind(qd: QDesc, local: SocketAddrV4)
+ listen(qd: QDesc, backlog: usize)
+ accept(qd: QDesc)
+ connect(qd: QDesc, remote: SocketAddrV4)
+ close(qd: QDesc)
+ async_close(qd: QDesc)
+ push(qd: QDesc, sga: &dmi_sgarray_t)
+ pushto(qd: QDesc, sga: &dmi_sgarray_t, remote: SocketAddrV4)
+ pop(qd: QDesc, size: Option<usize>)
+ poll()
+ schedule(qt: QToken)
+ pack_result(handle: TaskHandle, qt: QToken)
+ sgaalloc(size: usize)
+ free_sgarray(sga: dmi_sgarray_t)
- take_result(handle: TaskHandle)
<<Trait>>
Runtime
<<Structure>>
PosixRuntime
+ scheduler: Scheduler
+ new()
<<Trait>>
MemoryRuntime
+ into_sgarray(buf: DemiBuffer)
+ alloc_sgarray(size: usize)
+ free_sgarray(sga: dmi_sgarray_t)
+ clone_sgarray(sga: &dmi_sgarray_t)
<<Enumerations>>
SocketState
+ NotBound
+ Bound
+ Listening
+ Accepting
+ Connecting
+ Connected
+ Closing
+ Closed
<<Structure>>
Socket
- state: SocketState
- local: Option<SocketAddrV4>
- remote: Option<SocketAddrV4>
+ new()
+ bind(local: SocketAddrV4)
+ listen()
+ accept()
+ connect()
+ connect(remote: SocketAddrV4)
+ connected(remote: SocketAddrV4)
+ close()
+ local()
+ local()
+ remote()
+ is_connecting()
<<Enumerations>>
QType
+ IoQueue
+ TcpSocket
+ MemoryQueue
+ TestQueue
<<Structure>>
CatnapQueue
- qtype: QType
- fd: Option<Rawfd>
- socket: Socket
- pending_ops: HashMap<TaskHandle, YielderHandle>
+ new(qtype: QType, fd: Option<Rawfd>)
+ get_qtype()
+ get_fd(): Rawfd)
+ get_socket()
+ set_socket(socket: &Socket)
+ add_pending_op(handle: &TaskHandle, yielder_handle: &YielderHandle)
+ remove_pending_op(handle: &TaskHandle)
+ cancel_pending_ops(cause: Fail)
<<Trait>>
IoQueue
+ get_qtype()
IoQueueTable<CatnapQueue>
IoQueueTable<CatloopQueue>
<<Structure>>
IoQueueTable<: IoQueue>
- MAX_QD: u32
- table: Slab<T>
+ new()
+ alloc(queue: T)
+ free(qd: QDesc)
+ get_mut(qd: &QDesc)
+ free(qd: &QDesc)
+ get_values()
- get_index(qd &QDesc)
<<Structure>>
CatloopLIOS
- ephemeral_ports: RcRefCell<EphemeralPorts>>
- qtable: RcRefCell<IoQueueTable<CatloopQueue>>
- scheduler: Scheduler
- rating_gts: HashMap<Qtoken, (dmi_opcode_t, QDesc)>
- rameme_gts: HashMap<Qtoken, (dmi_opcode_t, QDesc)>
- qtokens: Vec<Qtoken>
- QTOKEN_SHIFT: u64
+ new(config: &Config)
+ socket(domain: c_int, typ: c_int, protocol: c_int)
+ bind(qd: QDesc, local: SocketAddrV4)
+ listen(qd: QDesc, backlog: usize)
+ accept(qd: QDesc)
+ connect(qd: QDesc, remote: SocketAddrV4)
+ close(qd: QDesc)
+ push(qd: QDesc, sga: &dmi_sgarray_t)
+ pop(qd: QDesc, size: Option<usize>)
+ sgaalloc(size: usize)
+ sgafree(sga: dmi_sgarray_t)
+ schedule(qt: QToken)
+ pack_result(handle: TaskHandle, qt: QToken)
- take_result(handle: TaskHandle)
+ is_magic_connect(sga: &dmi_sgarray_t)
- shift_ntohs(qt: QToken)
- try_shift_qtokens(qt: QToken)
<<Enumeration>>
Socket
- state: SocketState
- local: Option<SocketAddrV4>
- remote: Option<SocketAddrV4>
+ Active(Option<SocketAddrV4>)
+ Passive(SocketAddrV4)
<<Structure>>
EphemeralPorts
- ports: Vec<u16>
+ new(rng: &mut SmallRng)
+ is_private(port: u16)
+ alloc_any()
+ alloc_port(port: u16)
+ free(port: u16)
<<Structure>>
CatloopQueue
- qtype: QType
- socket: Socket
- pipe: Option<RcDuplexPipe>>
- pending_ops: HashMap<TaskHandle, YielderHandle>
+ new(qtype: QType)
+ get_socket()
+ set_socket(socket: Socket)
+ get_pipe()
+ set_pipe(Rc<DuplexPipe>)
+ add_pending_op(handle: &TaskHandle, yielder_handle: &YielderHandle)
+ remove_pending_op(handle: &TaskHandle)
+ cancel_pending_ops(cause: Fail)
<<Structure>>
DuplexPipe
- catmem Rc<RefCell<CatmemLIOS>>
- rx: QDesc
- tx: QDesc
+ tx()
+ rx()
+ create_duplex_pipe(catmem: Rc<RefCell<CatmemLIOS>>, ipv4: &Ipv4Addr, port: u16)
+ open_duplex_pipe(catmem: Rc<RefCell<CatmemLIOS>>, ipv4: &Ipv4Addr, port: u16)
+ close()
+ shutdown()
+ push(sga: &dmi_sgarray_t)
+ pop(size: Option<usize>)