Mach Microkernel Internals

Internal Data Structures

Correspond to Externally Visible Objects

Implementation of Interfaces

Remember: Mach is NOT U*X!

Externally Visible Objects

Tasks

Threads

Ports

Message Queues

Memory Objects

Processor Sets

Processors

Devices

Each has a corresponding internal object.

Common facilities:

- Access to Object via Port
- Reference Counting - Garbage Collection
- Locking - Hierarchy
- Statistics

Tasks

IPC Space

Address Space

List of Threads

Exception Handler

System Call Emulation Vector

Scheduling Control

Initial Ports

In: Processor Set

Threads

Machine State

Scheduling Control

IPC Status and Wait Links

Exception Handler

Timing

Initial Ports

In: Task, Processor Set

IPC

IPC Space

- Translations: (space, name) -> Port
- (space, port) -> Name
- Port Rights

Ports

- Receiving Task
- Message Queue
- Interested Tasks: Dead-Name Requests
- In: IPC Space

Address Space

Virtual Address Map (*vm_map_t*)

- Linear List of Regions
- Hint for quick search
- Hint for quick insertion
- Each region points to a Memory Cache Object (*vm_object_t*)
- Physical Map - Machine Dependent

Map Copy Object (*vm_map_copy_t*)

- List of Regions
- No Lookup Necessary
- Optimized for Message Passing, IO

Memory Cache Object

Internal Memory Object (*vm_object*)

Referenced by Many Map Regions - for copy-on-write, sharing

- Physical Page List
- External Memory Object Port

Currently unreferenced Memory Cache Objects may be held in object cache, to be used later

Physical Resources - Processors

Processor Set

- Run Queue
- Idle Processors
- Assigned Tasks and Threads
- Scheduling Control

Processor

- Local Run Queue - for Bound Threads
- Current Time Quantum
- In: Processor Set

Global list of all processor sets - can find all tasks and threads from these.

Internal Threads

Idle - Per Processor

Reaper - Garbage Collector for Dead Threads

Swapin - Allocates Kernel Stacks

Sched - Adjusts Priorities

Pageout - Recycles Physical Memory

Action - Reassigns Processors

IO Completion

Entry Points to Kernel from User Threads

- Page Faults
- Exceptions
- Rescheduling
- IPC entry - mach_msg_trap

User thread enters kernel and runs on kernel stack

User state saved in "pcb" (machine state)

Kernel state saved on kernel stack or in continuation

Calling Kernel Services

Abstract

- Kernel services are the same as user services: client sends a message to server and waits for reply.

Implementation

- Kernel services are directly invoked from mach_msg_trap, and run in user thread.
- Use Kernel address space, Global port space
- Must preserve IPC semantics - must execute as if kernel were a separate service with separate threads
- Must distinguish current thread and task (invoker) from thread or task to be controlled (object)
- May have to avoid deadlocks if operating on current thread: terminate, suspend, abort.

Other Kernel Components

Timers

- Queue of events to occur at specific times.
- Timeouts for IPC, user thread rescheduling, and devices.

Zone Memory Allocator

- Zone contains items of same size, for quick allocation and deallocation

Lock Package

- Simple (spin) Locks
- Complex (sleep, reader/writer) Locks

Interrupt Priority Levels

Reference Counts and Locks

Reference-counted Garbage Collection

- Explicit calls to increment and decrement reference count
- References are rooted in permanent objects (host, processors) or in currently executing threads.
- Some data structures have uncounted references. These are guaranteed never to be the only reference to an object.

Locking and Referencing Hierarchy:

- Avoids Deadlocks
- Ensures that Referenced Item Exists
- Locks and References may be acquired only by descending hierarchy
- May be released in any order

Explicit Termination

Explicit Termination implies that an object may start disappearing while an operation is in progress.

Reference Count prevents dangling pointers

Active flag prevents further operations once termination has started