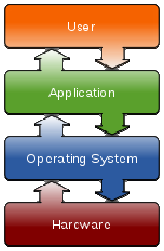
ENCE360 Operating Systems



OS X

'X' is also used to emphasize the relatedness between OS X and UNIX

iOS

runs on the iPhone, iPod Touch, iPad, and Apple TV,
shares the Darwin core and many frameworks with OS X

OS X
Overview

- History & Background
- Processor Modes & Process States
- Memory Management
- Threads
- File Management
- Scheduling

OS X
History

1990

1994 – Apple announces the beginning of Copland

1996 – Apple abandons Copland project

1996 – Apple purchases NeXT for $430 million

2001 – Apple releases Mac OS X

September 2001 – Apple releases Mac OS 10.1

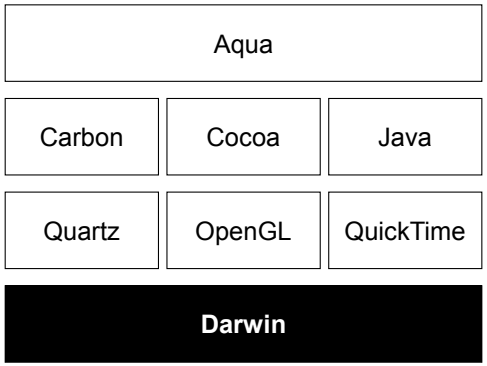
Mac OS X 10.7 "Lion“ runs exclusively on 64-bit Intel CPUs

OS X 10.10 Yosemite

2015

OS X
Background

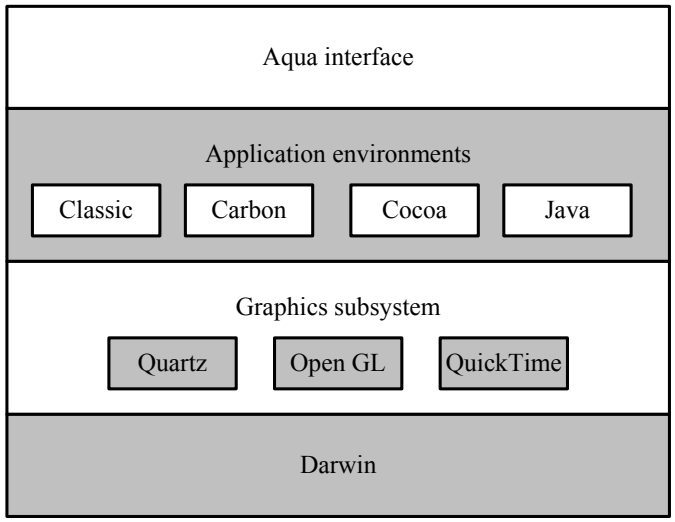
- **Darwin**
  - Open Source
  - BSD core services



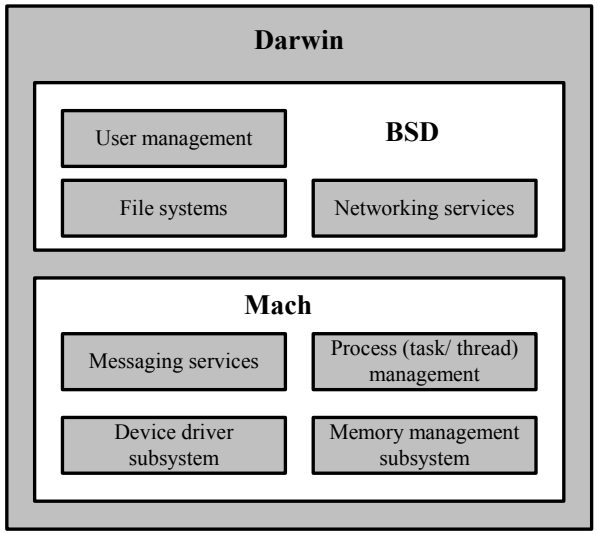
iOS

- runs on the iPhone, iPod Touch, iPad, and Apple TV, shares the Darwin core and many frameworks with OS X

OS X overview



Darwin, the OS X kernel



Mach

- Mach manages processor resources such as CPU usage, handles scheduling, provides memory protection, and provides a messaging-centered infrastructure to the rest of the operating-system layers.

- The Mach component provides:
  - Inter-process communication (IPC)
  - remote procedure calls (RPC)
  - scheduler support for symmetric multiprocessing (SMP)
  - support for real-time services
  - virtual memory support

BSD

- Above the Mach layer, the BSD layer provides advanced networking, security, and compatibility features

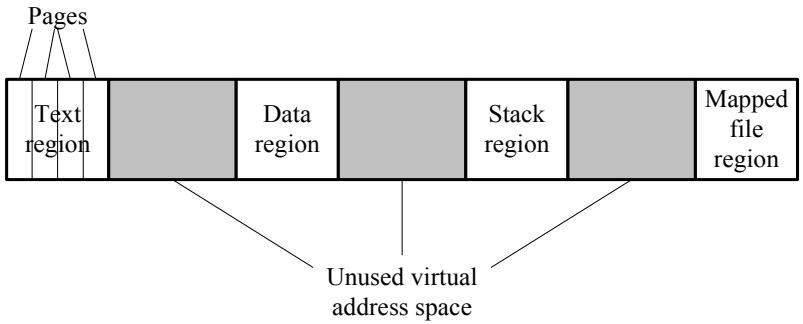
- The BSD component provides
  - File systems
  - networking services
  - UNIX security model (user IDs and Permissions)
  - BSD Application Program Interfaces (APIs)
  - The system framework for exporting APIs to the application layers
  - Multi-user Access

Mach Scheduling

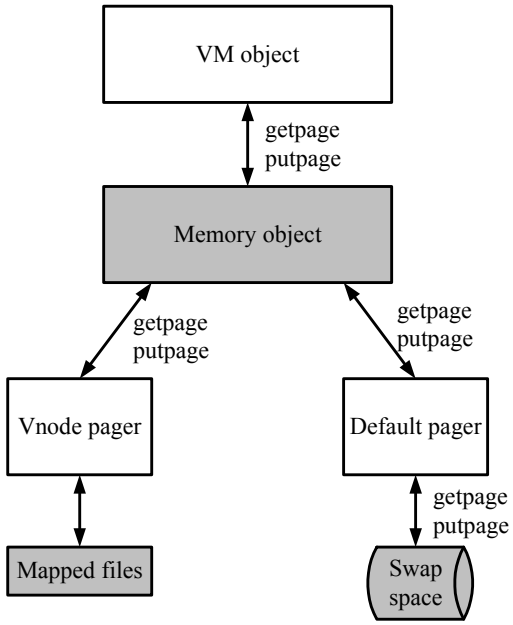
Mach scheduling is based on a system of run queues at various priorities that are handled in different ways. The priority levels are divided into four bands according to their characteristics.

Priority Bands	Characteristics
Normal	Normal application threads
System high priority	Threads whose priority has been raised above normal
Kernel mode threads	Threads created in the kernel that run at a higher priority than all user space threads.
Real-time threads	Threads that need a significant fraction of available clock cycles

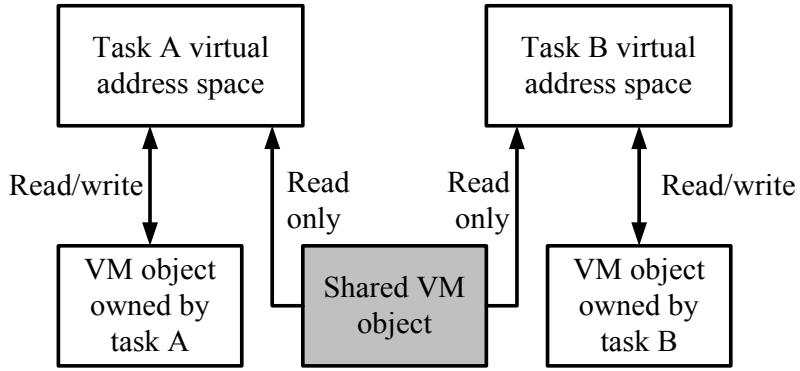
Virtual address space.



VM objects and pagers.

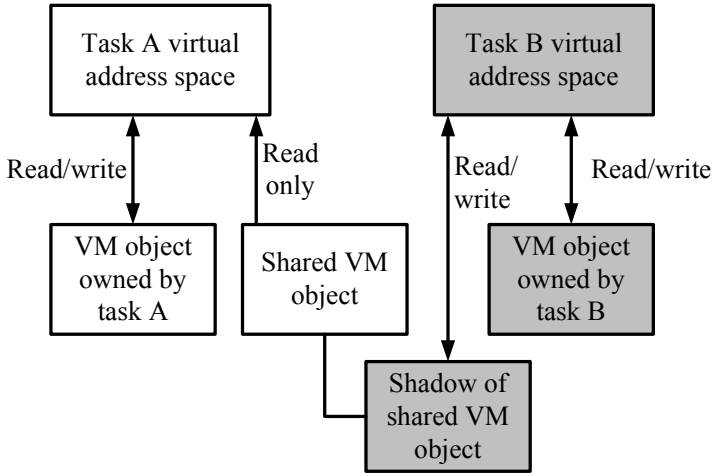


Copy on write.

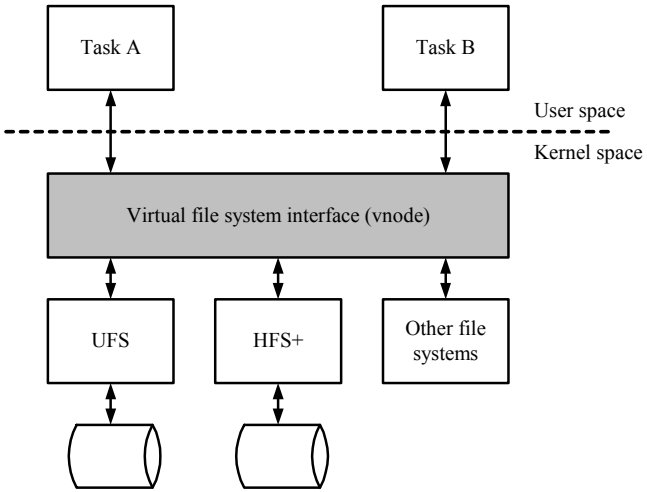


a. When a new task is created, it is cloned (or copied) from a parent.

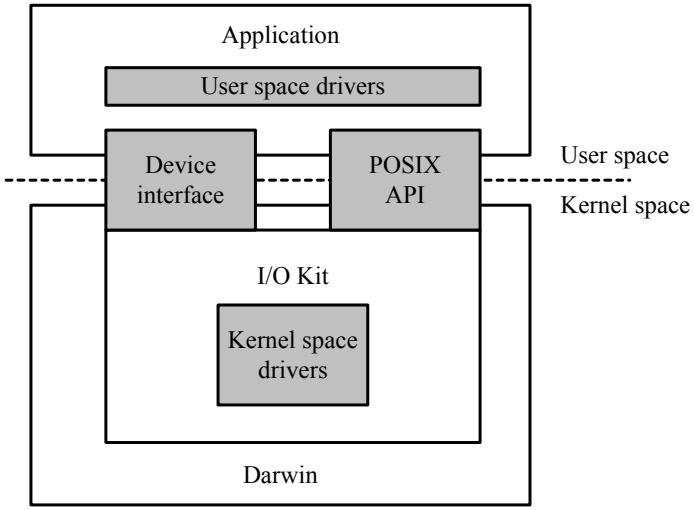
If a task modifies a portion of shared memory, OS X makes another copy and gives the task write access.



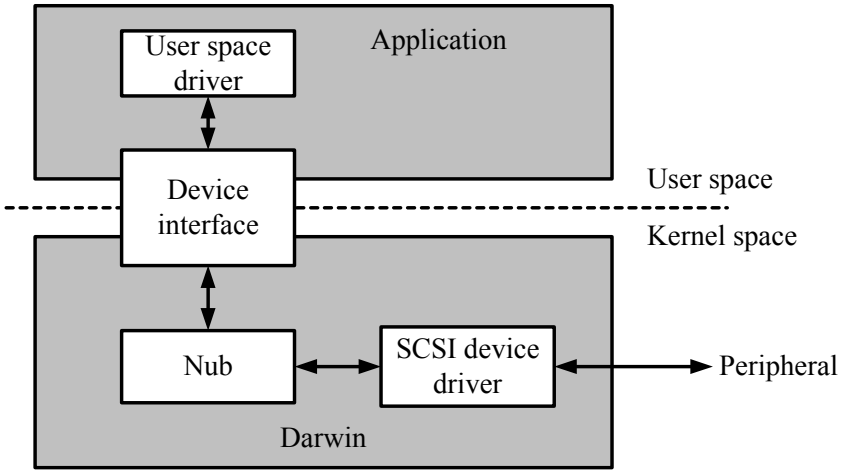
The virtual file system (VFS).



OS X supports user-space and kernel-space device drivers.



Accessing a device through the device interface.



OS X
Processor Modes & Privileged Instructions

- Supervisor Mode – allows execution of privileged instructions and access to privileged registers

  Swapper (PID = 0) – switches processes from main memory to secondary storage

  PageDaemon (PID = 2) – converts addresses to support the virtual memory subsystem

- User Mode - The processor mode that forbids execution of privileged instructions and access to privileged registers. Any attempt to do so will result in a privilege violation exception.

  Init (PID = 1) spawns processes for remote terminals; changes run levels; can assume role of parent for orphaned process

OS X
Allowable Process States

Process states for FreeBSD

- SIDL: Process is partially created
- SRUN: Process is runnable
- SSLEEP: Process is awaiting event
- SSTOP: Process is stopped (by signal or parent process)
- SZOMB: Process is partially terminated (waiting for parent process to collect status)

OS X
Memory Management

5 Basic OS X Abstractions

- Task
- Thread
- Port
- Message
- Memory Object

Basic VM Operations

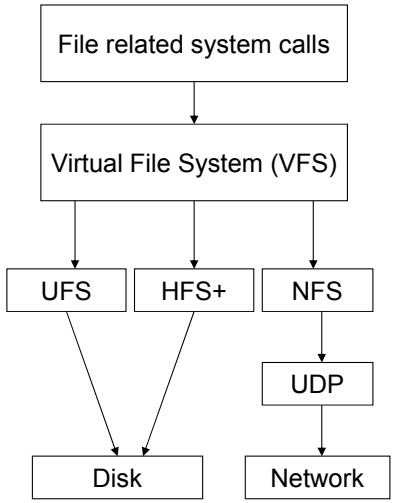
- allocate a region of virtual memory on a page boundary,
- deallocate a region of virtual memory,
- set the protection status of a region of virtual memory,
- specify the inheritance of a region of virtual memory and
- create and manage a memory object that can then be mapped into the address space of another task.

OS X
Memory Management (cont.)

Four basic memory management data structures are used in OS X:

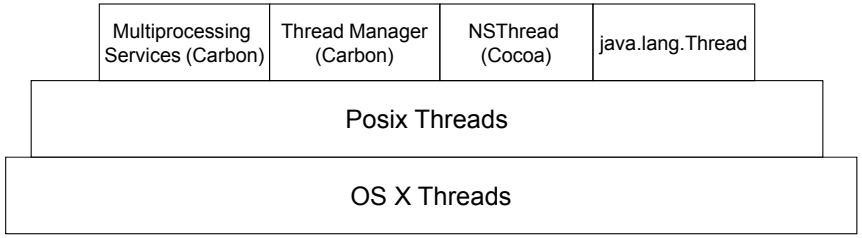
1. the resident page table - a table used to keep track of information about machine independent pages
2. the address map - a doubly linked list of map entries, each of which describes a mapping from a range of addresses to a region of a memory object
3. the memory object - a unit of backing storage managed by the kernel or a user task and
4. the pmap - a machine dependent memory mapping data structure (i.e., a hardware defined physical address map).

OS X
File Management



The Darwin kernel implements a Virtual File System (VFS) that translates a file-related system call into the matching call for the appropriate file system.

OS X
Threads



- Threads are scheduled to run preemptively or, with symmetric multiprocessing, concurrently. Threading models built on top of the kernel's can, however, use various synchronization mechanisms to present cooperative threading behavior.
- The kernel environment of OS X, provides the fundamental thread support. OS X maintains the register state of its threads and schedules them preemptively in relation to one another. In the case of symmetric multiprocessing, the kernel can preemptively schedule threads concurrently, one on each processor. The client API for OS X threads is implemented in the System framework.

OS X
Scheduling

- Marriage of FreeBSD and OS X
- Uses a priority scheduling algorithm combined with a time quantum
- Priorities are stored in the PCB and range from -20 to 20 with higher values indicating lower level priority
- "Pzero" refers to a priority variable of 0
- Reschedules every tenth of a second and recomputes priorities once every second

OS X
Scheduling

- No preemption in OS X scheduling
- Task will relinquish CPU after time quantum or when it must wait for an I/O completion
- Process feedback – priorities change dynamically based on wait time and amount of time that the process has had the processor
- Feedback prevents starvation

OS X

Future Outlook

“We are betting our future on OS X.” –Steve Jobs

Avie Tevanian, Chief Software Technology Officer at Apple, explained OS X was the platform of the future: “We built OS X hoping that it will last 15 to 20 years. I hope it does,” he joked. “I can’t go through this again.”