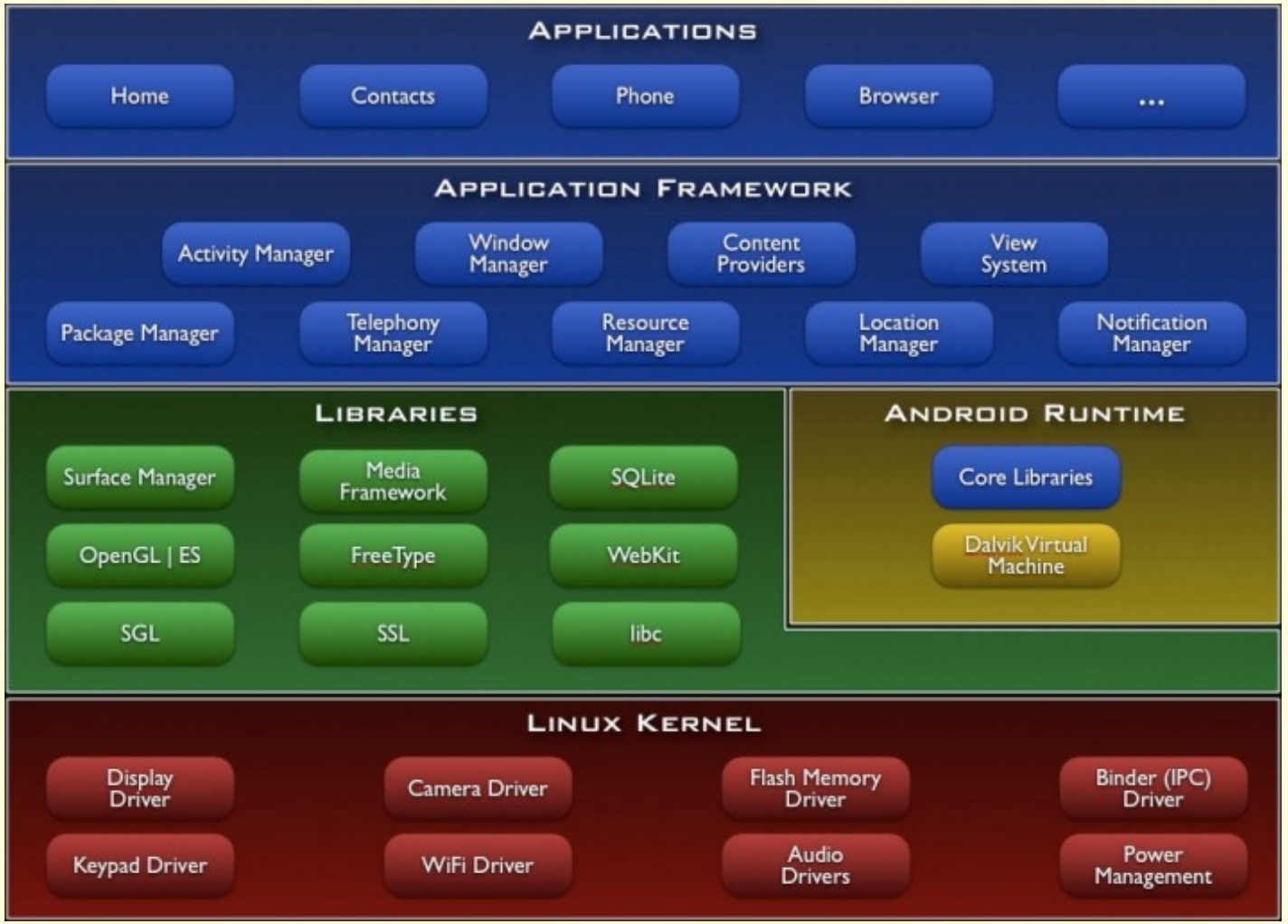
75-08 Sistemas Operativos
Lic. Ing. Osvaldo Clúa
2011

Facultad de Ingeniería
Universidad de Buenos Aires

Procesos en Mobile Os

Android


Applications
Home
Contacts
Phone
Browser
...
Application Framework
Activity Manager
Window Manager
Content Providers
View System
Package Manager
Telephony Manager
Resource Manager
Location Manager
Notification Manager
Libraries
Surface Manager
Media Framework
SQLite
OpenGL | ES
FreeType
WebKit
SGL
SSL
libc
Android Runtime
Core Libraries
Dalvik Virtual Machine
Linux Kernel
Display Driver
Camera Driver
Flash Memory Driver
Binder (IPC) Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management

Historia de Android Inc

- Fundada en Palo Alto, California en 2003
  - Andy Rubin (Danger),
  - Rich Miner (Wildfire Communications),
  - Nick Sears (VP de T-Mobile),
  - Chris White (diseñador en WebTV)
- Adquirida por Google en 2005
- El primer producto Android fue desarrollado por la Open Handset Alliance

Historia

- Las versiones finales son Open Source, no así las de desarrollo
  - 2.0-2.1 Eclair
  - 2.2 Froyo
  - 2.3 Gingerbread
  - 3.0 -3.2 Honeycomb (Tablet)
  - 4.0 Ice Cream Sandwich

Generalidades

- Basado en el Kernel 2.6 de Linux.
  - Aseguran que en unos años se unirán ambos forks.
  - No usa X.
- Usa una JVM llamada Dalvik
  - No corre los bytecodes .class sino .dex
    - Algunos .class pueden convertirse usando dx

Aplicaciones Android

- Vienen empaquetadas en un .apk
  - Una vez instalada tiene su sandbox, cada apk es un usuario de linux con permisos y directorio propios.
- Una Aplicación tien cuatro componentes
  - Activities, Services, Content providers y Boradcast Receivers
- Se activan con un mensaje llamado Intent

Aplicaciones Android

- Los componentes están descriptos en un Manifest (XML).
  - Usa recursos (ej: iconos) a los que el sistema les da un resourceId.
  - Hay mecanismos para compartir datos entre distintas aplicaciones.
  - Hay una previsión para notificaciones asincrónicas

Aplicaciones y Activities

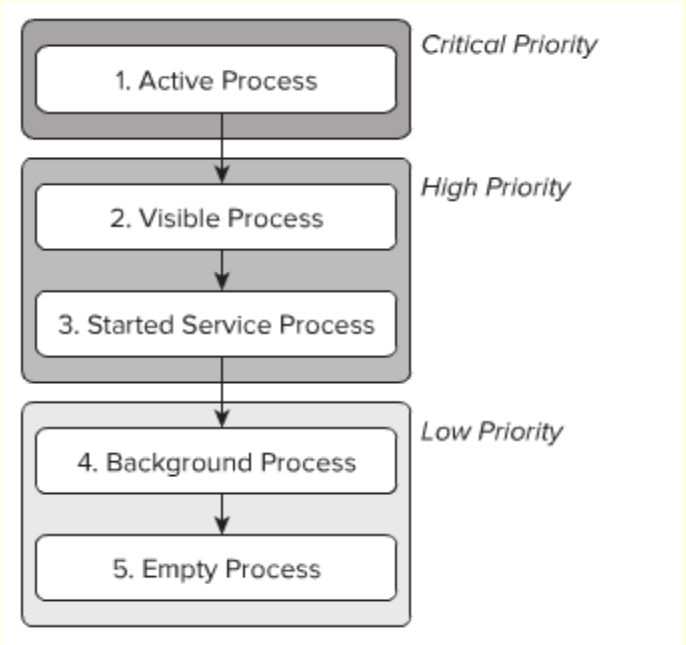
- Activity es una aplicación que se comunica por medio de una pantalla con el usuario.
  - Generalmente full-screen pero puede usar una pantalla flotante.
  - Una Aplicación consiste de una o mas activities
  - Solo una está activa, el resto se guarda en un stack (que se recorre con la tecla Back)

Activitys y Procesos

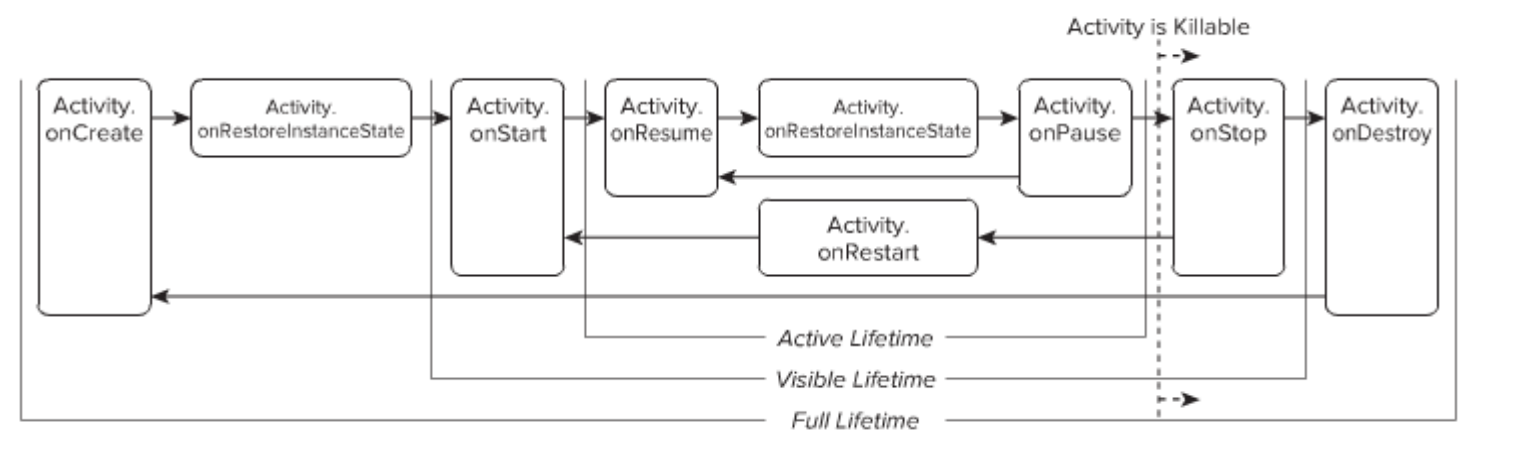
- Cada aplicación corre en su propio proceso con su propia copia de Dalvik.
  - Los procesos son provistos por el kernel y manejados por el Android Run Time (Dalvik + bibliotecas).
- Para mantener la respuesta del sistema, Android puede "matar" sin aviso procesos (y las aplicaciones contenidas).

Prioridad de una Activity

- Se determina por su proceso.
  - O por la de su interlocutor.
- El programador extiende la class Activity
  - Y programa los eventos a los que responde.



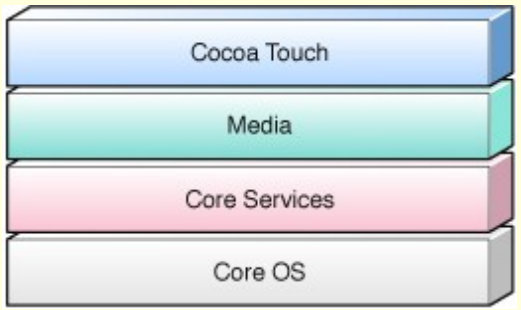
Ciclo de Vida de una Activity



- *Active*: está al tope del stack e interactuando con el usuario.
- *Paused:* Visible pero sin foco.
- *Stopped:* Queda en memoria pero ya terminó. Candidata al *kill.*
- Inactive: Fuera de la memoria. Debe lanzarse nuevamente.

Apple IOS

- No confundir con Cisco/IOS.
- Sistema operativo de iPhone, iPod touch e iPad
  - Basado en Darwin
  - Presenta 4 capas de abstracción

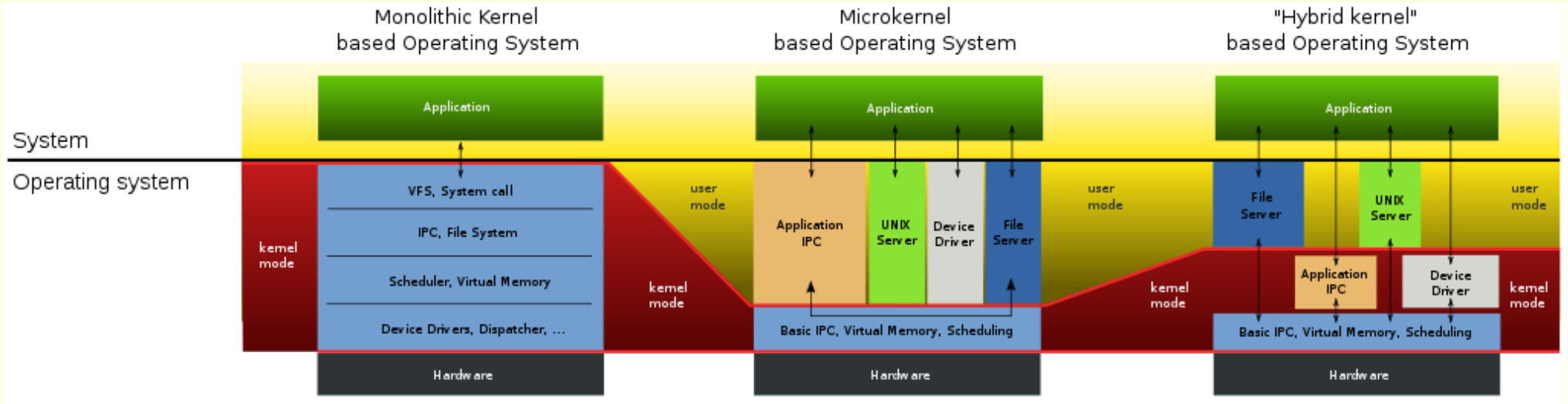


Darwin/XNU

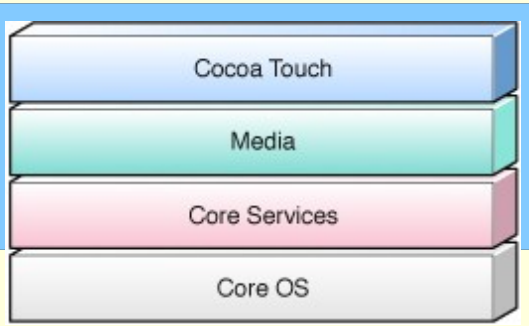
- X is Not Unix
- Desarrollado para la NexTSTEP.
- Un híbrido de Mach 3 y de 4.3BSD.
  - Tiene un componente OO para escribir drivers, el IO Kit.
- Es la base de todos los Sistemas Operativos de Mac (ARM, IA-34, X86-64).

Kernel Híbrido

- Con características de los sistemas microkernel y de monolíticos.
  - Pasaje de mensajes y protección de memoria.



IOS4

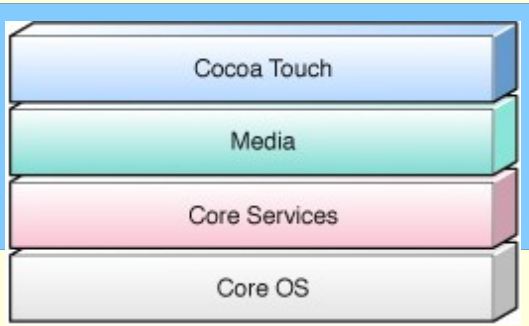


- Cocoa Touch
  - Es el framework para el desarrollo de aplicaciones
    - Multitasking y Printing
    - Data Protection (encriptado)
    - Push y Local Notifications
    - File Sharing (via iTunes) y p2p (juegos)
    - Reconocimiento de gestos y display externo
    - MVC standards

IOS Multitasking

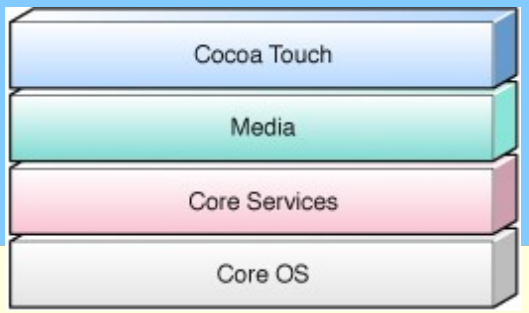
- Versión 4.3, habilitada por API para:
  - Background audio
  - Voice over IP
  - Background location
  - Push notifications
  - Local notifications
  - Task finishing
- Fast app switching

IOS4



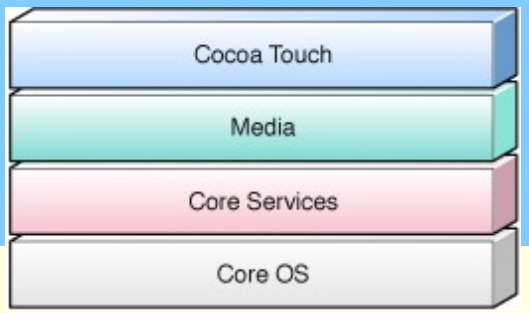
- Media Layer
  - Manejo de Audio, Video y Gráficos.
    - Audio AAC, Apple Lossless (ALAC), A-law, IMA/ADPCM (IMA4), Linear PCM, μ-law, DVI/Intel IMA ADPCM, Microsoft GSM 6.10, AES3-2003.
    - Video .mov, .mp4, .m4v, .3gp
      - H264, Mpeg4
    - Airplay, midi, Quartz, OpenAL y OpenGL.

IOS4



- Core Servicves Layer
  - Servicios fundamentales usados pro las aplicaciones
    - Bloqueo de Objetos, Grand Central Dispatch, SQLite, XML, InAppPurchase
  - Core Frameworks
    - Foundation,. AddressBook, Location, TE, Eventos, Store

IOS4

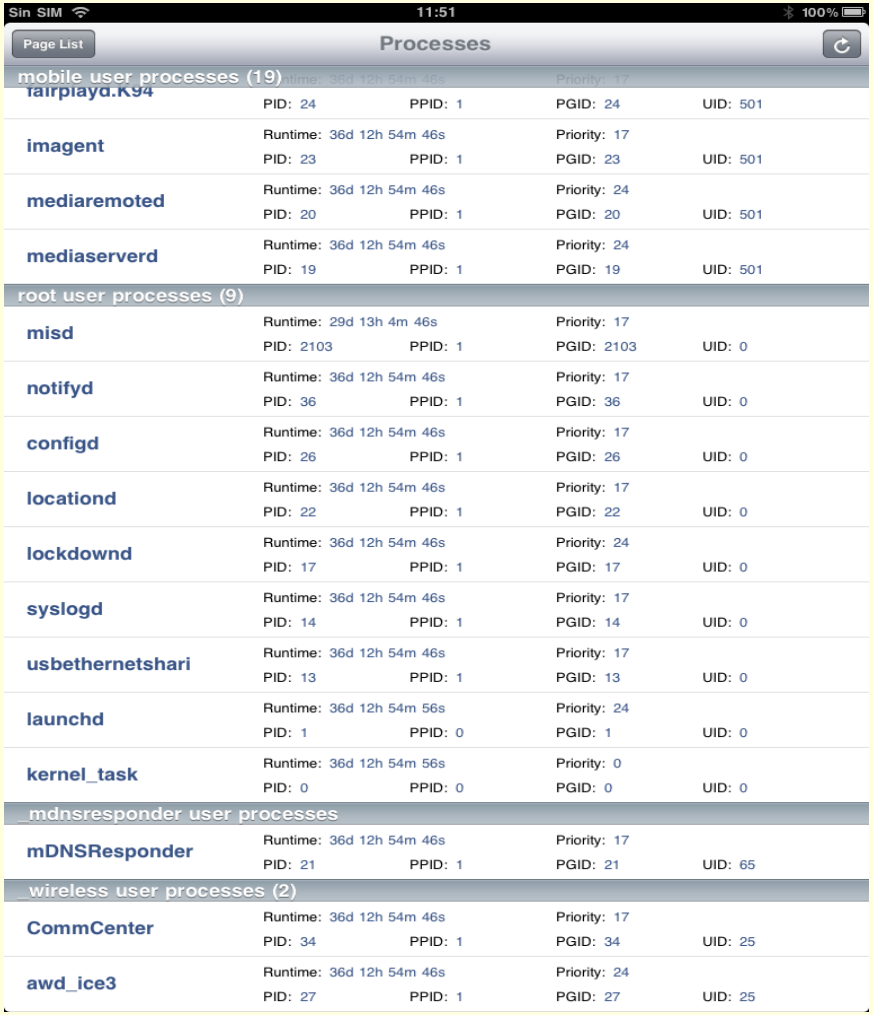


- Core OS Layer
  - Frameworks de bajo nivel.
    - Accelerate (math,DSP, vector)
    - External Accesories (hardware externo)
    - Security
    - System
      - Threading (POSIX threads), Networking (BSD sockets), File-system access, Standard I/O, Bonjour and DNS services, Locale information, Memory allocation

Procesos

- Corren bajo dos UIDs, root (0), algunos del sistema y mobile (501)
  - No se pueden manejar en forma directa.
  - Al pasar a background provocan un evento y quedan suspendido.
  - Pueden cerrarse usando la taskbar.
  - Hay una aplicación que muestra todo lo que Apple permite ver.

Captura de pantalla iPad

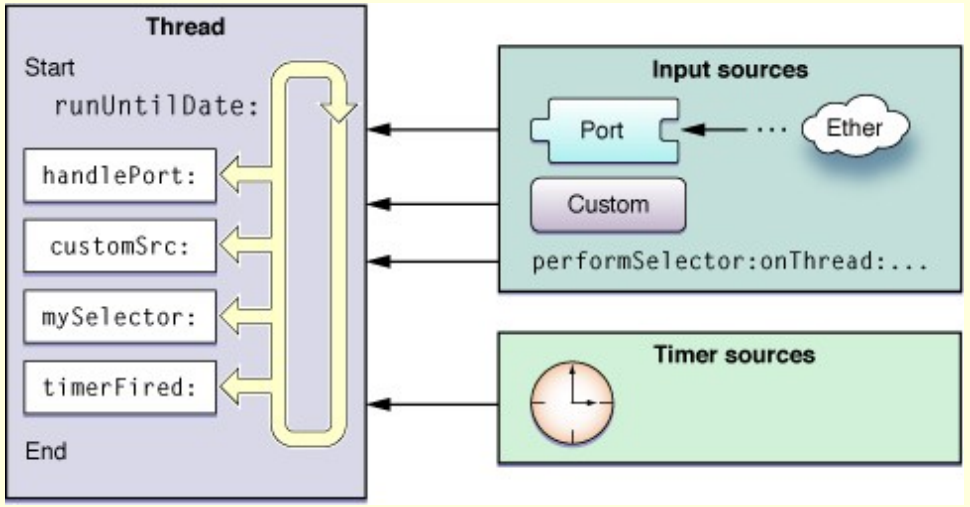


Procesos y Tasks en XNU

- Un proceso BSD tiene al menos una task.
- Las tasks indican una unidad ejecutable en su ambiente de ejecución (flavor). Puede haber tasks sin procesos BSD asociados.
  - Algunos flavors pueden ser COCOA, Kernel, BSD, Mach Threads, pthread.
- Los Threads son las unidades de ejecución.
  - El Kernel solo conoce los Mach Threads, el resto está mapeado.

Threads en IOS4

- Apple trata que no se usen threads en forma directa.
  - Cada thread tiene un run-loop para manejar eventos.
  - Un run-loop tiene un "modo" que indica que eventos recibe.



Grand Central Dispatch

- Tecnología para soportar multiprocesamiento simétrico.
  - Desarrollado por Apple y libre desde 2009 bajo licencia Apache.
  - Disponible en BSD y Linux como libdispatch.
- Es una implementación del patrón thread-pool

Concurrencia GCD

- Permite que las tasks corran en paralelo encolándolas y planificándolas ("routing")
  - Maneja la concurrencia usando Dispatch Queues.
  - El usuario indica las acciones atómicas usando Process Blocks o functions (closures).
  - Maneja eventos con Dispatch Sources.
  - Provee de agrupamientos y semáforos.